

# Security Assessment Report



# 1. Introduction

## 1.1 Purpose

This document describes the contents of security assessment for the conference system implemented by Team 5.

## 1.2 Scope

- In the system implemented by team5, the evaluation target of the server side is limited to the sfid server and the operating system in which the sfid server is running.
- In the system implemented by team5, the evaluation target of the client is limited to the sfid client, and the operating system in which the sfid client is executed is excluded from the evaluation target.

## 1.3 Definitions, acronyms, and abbreviations

- AI : Artificial Intelligence
- jpeg : A commonly used method of lossy compression for digital images
- X.509 : In cryptography, it is an ITU-T standard based on public keys (PKI) among the standards of public key certificates and authentication algorithms.
- CA : Certificate Authority
- QT : A cross-platform application and graphical user interface (GUI) framework, a toolkit, that is used for developing software.
- GUI : Graphical User Interface
- DoS : Denial of Service
- sfid : Secure Face ID, the face recognition system securely implemented by Team 5
- CVE : Common Vulnerabilities and Exposures
- sql : Structured Query Language
- xml : Extensible Markup Language
- TLS : Transport Layer Security
- SSL : Secure Sockets Layer

## 1.4 Reference

- Project Asset List.pdf
- Software Requirement Specification.pdf
- Security Requirements.xlsx
- Test Cases.xlsx
- <https://owasp.org/www-project-top-ten/>
- <https://github.com/google/AFL>
- <https://tools.kali.org/vulnerability-analysis/sfuzz>
- <https://www.wireshark.org/>

- Static\_analysis-flawfinder-5team.xlsx
- Static analysis and exploit trials.pdf
- Fuzz\_testing\_report.pdf
- TestCaseAnalysis.xlsx
- team5 document review.pdf

## 2. Assessment Approach

### 2.1 Assessment Objectives

Find vulnerabilities to interfere with achieving security goals of target system

- Confidentiality - Vulnerabilities to leak video stream data or stored personal data in the system
- Integrity - Vulnerabilities to tamper or delete Video stream data and logs
- Availability - Vulnerabilities in which to stop the service or inhibit quality of service
- Non-repudiation - Vulnerabilities to avoid the usage history of the service and records of access

### 2.2 Strategies

Identify vulnerabilities in the target system with the following strategy:

- Finds that there is a common vulnerability that may occur.
- Identify the vulnerability to the subsystem used by the target system.
- Investigate mitigation for security assets and find weaknesses of mitigations.
- Find mistakes or flaws in Design or Implementation.

### 2.3 Tactics

For each strategy, perform the following tactics.

- Find vulnerabilities that correspond to OWASP top 10.
- Do fuzz testing in order to find vulnerabilities about input validation and network ports.
- Find whether there are vulnerabilities in Jetson nano board and configuration or not.
- Review the result of static analysis in order to find vulnerabilities in code.
- Review design documents and security requirement documents to find out if there is anything missing.
- Review the code for looking for factors that can cause vulnerabilities.

### 2.4 Rules of Engagement

- The investigation scope includes the entire product submitted by Team5 and the environment in which the target system operates.
- The discovered vulnerabilities should not be described or disclosed anywhere other than the assessment report.

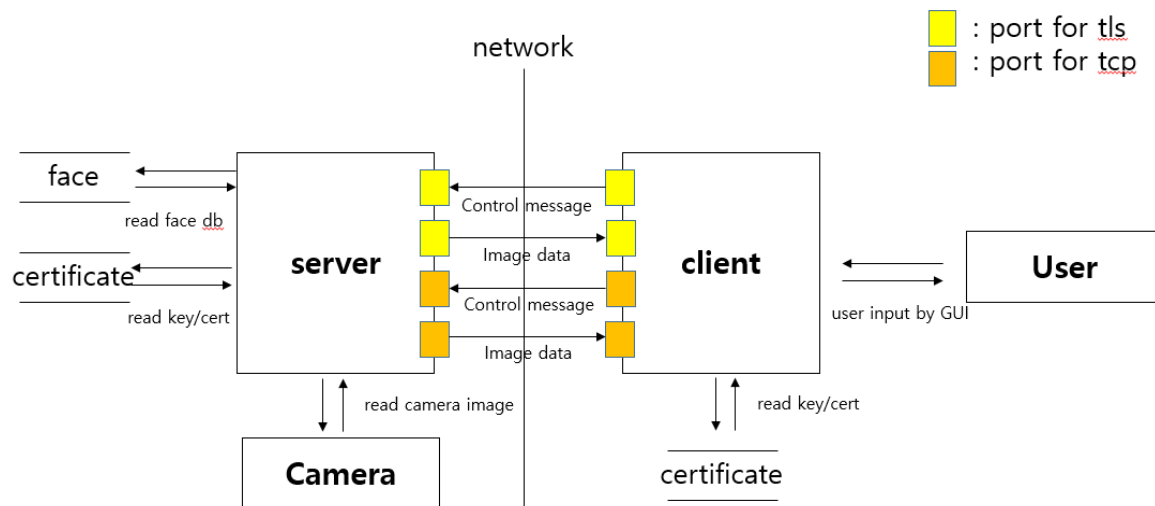
- Protect confidential information (certificate, user information, etc.) used for investigation from exposure.
- Requests are made to the contact point ([ss-5verflow@googlegroups.com](mailto:ss-5verflow@googlegroups.com)) during the investigation.
- Vulnerabilities discovered are prepared as an assessment report and presented at the final presentation meeting.

## 3. Target System Overview

### 3.1 Project Goal

The purpose of the face recognition system is to monitor attendees of video conference systems and to manage known attendees by using face AI. Especially the system must be secured from any threats as specified.

### 3.2 Software Architecture Design



Interface	Data	Threat
Client → Server	Control message (include username when registering)	Information disclosure, spoofing, tempering
Server → Client	Image Data	Information disclosure, spoofing
Certificate → Client	Certificate	Information disclosure
Certificate → Server	Certificate	Information disclosure
Face → Server	(Face)Image data	Information disclosure, spoofing

## 3.3 Functional and Quality Attribute Requirement

### 3.3.1 Functional Requirements

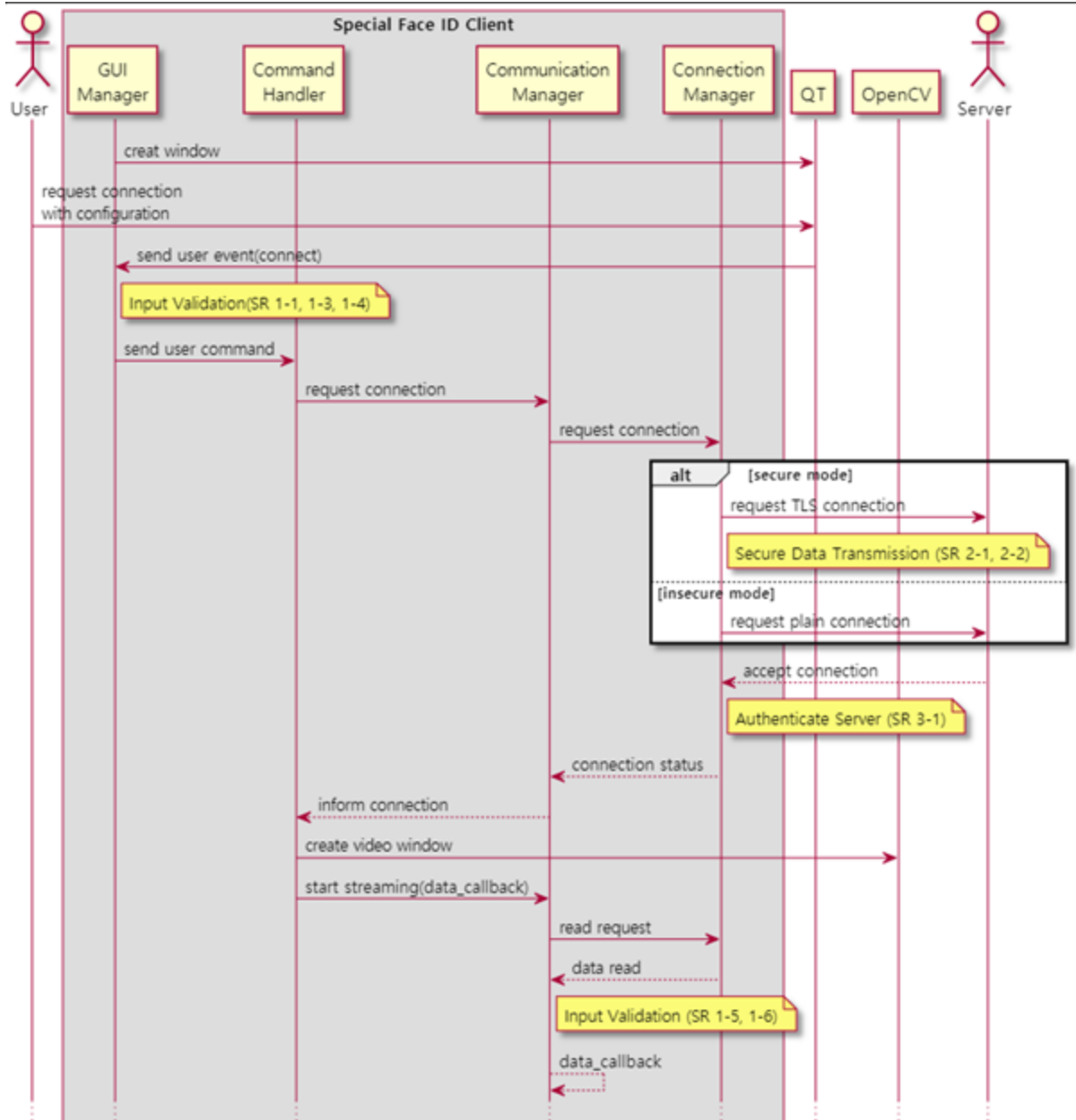
- (1) The client app shall have a user selection menu.
- (2) The client app shall be able to change communication mode with the server app to secure mode.
- (3) The client app shall be able to change communication mode with the server app to insecure mode.
- (4) The client app shall be able to add new user images to the image database with a user-specified name.
- (5) The client app shall be able to display camera video stream and face recognition results from the server app.
- (6) The client app shall be able to receive video file streams and face recognition results from the server app with a user-specified filename.
- (7) The client app shall be able to detect fault/error and then recover and report.

### 3.3.2 Quality Attribute Requirements

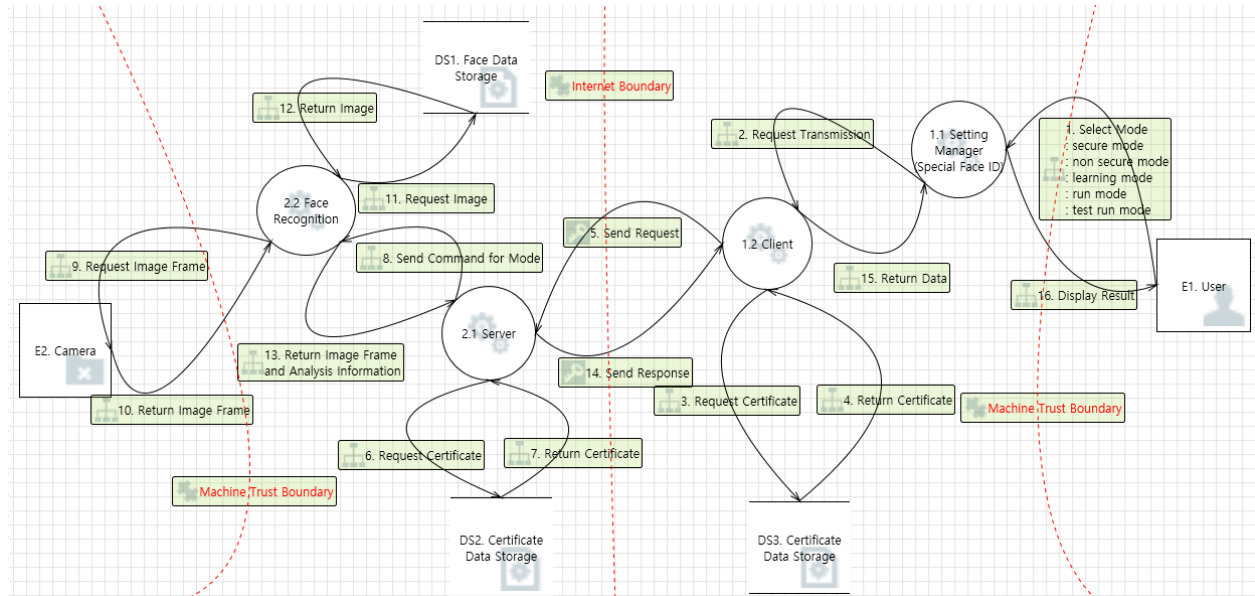
- (1) Ensure application architecture is secure.
- (2) Ensure code is written and implemented in a secure manner
- (3) Ensure application network communication is secure
- (4) Practice finding security flaws in code / applications both statically and dynamically
- (5) Ensuring that all software in both applications are architected and coded to be secure and free of vulnerabilities
- (6) Analyzing the provided initial implementation for vulnerabilities and developing solutions to mitigate.

## 3.4 Use Case Diagram

Final artifacts in team5, there are 2 use case diagrams. One is the connect scenario and the other is disconnect scenario. But actually the disconnect function is not implemented. And there is no use case diagram for registering a new user.



### 3.5 Threat Modeling and Assessment



Team5 uses DFD tools of Microsoft to get thread lists. And there are 58 threats total.

### 3.6 Mitigation and Security Requirement

Finally, below security requirements are derived. There are a total of 14 security requirements in 6 categories.

Category	Security requirement s ID	Security Requirements
Input Validation for Client Application	SR1-1	Client Application must check if the format of input IP address is in valid format
	SR1-2	Server and Client should check respectively whether the input for Username field on the Register mode is valid as a filename.
	SR1-3	Client should check if the input of the Port field is within the valid port number range.
	SR1-4	Server and client should check input validation respectively whether the input for the video file name field on the Playback mode has video file format such as .mp4.



	SR1-5	Client should check whether the image received from server is format of jpeg before displaying it.
	SR1-6	Client should compare the number of detected face and the number of its information, which are received from server, and they should be same.
Secure Data Transmission	SR2-1	After connection establishment all the data transferred between server and client must be securely encrypted
	SR2-2	Must check integrity of all the transmitted data between server and client
Secure Authentication	SR3-1	Server and Client must mutually authenticate each other with X.509 certificates
Secure Data Store	SR4-1	Images and name of registered users must be stored in secure storage to prevent access from unauthorized users
	SR4-2	Root and CA certificates must be stored in secure storage
	SR4-3	Client certificates must be stored in secure storage
Logging	SR5-1	Server and client should leave the message about the connection status as a log, respectively.
Policy	SR6-1	Client Application should run on legitimate Windows with firewall and surveillance enabled.

## 4. Vulnerability Diagnosis

### 4.1 Fuzz Testing

We tried to discover vulnerabilities through fuzzing on the module with user input of the sfid system and the network port opened to receive traffic from the external network.

#### 4.1.1 Summary of vulnerabilities derived from fuzzing

Fuzzing target	Number of vulnerabilities	Types of vulnerabilities	Fuzzing tools used	Test execution time
IP input value in client app	0	Not applicable	AFL	4 hours
Person name input value in client app	0	Not applicable	AFL	8 hours
server ports	1	Uncontrolled resource consumption	sfuzz	8 hours

#### 4.1.2 Fuzzing target

We tried to perform fuzzing on a module that has input values from the outside for each of the client and server, and as a result of the system analysis, the following targets were identified.

- IP input value in client app
- Person name input value in client app
- Port for receiving data from clients in the server

#### 4.1.3 Fuzzing strategy

For each input of server and client, fuzzing strategy was established as follows.

- Inputs in client app :
  - The client app is a QT-based GUI app, and as a result of the survey, there is no applicable QT-based GUI fuzzing tool.
  - To use the existing CLI based fuzzing tools, create a harness that tests the API that processes user input in the client app.
  - Perform fuzzing on the created harness.
- Inputs in server :
  - Identifies the port for the server to send and receive control and data with the client.
  - Attempts to connect to the identified port.

- Send random data to the connected port.

#### 4.1.4 Fuzzing tools

We analyzed the tools used for the purging of the sfid system with the following five evaluation criteria.

Fuzzing tools	QT GUI app fuzzing support	Network payload transport support	Fuzzing type	Previous experience
AFL	Not supported	Not supported	mutation based dumb fuzzer	Experienced
Peach Fuzzer	Not supported	Not supported	smart or dumb fuzzer	Not experienced
WinAFL	Not supported	Not supported	mutation based dumb fuzzer	Not experienced
sfuzz	Not supported	Supported	dumb	Not experienced

As a result of the survey, there were a number of various fuzzing tools, but to perform fuzzing within a limited time, the applicability was reviewed mainly with familiar tools with previous experience, and the following two fuzzing tools were selected.

- AFL (IP input value in client app, Person name input value in client app)
- sfuzz (Port for receiving data from clients in the server)

#### 4.1.5 Fuzzing execution

- IP input value
  - Fuzzing tool used : AFL
  - Fuzzing execution steps :
    - a. Identify the api that is called when the sfid client's ip input operation occurs.
    - b. Modify sfid client to receive QT GUI input as CLI input.
    - c. Generate seeds for fuzzing the ip input value.
    - d. Execute fuzzing using AFL.
- Person name input value
  - Fuzzing tool used : AFL
  - Fuzzing execution steps :
    - a. Identify the api that is called when the sfid client's person name input operation occurs.
    - b. Modify sfid client to receive QT GUI input as CLI input.
    - c. Generate seeds for fuzzing the person name input value.
    - d. Execute fuzzing using AFL.

- Sfid server ports
  - Fuzzing tool used : sfuzz
  - Fuzzing execution steps :
    - a. Identify the ip and port open by the sfid server.
    - b. Set input data for fuzzing
    - c. Fuzzing the network ports which are used by the server.

#### 4.1.6 Fuzzing results

- IP input value
  - When input validation (regex) is not applied to ip input value, 0 crash occurs.
  - When input validation (regex) is applied to ip input value, 0 crash occurs.
  - As a result of code analysis, after validation using regular expressions for the input value of the IP address, it is confirmed that the coding is done so that there is no separate processing until delivery to getaddrinfo().
- Person name input value
  - When input validation (regex) was not applied to person name input, 1 crash occurs.

```

american fuzzy lop 2.57b (sfid-client)

process timing
  run time : 0 days, 0 hrs, 32 min, 47 sec
  last new path : 0 days, 0 hrs, 32 min, 37 sec
  last uniq crash : 0 days, 0 hrs, 22 min, 45 sec
  last uniq hang : none seen yet
cycle progress
  now processing : 3* (60.00%)
  paths timed out : 0 (0.00%)
stage progress
  now trying : splice 13
  stage execs : 27/32 (84.38%)
  total execs : 294k
  exec speed : 95.37/sec (slow!)
fuzzing strategy yields
  bit flips : 0/544, 0/539, 0/529
  byte flips : 0/68, 0/63, 0/53
  arithmetics : 0/3804, 0/521, 0/261
  known ints : 0/274, 0/1627, 0/2201
  dictionary : 0/0, 0/0, 0/0
  havoc : 1/125k, 1/157k
  trim : 99.51%/74, 0.00%

overall results
  cycles done : 94
  total paths : 5
  uniq crashes : 1
  uniq hangs : 0

map coverage
  map density : 0.06% / 0.06%
  count coverage : 1.00 bits/tuple

findings in depth
  favored paths : 2 (40.00%)
  new edges on : 2 (40.00%)
  total crashes : 1 (1 unique)
  total tmouts : 4949 (5 unique)

path geometry
  levels : 2
  pending : 0
  pend fav : 0
  own finds : 1
  imported : n/a
  stability : 100.00%

[cpu:390%]

```

- As a result of performing AFL fuzzing for the case of not applying regex-based input validation, the input values that cause crash are as follows.

```

yujin@yujin-VirtualBox:~/work/phase2/sfid-assessment/sfid-client-master/src/build$ ./sfid-client findings/crashes/id\:000000\,
sig\:06\,src\:000003+000004\,op\:splice\,rep\:64
hello fuzz for client
before call registerPerson()
*** buffer overflow detected ***: ./sfid-client terminated
Aborted (core dumped)

```

- When input validation (regex) was applied to person name input, 0 crash occurs.
  - Confirm that the person name's mitigation (input validation) is meaningful through the validator.
- Sfid server ports
    - 1 crash caused by too many open files.

#### 4.1.7 Exploiting

As a result of the analysis, the symptom of too many open files of sfid server ports was confirmed as a vulnerability due to abnormal close of an opened socket port.

And we were able to confirm that there was an exploit that puts the server in DoS state through fuzzing.

- a. Execute sfuzz against the sfid sever

```

(kali@kali)-[/usr/share/sfuzz-db]
$ sfuzz -S 192.168.0.166 -p 6100 -T -f /usr/share/sfuzz-db/basic.http

```

- b. Server is in DoS status

After the server opens port up to 990, when it receives an additional connection request from the client, it outputs "Too many open files" and enters DoS state.

```

open: cnt: 981 port 0x7f3401a420 cfd 3f6 0
open: cnt: 982 port 0x7f3401a440 cfd 3f7 0
open: cnt: 983 port 0x7f3401a460 cfd 3f8 0
open: cnt: 984 port 0x7f3401a480 cfd 3f9 0
open: cnt: 985 port 0x7f3401a4a0 cfd 3fa 0
open: cnt: 986 port 0x7f3401a4c0 cfd 3fb 0
open: cnt: 987 port 0x7f3401a4e0 cfd 3fc 0
open: cnt: 988 port 0x7f3401a500 cfd 3fd 0
open: cnt: 989 port 0x7f3401a520 cfd 3fe 0
open: cnt: 990 port 0x7f3401a540 cfd 3ff 0
ERROR on accept: Too many open files
AcceptTcpConnection Failed

```

### 4.1.8 Additional work for fuzzing

The sfid system is a server-client based system, and when the server crashed during fuzzing, there was a problem that the server was not recovered to its normal state. This made it difficult to perform continuous fuzzing. So, when the server goes down, we create a code modification and a script so that it can be restarted in a normal state.

```
@@ -643,7 +649,7 @@ int main(int argc, char *argv[])
    if ((TlsConnectedPort = AcceptTcpConnection(TlsListenPort, &tls_cli_addr,
&tls_clilen)) == NULL) {
        conn_log.err("AcceptTcpConnection Failed\n");
        printf("AcceptTcpConnection Failed\n");
+        return(-1);
```

## 4.2 Penetration testing

Try to analyze and assess to target system, gather information and try to find vulnerabilities and exploit it. Sfid system consists of a client program running on windows and a server program running on ubuntu.

Environment of Windows OS is not fixed and guided. Following is the environment of ubuntu OS.

#### **uname -a**

```
Linux LgFaceRecProject 4.9.201-tegra #1 SMP PREEMPT Fri Feb 19 08:40:32 PST 2021
aarch64 aarch64 aarch64 GNU/Linux
```

#### **lsb\_release -a**

No LSB modules are available.

Distributor ID: Ubuntu

Description: Ubuntu 18.04.5 LTS

Release: 18.04

Codename: bionic

#### **cat /proc/version**

```
Linux version 4.9.201-tegra (buildbrain@mobile-u64-5294-d8000) (gcc version 7.3.1
20180425 [linaro-7.3-2018.05 revision d29120a424ecfbc167ef90065c0eeb7f91977701]
(Linaro GCC 7.3-2018.05) ) #1 SMP PREEMPT Fri Feb 19 08:40:32 PST 2021
```

The below library and service should be installed to the operating system and that are mentioned in the Installation guide.

- Opencv, openssl, cryptmount, cryptsetup

Vulnerability analysis is performed sequentially for the following:

- Analyze data sent and received by servers and clients
- Analysis of libraries used/installed by servers and clients for operations
- Analyze vulnerabilities to other services, ports operating in a system/OS-enabled environment

	Analysis step	Tools / Method
Data analysis	It checks the network packets sent and received by the server and client, and checks whether important information is exposed or if authentication is underway between each other.	Wireshark, netcat
Library analysis	It checks the version of the library that the system must be installed to operate or required to operate. It checks the official page to see if there are any vulnerabilities in the library.	Internet search
Service and port analysis	It identifies which ports are being provided for which services in the environment in which the program operates. Verify that these services have vulnerabilities and that they can also affect the functionality of this program.	Netcat, netstat

After conducting each analysis, we will conduct the exit test for the relevant analysis.

#### 4.2.1 Network analysis

In non-secure mode, the wireshark tool captures network packets that clients and servers send and receive, and based on this, it checks whether secure mode sends and receives data without multiple security threats.

1. It contains information on how many people are included in Camera, what are the names of each person, and whether each person is registered.

**Total data size**

Offset(h)	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	Decoded text
00000000	00	00	f0	a7	8f	ef	00	00	af	00	00	00	e7	00	00	00	..8\$.i..
00000010	00	00	00	00	a0	00	00	00	00	55	6e	6b	6e	6f	77	6e	h... ..Unknown
00000020	00	83	f5	81	55	00	00	00	70	1e	00	14	7f	00	00	00	.fδ.U...p.....
00000030	c8	e0	71	26	7f	00	00	00	40	d9	71	26	7f	00	00	00	.....p.....
00000040	00	8d	cb	79	c0	e4	30	19	60	d9	71	26	7f	00	00	00	.....p.....
00000050	40	76	f5	81	55	00	00	00	10	da	71	26	7f	00	00	00	@vδ.U...Ûq&....
00000060	02	00	00	00	00	00	00	00	70	1e	00	14	7f	00	00	00	.....p.....
00000070	40	db	71	26	7f	00	00	00	90	d9	71	26	7f	00	00	00	@Ûq&....Ûq&....
00000080	00	64	f5	81	55	00	00	00	98	dc	71	26	7f	00	00	00	.dδ.U...~Ûq&....
00000090	40	db	71	26	7f	00	00	00	a0	d9	71	26	7f	00	00	00	@Ûq&....Ûq&....
000000a0	70	4c	f5	81	55	00	00	00	c0	d9	71	26	7f	00	00	00	pLδ.U...ÀÛq&....
000000b0	40	db	71	26	7f	00	00	00	c0	d9	71	26	7f	00	00	00	@Ûq&....ÀÛq&....
000000c0	2c	4d	f5	81	55	00	00	00	d0	d9	71	26	7f	00	00	00	,Mδ.U...ÐÛq&....
000000d0	24	b5	aa	7d	7f	00	00	00	a8	47	ef	8c	7f	00	00	00	\$µ*}.....Gi&....
000000e0	98	dc	71	26	7f	00	00	00	10	da	71	26	7f	00	00	00	~Ûq&....Ûq&....
000000f0	80	b5	f5	81	55	00	00	00	e0	1b	00	14	7f	00	00	00	€µδ.U...à.....
00000100	6c	b5	f5	81	55	00	00	00	98	dc	71	26	7f	00	00	00	lµδ.U...~Ûq&....
00000110	cf	06	00	00	00	00	00	00	e2	d1	de	19	ff	d8	ff	e0	Ï.....ãÑþ.yøÿà
00000120	00	10	4a	46	49	46	00	01	01	00	00	01	00	00	01	00	..JFIF.....
00000130	ff	db	00	43	00	06	04	05	06	00	00	00	00	00	00	00	ÿÿ.....
00000140	07	06	08	0a	10	0a	0a	09	09	0a	14	0e	02	0c	10	17	.....
00000150	14	18	18	17	14	16	16	1a	1d	25	1f	1a	1b	23	1c	16	.....\$...#..
00000160	16	20	2c	20	23	26	27	29	2a	29	19	1f	2d	30	2d	28	...-#s\)*)-0-/

- Video screen information is sent from the server to the client.  
After receiving the data, you can collect the parts after file signature and acquire the image screen through assemble data.



- When using the User Register feature in the Client, it is sent to the server with the name of the user who wants to register.



## control msg num

## new register name

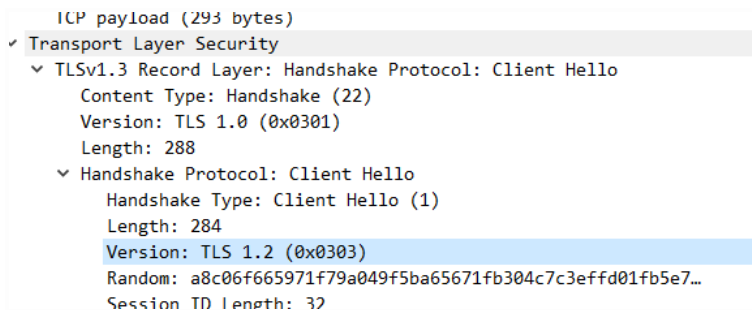
```

000001e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000001f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000200 00 03 00 09 68 65 64 61 65 73 69 6b 00 00 00 00 ..hedaesik..
00000210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

Check the secure mode to see if the data exposed to the network communication is encrypted and communicated, and mutual authentication is taking place.

### 1. tls handshake for encrypt communication



No.	Time	Source	Destination	Protocol	Length	Info
59	2021-06-25 10:43:32.772224	192.168.0.149	192.168.0.166	TCP	54	55456 → 8201 [ACK] Seq=294 Ack=2921
60	2021-06-25 10:43:32.772292	192.168.0.166	192.168.0.149	TLSv1.3	867	Application Data, Application Data,
61	2021-06-25 10:43:32.785718	192.168.0.149	192.168.0.166	TLSv1.3	1514	Change Cipher Spec
62	2021-06-25 10:43:32.785722	192.168.0.149	192.168.0.166	TCP	1514	55456 → 8201 [ACK] Seq=1754 Ack=373
63	2021-06-25 10:43:32.785729	192.168.0.149	192.168.0.166	TCP	1230	55456 → 8201 [PSH, ACK] Seq=3214 Ac
64	2021-06-25 10:43:32.789675	192.168.0.149	192.168.0.166	TCP	66	55457 → 8200 [SYN] Seq=0 Win=65535
66	2021-06-25 10:43:32.816361	192.168.0.166	192.168.0.149	TCP	56	8201 → 55456 [ACK] Seq=3734 Ack=175
67	2021-06-25 10:43:32.816362	192.168.0.166	192.168.0.149	TCP	56	8201 → 55456 [ACK] Seq=3734 Ack=321
68	2021-06-25 10:43:32.816363	192.168.0.166	192.168.0.149	TCP	56	8201 → 55456 [ACK] Seq=3734 Ack=435
69	2021-06-25 10:43:32.816438	192.168.0.149	192.168.0.166	TLSv1.3	460	Application Data, Application Data,
70	2021-06-25 10:43:32.836648	192.168.0.166	192.168.0.149	TCP	68	8200 → 55457 [SYN, ACK] Seq=0 Ack=1
71	2021-06-25 10:43:32.836651	192.168.0.166	192.168.0.149	TCP	56	8201 → 55456 [ACK] Seq=3734 Ack=475
72	2021-06-25 10:43:32.836806	192.168.0.149	192.168.0.166	TCP	54	55457 → 8200 [ACK] Seq=1 Ack=1 Win=
73	2021-06-25 10:43:32.843941	192.168.0.149	192.168.0.166	TLSv1.3	347	Client Hello
74	2021-06-25 10:43:32.849975	192.168.0.166	192.168.0.149	TCP	1514	8201 → 55456 [ACK] Seq=3734 Ack=475
75	2021-06-25 10:43:32.863233	192.168.0.149	192.168.0.166	TLSv1.3	100	Application Data
76	2021-06-25 10:43:32.863236	192.168.0.166	192.168.0.149	SSH	166	Server: Encrypted packet (len=112)
77	2021-06-25 10:43:32.863237	192.168.0.166	192.168.0.149	TCP	1514	8201 → 55456 [ACK] Seq=5237 Ack=475

2. Data is encrypted so we can't get user name and command message

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f Decoded text
00000000 17 03 03 02 11 a2 72 e5 24 82 10 50 a8 36 91 ef 0....crâ$, .P"6'i
00000010 2b 40 b5 21 18 67 ac 0a b5 d4 52 04 6e 17 0f 66 +@p!.g-.µÔR.n..f
00000020 ce 21 c7 3d ac 1a 17 cf 3b c7 70 99 ca fc 2e a8 Î!Ç=...Î;Çp"Ëü."
00000030 9f f9 e5 c0 94 74 d9 e7 79 64 e8 5e 18 99 fe 69 Ÿùää"tÛçydè^."pi
00000040 b6 44 fa c2 34 55 20 b3 61 f0 35 57 b8 65 f6 3d ¶DúÂ4U 'a85W,eö=
00000050 5b 80 cd 8a 00 62 86 3d 9c e2 b9 2e 74 0b f9 c1 [€ÍŠ.b†=œâ¹.t.ùÁ
00000060 04 e6 c4 9d ed 5c b2 ef f0 56 eb 1d 02 f5 e6 56 æÏ ÿ\5YAVE ÅæV

```

#### 4.2.2 Service/Library analysis used in server/client program

Server and Client use are openssl and opencv. And server use crypt mount utilities and libraries.

	Library	Version	Detail	CVE
Server	OpenCV	4.5.1		
	openssl	1.1.1	Server's openssl version is lower than client's version. In the developer guide, there is no specific installation guide. When using apt install command, 1.1.1 version was installed.	<a href="#">Openssl CVE list</a>
	cryptmount	5.2.4-1		
	cryptsetup	2.0.2-1		<a href="#">CVE list for cryptsetup</a>
	libdevmapper	1.02.145-4.1		
	libgcrypt20	1.8.1		<a href="#">CVE list for libgcrypt</a>
Client	OpenCV	4.5.1		
	openssl	1.1.1k		
	qt	5.15.2		

#### 4.2.3 Port analysis

We use netstat and nmap to list open port.

```
Host is up (0.00051s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
53/tcp    open  domain       dnsmasq 2.79
111/tcp   open  rpcbind      2-4 (RPC #100000)
3389/tcp  open  ms-wbt-server xrdp
5000/tcp  open  upnp?
5001/tcp  open  tcpwrapped
6000/tcp  open  X11?
6001/tcp  open  X11:1?
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Total 8 ports are open and 4 ports are used for the camera program. (5000/5001/6000/6001) And 111 and 3389 ports are open. So we try to find the vulnerability of xrdp and rpcbind.

There is no CVE of xrdp but some CVE of rpcbind are exist.

CVE ID	Category	Exploit code exist
CVE-2010-2061	Escalation of privilege	Not find
CVE-2010-2064	Escalation of privilege	Not find
CVE-2017-8779	Dos	Exist

4.2.4 Exploit Test

1. Attempts and results of an exit to the port used by the program  
  
(1) When the server program is running and the client is not yet connected, the connection is denied if someone outside reads the information on port 6000 and 6001, or attempts to connect invalidly, but the server program or port enters the invalid state.  
  
(2) After that when a normal client attempts to connect, the server program terminates with segment fault.

```
TRY TO RELEASE VSTREAMER
VSTREAMER RELEASED
Counted 0 frames in 0 seconds! This equals nanfps.
(LgFaceRecDemoTCP_Jetson_NanoV2:7550): GStreamer-CRITICAL **: 19:17:27.636: gst_mini_object_set_qdata: assertion 'object != NULL' failed
Segmentation fault (core dumped)
```

2. The dos attack was successful through the rpcbind port (111) open to the system, and the client was unable to receive the camera image normally.

3. An attempt was made to exploit through a vulnerability in openssl or other library/service used by the server, but was unsuccessful because no script existed in the metasploit or the poc code could not be found.

#### 4.2.5 Result

Overall, the security requirements for the output seem to be well applied. Encryption, mutual authentication, etc. were considered.

However, when we scan the port of the program through netstat before connecting from client, server status is changed to abnormal status. When a normal client tries to connect to the server, the server program is down. The program needs to be modified so that the program can shut down the socket connection normally and wait for the connection again.

Because dos attacks are possible through rpcbind, commercial products should need to disable the rpcbind service and additionally set up firewalls on the network to prevent dos attacks.


The installation guide on the Client side clearly stated the installation version of openssl, but there was no separate guide for server side openssl installation, and 1.1.1 version is automatically installed on Ubuntu, requiring the latest 1.1k version installation guide and application. There are many security patches between 1.1.1 and 1.1.1k versions, which require library updates.

### 4.3 Static Analysis

We reviewed all issues and tried to find vulnerabilities that can be the actual target of attack. Since the client does not have valuable assets, we focused on the server side. If we are lucky, we may be able to run a shell through code injection on the server.

#### 4.3.1 Static analysis using FlawFinder

Based on the goals described above, we selected which static analysis tool to use from among the options below. The options were based on the tools we had used in the previous phase. (

 static-analysis report )

- SonarCloud : Due to a limitation to prepare the build environment on linux, it had not been used on phase 1.
- Code x-ray : Most of the issues were not buffer related but they were about variable uninitialized issues.
- FlawFinder : Simple but easy to find buffer overflow related issues, like fixed-size buffer or detect risky functions which manipulate string.

### 4.3.2 Categorize actual issues

There were 46 issues on the server-side and 12 issues on the client-side. I have reviewed all issues and tried to find vulnerabilities that can be actual targets of attack. If there was a possibility of an attack, set 'Need Investigation', otherwise, if there was no possibility, set 'Ignore' and set 'False positives' if it was not an actual issue.

	False positives	Ignore	Need Investigation	Total
sfid-server-master	3	31	12	46
sfid-client-main	1	3	8	12

### 4.3.3 Classify where the issue come from

Compare the results of FlawFinder with original source code and figure out whether the issue comes from base code or generated from the code which is written by dev-team(a.k.a Team 5). As the result of the classification as below shows that all of the issues which 'need investigation' come from the written code by the dev-team. Even though several issues were still detected from base code, they had been already classified as 'ignored' on 1st review.

		False positives	Ignore	Need Investigation	Total
sfid-server-master	base code	3	22	-	46
	modified	-	9	12	
sfid-client-main	base code	-	-	-	12
	modified	1	3	8	

### 4.3.4 In-depth code review

Process in-depth code review to find actual attack point through the issue.

#### Issue #19

19	/LgFaceRecDemoTCP_Jetson_NanoV2/src/main.cpp	57	2	buffer	char:Statically-sized arrays can be improperly restricted, leading to potential overflows or other issues (CWE-119/CWE-120). Perform bounds checking, use functions that limit length, or ensure that the size is larger than the maximum possible length.	Need investigation	fixed(256) buffer for nameToRegister, but not a local but global
----	--	----	---	--------	--	--------------------	--

- `nameToRegister` is statically-sized global array
- Since the incorrect use of the logical operator, even if `recvData.dataLen` is longer

than MAX\_NAME\_LEN(256) it can proceed.

```

347     } else if (recvData.msgType == E_MSG_ADD_USER ) {
348         if(recvData.dataLen > 0 || recvData.dataLen < MAX_NAME_LEN) {
349             memset(nameToRegister, 0, sizeof(nameToRegister));
350             printf("datalen = %d\n", recvData.dataLen);
351             strcpy(nameToRegister, (char*)(buffer + sizeof(TClientMsg)), recvData.dataLen); //flawfinder_5ver
352             printf("name to register = %s\n", nameToRegister);
353             operationMode = E_MODE_ADD_USER;
354         } else {

```

- The attack trials with this issue is continued in '5.3.2 Buffer overrun by sending manipulated data to control socket'

## Issue #20

20	/LgFaceRecDemoTCP_Jetson_NanoV2/src/main.cpp	304	2	buffer	char:Statically-sized arrays can be improperly restricted, leading to potential overflows or other issues (CWE-119/CWE-120). Perform bounds checking, use functions that limit length, or ensure that the size is larger than the maximum possible length.	Need investigation	modified	fixed size(512) of local buffer in socketChat()
----	--	-----	---	--------	--	--------------------	----------	---

- A fixed-length buffer used to receive data from a client through a socket at line 304 in main.cpp.
- But, overflow could not be made due to the code receiving the data by limiting the length.

```

300 void* socketChat(void *arg){
301     // find if same connFd exist
302     // if exist do send video
303
304     unsigned char buffer[BUF_SIZE] = {0}; //flawfinder_5verflow : ignore - perform bounds checking
305     int retval;
306     TConnCli* pConnCli = (TConnCli*) arg;
307     //int clientfd = pConnCli->connFd;
308     TConnPort* TcpConnectedPort = pConnCli->TcpConnectedPort;
309     int clientfd = TcpConnectedPort->ConnectedFd;
310     CONN_MODE mode = pConnCli->mode;
311     SSL* ssl = pConnCli->ssl;
312
313     printf("socketChat\n");
314     while(1){
315         memset(buffer, 0, BUF_SIZE);
316         if (mode == E_CONN_TCP) {
317             retval = ReadDataTcp(TcpConnectedPort, buffer, BUF_SIZE);
318         } else {
319             retval = SSL_ReadDataTcp(ssl, TcpConnectedPort, buffer, BUF_SIZE);
320         }
321     }

```

## 4.4 Mitigations Review

We evaluate what and how team5 decided to protect and review whether their mitigations are appropriate. Also we assess whether the commonly occurring vulnerability was present in the target system and their mitigations sufficiently prevented it.

## 4.4.1 Security asset, Mitigations, Security Requirements Review

### Unprotected Security asserts

Team5 identified the Project Asset List as the asset to protect.(See Project Assert List.docx)  
Among these assets, the following is defined as those that are not covered by security requirements and risk assessments that are not covered in the Project Assert List:

- The Client Application:  
It is identified by Asset, but there is no threat analysis associated with it, and there is no security requirement, risk assessment.
- The Server Application:  
It is identified by Asset, but there is no threat analysis associated with it, and there is no security requirement, risk assessment.
- Communication Protocol:  
It is identified by Asset, but there is no threat analysis associated with it, and there is no security requirement, risk assessment.
- Video file:  
It is identified by Asset, but there is no threat analysis associated with it, and there is no security requirement, risk assessment. Unlike the use cases and the descriptions described in Req13, only the specified files are playable and not fully implemented.
- Error / Fault log:  
As identified by Asset, no error log or fault log actually exists.
  - FR8 : the client app must be able to check for a fault or error.
  - SR5-1 : both server and client have connection status logs

However, the current implementation records only the connection log of the client on the server. And the connection log is not protected.

There are no judgment descriptions about client app, server app, communication protocol, but it seems to have been omitted because it is considered to be of low value to protect. (We also didn't set it as an asset to protect either)

Video file and Error/Fault log are not currently implemented, so we are not classified as vulnerabilities where threats exist.

### Assets not specific prepared of mitigation

- Certificate (client):  
A deodorization mitigation scheme has been designed and implemented for server Side certificates, but a client's certificate has been exposed without mitigation.  
Security requirements specify only 'Assume APIs for secure stage is used'. Unlike server Side, We couldn't find a reason to assume.(It appears that the easing of this part has not

been resolved within the project period.)

Clientside's certificate was specified only as 'Assume APIs for secure storage is used' and thus classified as a present vulnerability.

#### 4.4.2 OWASP Top 10 Vulnerabilities

In order to find possible vulnerabilities, we refer to a list of commonly existing vulnerabilities to determine whether they exist.

A list of common vulnerabilities was to use OWASP Top 10 Vulnerabilities. This is because not too many of the known vulnerabilities lists are being used based on authoritative statistics. OWASP top 10 is a statistic for web applications, but it is expected to be fully tolerable because it is a service in a network environment.

ID	Project related	Mitigation status	Details
A1:2017-Injection	Not-related		
A2:2017-Broken Authentication	Related	Mitigated	SR6-1 <a href="#">3.6 Mitigation and Security Requirement</a>  Confirmed by test <a href="#">4.6.3.2 verify</a>
A3:2017-Sensitive Data Exposure	Not-related		
A4:2017-XML External Entities (XXE)	Not-related		
A5:2017-Broken Access Control	Related	Mitigated	SR4-1, SR4-2 <a href="#">3.6 Mitigation and Security Requirement</a>  Confirmed by test <a href="#">4.6.3.2 verify</a>
A6:2017-Security Misconfiguration	Related	Mitigated	Confirmed by researching and penetration test <a href="#">4.2 Penetration testing</a>
A7:2017-Cross-Site Scripting XSS	Not-related		



A8:2017-Insecure Deserialization	Not-related		
A9:2017-Using Components with Known Vulnerabilities	Related	Need mitigation	Openssl library which is required for the server program should be updated to the latest version(1.1.1k)
A10:2017-Insufficient Logging & Monitoring	Related	Need mitigation	There are not enough logs left to prevent non-repudiation.

The items below belong to the top 10 but are not related to this project

- A1:2017-Injection: Target system does not use SQL.
- A3:2017-Sensitive Data Exposure: System does not use related API.
- A4:2017-XML External Entities (XXE): System does not use XML entity.
- A7:2017-Cross-Site Scripting XSS: System is not related to this threat.
- A8:2017-Insecure Deserialization: Execution code is not serialized/deserialized.

Here, we review requirements documents to find relevance to the target system and investigate whether there are any associated API calls or associated function calls in source code.

#### Mitigated Vulnerabilities

- A2:2017-Broken Authentication : According to the design documentation, both the server and the client were using mutual authentication methods, and a test in secure mode confirmed that there was no problem with the implementation, confirming that the client and server were connected when tampered with each certificate.
- A5:2017-Broken Access Control : No other privilege classification exists in the target system except the user running server. We tested whether other users of the Jetsonano system could access protected data, and found that they had no access other than that user and that cryptmount would not allow them to be encrypted and viewed.
- A6:2017-Security Misconfiguration : Investigating known vulnerabilities for the OS environment we do not find significant vulnerabilities.

#### Not Mitigated Vulnerabilities

- A9:2017-Using Components with Known Vulnerabilities: We found a known vulnerability to openssl library v1.1.1 among the components used by the server. [\[library analysis\]](#)
- A10:2017-Insufficient Logging & Monitoring: According to Target system requirements, a fault/error must be reported and each client and server must leave a log of the connection status (SR 6-1), but it is currently implemented to leave only connection records for the server. It does not appear to be enough logarithm to be mitigated against the deniability threat for each action.

## Results

Two possible vulnerabilities were identified by investigating the presence of the OWASP Top 10 Vulnerabilities. One of these (A9:2017) was discovered in other vulnerability analysis activities, Only one(A10:2017) was discovered in this activity.

The results were below expectations as a result of the investigation with about 1 man-day.

In the case of non-Web applications, only about half of the items are applied, and many overlap with other approaches.

## 4.5 Architecture / Code Review

### Approach

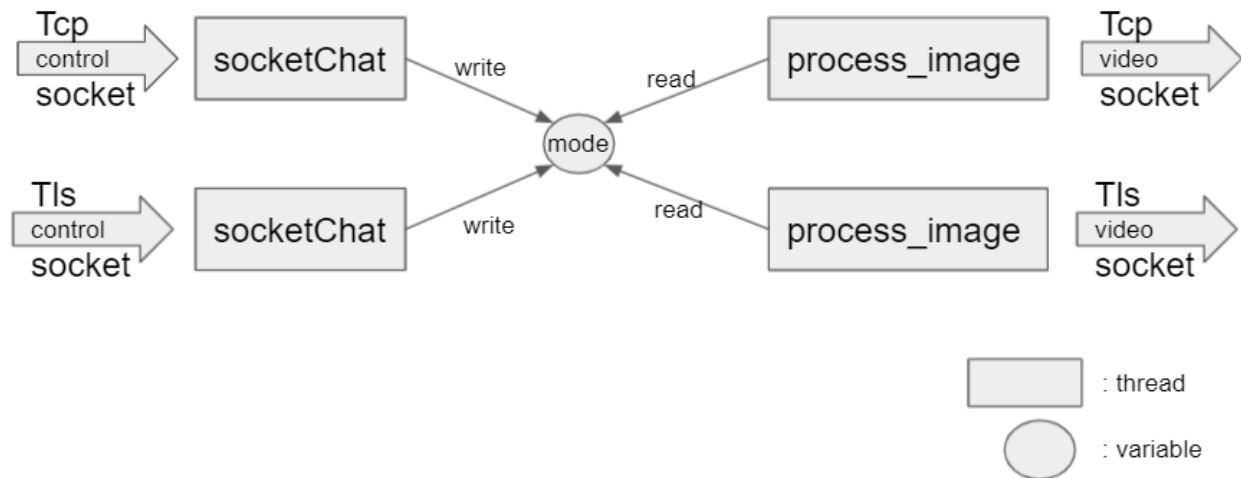
In order to find out whether there were any mistakes in the execution structure of the target system and the implementation code, a review was performed from the viewpoint of finding weaknesses in whether there were any discrepancies between the design and implementation. This is because the discrepancy between design and implementation is highly likely to indicate an unintended defect, and it is determined that mistakes or unexpected defects are highly likely to occur in the thread handling method that handles data received through the socket or communication between threads or processes.

The investigation target was scoped to the server side of the target system. This is because they thought that the impact of the client's defect on the entire system was insignificant, and that the server side's defect had a much greater effect.

(Unfortunately, We couldn't get enough information from the architecture design documentation provided, so I looked into the implemented code.)

### Findings

The figure below shows the structure of receiving control from the socket and passing the command to the thread that transmits the video.



One control variable (global variable) that controls the operation of the server is shared between the thread handling insecure mode and the thread handling secure mode. In this way, the socketChat thread handling insecure mode can control process\_image handling secure mode.

If an attacker uses this current structure, the attacker can connect to the insecure mode and manipulate the operation while receiving the live stream in the secure mode.

## 4.6 Test case analysis

Among the artifacts, we specifically reviewed the Test Cases and tried to discover vulnerabilities from them. The reason we decided to check the test case first is because we thought that if we check the test case, we can see what was important by the developing team. Also, we will be able to find an attackable case that they missed, if we are lucky.

### 4.6.1 Design vs Implementation

Before checking the test case, we checked the degree of implementation compared to the design to understand the scope that the test case should cover.

#### 4.6.1.1 Analysis

Done	100% meet requirements
Some features are not implemented	50% meet requirements; Implemented, but there are some problems in operation.
Not Implemented	not implemented

Design	Implementation
--------	----------------

Requirements ID	Requirements	Implementation result	Remark
SR 1-1	Client Application must check if the format of input IP address is in valid format	Done	
SR 1-2	Server and Client should check respectively whether the input for Username field on the Register mode is valid as a filename.	Some features are not implemented	If username is Extremely long characters, not saved to image folder despite of display the name on.
SR 1-3	Client should check if the input of the Port field is within the valid port number range.	Not Implemented	There is no input field to put the port in. port are fixed as 5000,5001/6000/6001 in source code.
SR 1-4	Server and client should check input validation respectively whether the input for video file name field on the Playback mode has video file format such as .mp4.	Not Implemented	Input field for video filename is not activated.
SR 1-5	Client should check whether the image received from server is format of jpeg before displaying it.	Done	
SR 1-6	Client should compare the number of detected face and the number of its information, which are received from server, and they should be same.	Not Implemented	
SR 2-1	After connection establishment all the data transferred between server and client must be securely	Done	

	encrypted		
SR 2-2	Must check integrity of all the transmitted data between server and client	Done	
SR 3-1	Server and Client must mutually authenticate each other with X.509 certificates	Done	
SR 4-1	Images and name of registered users must be stored in secure storage to prevent access from unauthorized users	Done	
SR 4-2	Root and CA certificates must be stored in secure storage	Done	
SR 4-3	Client certificates must be stored in secure storage	Not implemented	Stored unencrypted on the client laptop
SR 5-1	Server and client should leave the message about the connection status as a log, respectively.	Some features are not implemented	Connection Status is saved but there is no disconnect log. disconnect log is also needed.
SR 6-1	Client Application should run on legitimate Windows with firewall and surveillance enabled.	Not implemented	they comments that "Let OS do it"
FR1	The client app shall have a user selection menu. - SecureMode - InsecureMode - Live Mode - Playback Mode - Register Mode	Some features are not implemented	Playback Mode is not supported. To use playback mode, specific and name fixed video should be exist in server device.

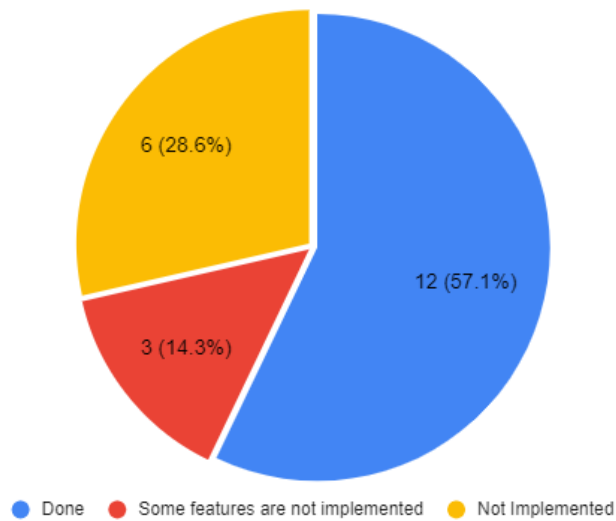
FR2	The client app shall be able to change communication mode with the server app to secure mode.	Done	
FR3	The client app shall be able to change communication mode with the server app to insecure mode.	Done	
FR4	The client app shall be able to add new user images to the image database with a user-specified name.	Done	
FR5	The client app shall be able to display camera video stream and face recognition results from the server app.	Done	
FR6	The client app shall be able to receive video file streams and face recognition results from the server app with a user-specified filename	Done	
FR7	The client app shall be able to detect fault/error and then recover and report.	Not implemented	No implementation exists in the source code. Implementation is not confirmed in tests.

#### 4.6.1.2 Summary

Implementation status for a total of 21 requirements. The results of the analysis are as follows.

Done	12
Some features are not implemented	3
Not Implemented	6

Design vs Impl



## 4.6.2 Test Execution Coverage

The written test case was written by matching the SR. However, it was a pity that a test case was not written for FR. And in most cases, the step was to press the radio button and then the Connect button during the test step, but in reality, it was designed so that the radio button could not be pressed unless the 'Connect' button was pressed first.

### 4.6.2.1 Verify

We simply looked at the written test case and checked whether the written test case was pass/fail.

Pass
Fail
team5 created a test case and then removed as the implementation changed.

Test Case ID	Test Step	Test Data	Expected Result	Requirement ID	comment
	Prepare the server application on Jetson Nano with fixed port number to connect with the client application.	./LgFaceRecDemoTCP_Jetson_NanoV2	Verify the server application is ready with displaying 'waiting'	N/A	

	Execute the client application on window laptop.		The client application displays and has control items.	N/A	
TC-01	[Positive] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Live' radio button. 3. Enter a valid ip address. 4. Click 'Connect' button	Valid IP Address : 192.168.0.100	The Jetson Nano camera stream displays with face recognized results.	SR 1-1	- radio button can be selected after pressing the 'Connect' button
TC-02	[Negative] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Live' radio button. 3. Enter a invalid ip address. 4. Click 'Connect' button	Invalid IP Address 1. Empty string 2. Include characters or symbols not IP formatted. 3. Extremely long characters	An error message pops up with "Invalid IP address. Try again" --> 'Connect' button is not activated	SR 1-1	- radio button can be selected after pressing the 'Connect' button - no error message pops up
TC-03	[Positive] 1. Select secure mode by checking 'Secure' check box. 2. Select 'Register' radio button. 3. Click 'Connect' button with valid IP address 4. Enter valid user name. 5. Click 'Register Person' button when a new person is recognized. 6. Change mode to 'Live' by selecting radio button.	Tom Cruise	1. An image file "Tom Cruise_1.jpg" is created in <img_path> 2. A new registered person 'Tom Cruise' is recognized on Live video.  [Policy of Image file creation] - A filename of a new user is composed of username to be registered and index number considering to different users who have same name.	SR 1-2	- radio button can be selected after pressing the 'Connect' button
TC-04	[Positive] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Register' radio button. 3. Click 'Connect' button with valid IP address 4. Enter an user name same	Tom Cruise  [Precondition] - Same name (Tom Cruise) is supposed to be registered already through conducting like	1. A file for duplicated name is stored with index number. (e.g. Tom Cruise_2.jpg) 2. Recognized people are shown on Live video without index number. (e.g. Tom Cruise)	SR 1-2	step2 and step3 need to be reversed





		not on the server application	activated.		
TC-08	[Negative] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Live' or 'Playback' radio button. 3. Click 'Connect' button with valid ip address.	sample.mp4' if using Playback mode.	Verify images from server are displayed well. Skip displaying an image and leave error message log if start byte of image data is compromised.	SR 1-5	- The radio button can be selected by pressing the 'Connect' button first. - Crash when switching between play mode and live mode.
TC-09	[Negative] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Live' or 'Playback' radio button. 3. Click 'Connect' button with valid ip address.	sample.mp4' if using Playback mode.	Verify images and face regions from server are displayed well. Skip displaying face regions and leave error message log if the number of recognized faces is different to the number of face region data.	SR 1-6	
TC-10	[Precondition] Prepare a packet sniffer and analysis tool such as Wireshark.  [Positive] 1. Select Secure mode by checking 'Secure' check box. 2. Select 'Register' radio button. 3. Enter valid user name. 4. Click 'Register Person' Button	Tom Cruise	1. Verify to use TLS protocol. 2. Verify the username is encrypted.	SR 2-1	need to check with wireshark - packet is encrypted
TC-11	[Precondition] Prepare a packet sniffer and analysis tool such as Wireshark.  [Negative] 1. Select Insecure mode by unchecking 'Secure' check	Tom Cruise	Verify the username is not encrypted.	SR 2-1	need to check with wireshark - As a result of checking with wireshark, the username is confirmed.

	box. 2. Select 'Register' radio button. 3. Enter valid user name. 4. Click 'Register Person' Button				
TC-12	[Precondition] Use valid credentials  [Positive] 1. Select Secure mode by checking 'Secure' check box. 2. Select 'Live' radio button. 3. Enter a valid ip address. 4. Click 'Connect' button	Certificate from Trusted Issuer (Jetson Nano Server)	Verify the server application and the client application success connection and communication.	SR 2-2	
TC-13	[Precondition] Use invalid credentials  [Negative] 1. Select Secure mode by checking 'Secure' check box. 2. Select 'Live' radio button. 3. Enter a valid ip address. 4. Click 'Connect' button	Certificate from Untrusted Issuer	Verify the server application and the client application fails connection and communication.	SR 2-2	try to run after delete the Certificate
TC-14	[Positive] 1. Run shell with admin account to the Jetson Nano 2. Access Jetson Nano imgs directory and move image files to personal laptop 3. Open an image file in the personal laptop	Use admin account	1. Verify the admin can access the Jetson Nano imgs directory 2. Verify the image files do not displays normally because of encryption	SR 4-1	image file can see without encryption
TC-15	[Negative] 1. Run shell with non-admin account to the Jetson Nano 2. Access Jetson Nano imgs directory and move image files to personal laptop	Use not admin account	Verify a user not admin can not access the Jetson Nano imgs directory	SR 4-1	
TC-16	[Positive] 1. Run shell with admin account to the Jetson Nano 2. Access Jetson Nano directory of certificates.	Use admin account	Verify the admin can access the Jetson Nano directory of certificates.	SR 4-2	

TC-17	[Negative] 1. Run shell with non-admin account to the Jetson Nano 2. Access Jetson Nano directory of certificates.	Use not admin account	Verify a user not admin can not access the Jetson Nano directory of certificates.	SR 4-2	
TC-18				SR 4-3	
TC-19				SR 4-3	
TC-20	1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Live' radio button. 3. Enter a valid ip address. 4. Click 'Connect' button	Valid IP Address : 192.168.0.100	Verify log messages in <log_path>	SR 5-1	There is a connection log, but no disconnection log.

#### 4.6.2.2 Summary

TC-02, TC-05, TC-14 do not satisfy the expected result.

Among these results, when a 256-length file name was given as a boundary test for TC-05, it was found that the file name was displayed on the screen but not stored in the storage. This can be a vulnerability that violates the integrity of the data.

Calculating Executed Tests according to the formula below :

$$\begin{array}{l} \text{Executed Tests} \\ \text{Or} \\ \text{Test Execution} \\ \text{Coverage Percentage} \end{array} = \frac{\text{Number of tests run}}{\text{Total number of tests to be run}} \times 100\%$$

Total number of test to be run = 15

Number of tests run(Number of tests passed) = 12

**Percentage of tests passed =  $12/15 \times 100 = 80\%$**

#### 4.6.3 Test Coverage

We decided to check the requirements coverage to see how well the test cases cover the software requirements.

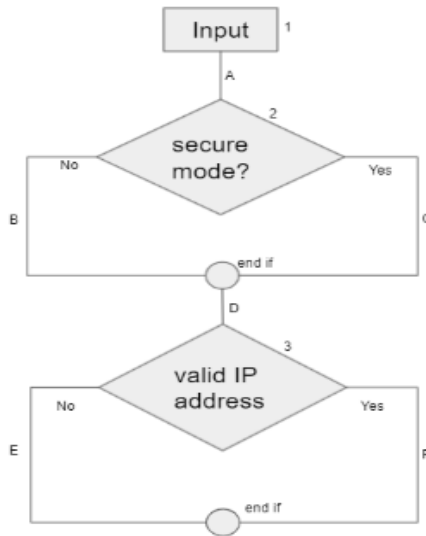
##### 4.6.3.1 Test coverage approach

To obtain requirements coverage, we checked how much the requirements are covered by the test case written with artifacts. The number of test cases required for each requirement was calculated based on the path coverage in the test coverage approach.

- Path coverage?

- Path testing is a structural testing method in order to find every possible executable path.
- Why did you choose Path Coverage?
  - Path testing is a structural testing method in order to find every possible executable path.

Ex) For SR 1-1, there are 4 possible routes as follows.



1. 1A-2B-D-3E
2. 1A-2B-D-3F
3. 1A-2C-D-3E
4. 1A-2C-D-3F

#### 4.6.3.2 verify

For each requirement, we wrote the test cases based on the path coverage and the actual number of test cases written in the ratio, and left comments on the test cases that we thought were insufficient.

The ratio was obtained by excluding test cases for unimplemented requirements.

Design		Implementation	Verification list		Test coverage *)
Requirement	Requirements	Implementation result	Test Case ID	Test Step	Test coverage result

nts ID					
SR 1-1	Client Application must check if the format of input IP address is in valid format	Done	TC-01	[Positive] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Live' radio button. 3. Enter a valid ip address. 4. Click 'Connect' button	50% No valid/invalid IP case in secure mode
			TC-02	[Negative] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Live' radio button. 3. Enter a invalid ip address. 4. Click 'Connect' button	
SR 1-2	Server and Client should check respectively whether the input for Username field on the Register mode is valid as a filename.	Some features are not implemented	TC-03	[Positive] 1. Select secure mode by checking 'Secure' check box. 2. Select 'Register' radio button. 3. Click 'Connect' button with valid IP address 4. Enter valid user name. 5. Click 'Register Person' button when a new person is recognized. 6. Change mode to 'Live' by selecting radio button.	75% No invalid user name case in secure mode
			TC-04	[Positive] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Register' radio button. 3. Click 'Connect' button with valid IP address 4. Enter an user name same with TC-03 5. Click 'Register Person' Button. 6. Change mode to 'Live' by selecting radio button.	
			TC-05	[Negative] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Register' radio button. 3. Click 'Register Person' Button. 4. Enter an invalid user name.	

SR 1-3	Client should check if the input of the Port field is within the valid port number range.	Not Implemented	-	-	not included
SR 1-4	Server and client should check input validation respectively whether the input for video file name field on the Playback mode has video file format such as .mp4.	Not Implemented	TC-06	[Positive] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Playback' radio button. 3. Enter a valid filename.	not included
			TC-07	[Negative] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Playback' radio button. 3. Enter an invalid filename.	
SR 1-5	Client should check whether the image received from server is format of jpeg before displaying it.	Done	TC-08	[Negative] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Live' or 'Playback' radio button. 3. Click 'Connect' button with valid ip address.	50% No case in secure mode
SR 1-6	Client should compare the number of detected face and the number of its information, which are received from server, and they should be same.	Not Implemented	TC-09	[Negative] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Live' or 'Playback' radio button. 3. Click 'Connect' button with valid ip address.	not included
SR 2-1	After connection establishment all the data transferred between server and client must be securely encrypted	Done	TC-10	[Precondition] Prepare a packet sniffer and analysis tool such as Wireshark.  [Positive] 1. Select Secure mode by checking 'Secure' check box. 2. Select 'Register' radio button. 3. Enter valid user name. 4. Click 'Register Person' Button	

			TC-11	[Precondition] Prepare a packet sniffer and analysis tool such as Wireshark.  [Negative] 1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Register' radio button. 3. Enter valid user name. 4. Click 'Register Person' Button	
SR 2-2	Must check integrity of all the transmitted data between server and client	Done	TC-12	[Precondition] Use valid credentials  [Positive] 1. Select Secure mode by checking 'Secure' check box. 2. Select 'Live' radio button. 3. Enter a valid ip address. 4. Click 'Connect' button	100%
			TC-13	[Precondition] Use invalid credentials  [Negative] 1. Select Secure mode by checking 'Secure' check box. 2. Select 'Live' radio button. 3. Enter a valid ip address. 4. Click 'Connect' button	
SR 3-1	Server and Client must mutually authenticate each other with X.509 certificates	Done	-	-	100% (TC12/TC13 are include this positive/negative test case)
SR 4-1	Images and name of registered users must be stored in secure storage to prevent access from unauthorized users	Done	TC-14	[Positive] 1. Run shell with admin account to the Jetson Nano 2. Access Jetson Nano imgs directory and move image files to personal laptop 3. Open an image file in the personal laptop	100%
			TC-15	[Negative] 1. Run shell with non-admin account to the Jetson Nano 2. Access Jetson Nano imgs directory and move image files	



				to personal laptop	
SR 4-2	Root and CA certificates must be stored in secure storage	Done	TC-16	[Positive] 1. Run shell with admin account to the Jetson Nano 2. Access Jetson Nano directory of certificates.	100%
			TC-17	[Negative] 1. Run shell with non-admin account to the Jetson Nano 2. Access Jetson Nano directory of certificates.	
SR 4-3	Client certificates must be stored in secure storage	Not implemented	TC-18	-	not included
			TC-19	-	
SR 5-1	Server and client should leave the message about the connection status as a log, respectively.	Some features are not implemented	TC-20	1. Select Insecure mode by unchecking 'Secure' check box. 2. Select 'Live' radio button. 3. Enter a valid ip address. 4. Click 'Connect' button	50% also need to check secure mode
SR 6-1	Client Application should run on legitimate Windows with firewall and surveillance enabled.	Not implemented	-	-	not included
FR 1	The client app shall have a user selection menu. - SecureMode - InsecureMode - Live Mode - Playback Mode - Register Mode	Some features are not implemented	-	-	Test case is not exist for FR(Functional Requirement)
FR 2	The client app shall be able to change communication mode with the server app to secure mode.	Done	-	-	Test case is not exist for FR(Functional Requirement)
FR 3	The client app shall be able to change communication mode with the server app to insecure mode.	Done	-	-	Test case is not exist for FR(Functional Requirement)
FR 4	The client app shall be able to add new user images to the image database with a user-specified name.	Done	-	-	Test case is not exist for FR(Functional Requirement)

FR 5	The client app shall be able to display camera video stream and face recognition results from the server app.	Done	-	-	Test case is not exist for FR(Functional Requirement)
FR 6	The client app shall be able to receive video file streams and face recognition results from the server app with a user-specified filename	Done	-	-	Test case is not exist for FR(Functional Requirement)
FR 7	The client app shall be able to detect fault/error and then recover and report.	Not implemented	-	-	not included

#### 4.6.3.3 Summary

The relationship between the verified requirements and the test case is shown in the table below. No test cases were written for FR, and some of the written cases were lacking.

X : not included

Requirement ID	Reqs Tested	S R 1-1	S R 1-2	S R 1-3	S R 1-4	S R 1-5	S R 1-6	S R 2-1	S R 2-2	S R 3-1	S R 4-1	S R 4-2	S R 4-3	S R 5-1	S R 6-1	FR 1	FR 2	FR 3	FR 4	FR 5	FR 6	FR 7
Test Cases	17	2	3	0	X	1	X	2	2	2	2	2	X	1	0	0	0	0	0	0	0	0
Test coverage result		50 %	75 %	X	X	50 %	X	100 %	100 %	100 %	100 %	100 %	X	50 %	X	0	0	0	0	0	0	X
TC-01	1	O																				
TC-02	1	O																				
TC-03	1		O																			
TC-04	1		O																			
TC-05	1		O																			
TC-06	0				O																	
TC-07	0				O																	
TC-08	1					O																
TC-09	0						O															
TC-10	1							O														

TC-11	1							O																
TC-12	2								O	O														
TC-13	2								O	O														
TC-14	1										O													
TC-15	1										O													
TC-16	1											O												
TC-17	1											O												
TC-18	0												O											
TC-19	0												O											
TC-20	1													O										

\*) not included : Not implemented cases are excluded from the test case coverage check

\*\*) Items that are grayed out are items that 5 teams created a test case and then removed as the implementation changed.

Calculating Requirements Coverage according to the formula below :

$$\text{Requirements Coverage} = \frac{\text{Number of requirements covered}}{\text{Total number of requirements}} \times 100\%$$

Total number of requirements = 18

Number of requirements covered = 0.5 + 0.75 + 0.5 + 1+ 1+ 1+ 1+ 1+0.5 = 7.25

**Requirements Coverage = 7.25/18 \*100 = 40.28%**

## 5. Vulnerabilities

We list the vulnerabilities discovered through 4. Vulnerability Diagnosis and analyze what impact the vulnerabilities will have.

### 5.1 Overview

#### 5.1.1 List of Vulnerabilities

The discovered vulnerabilities are as follows.

Vulnerability	severity	priority
Client certificate is exposed	High	5
Openssl vulnerability	Medium	6
Insufficient Logging & Monitoring	Low	7
OperationMode tampering	High	1
buffer overrun is possible for the nameToRegister global variable	High	1
DoS attack to opened port of rpcbind	High	1
Uncontrolled socket port due to invalid socket close handling	High	1

#### 5.1.2 Priority, Severity Classification Criteria

If a vulnerability is a vulnerability related to a threat identified and analyzed by Team 4 or Team 5, the severity is determined, and if it is a new type that has not been identified previously, it is determined to be rated according to the OWASP Risk Rating Methodology method.

The priority was determined by the degree of the direct impact on the security properties. (The lower the number of Priority, the higher the priority.) Vulnerabilities with priority 1 were selected as the same priority because it was difficult to accurately compare the relative size of the impact. (required)

If the threat is already identified by team4 or team5, evaluate the severity already analyzed.

Priorities are determined by the magnitude of the direct impact on security attributes. The lower the number of priorities, the higher the priority.) Vulnerabilities with a priority of one decided on the same priority because it was difficult to accurately compare the relative size of the impact. (Mandatory requirements)

## 5.2 Vulnerability Details

### 5.2.1 Client Certificate is exposed

description	Certificates to authenticate servers are being exposed without protection. (Assume protected without any suggestion)
method	4.4.1 securify asset, mitigation review
artifact	Security Requirements
trigger	the exposed certificate is stolen
problem	If a certificate is stolen, it is exposed to a spoofing attack, exposing the information sent by the server, and even in secure mode, the server is exposed to the attack. MITM attacks can enable modulation of the data that servers and clients send and receive.
severity	High
date	21/Jun/2021
reporter	Hanil Jang

### 5.2.2 Buffer overrun is possible for the `nameToRegister` global variable.

description	If the manipulated data is sent through the control port(6000), buffer overrun is possible for the <code>nameToRegister</code> global variable.
method	4.3 Static analysis
artifact	Static_analysis-flawfinder-5team.xlsx Static analysis and exploit trials.pdf
trigger	The exposed client's certificate is leaked.
problem	It is possible to access the control port (6000) by using the leaked certificate, and it is possible to overflow the global variable <code>nameToRegister</code> in the server by sending the manipulated data after connection.
severity	High
date	25/Jun/2021
reporter	Jinchul Kim

### 5.2.3 DoS attack to opened port of rpcbind

description	The rpcbind service is operating on the system where the server program is operating, and there is a current dos-related vulnerability in rpcbind that can be exploited.
method	4.2.3 Port analysis
artifact	4.2.4 Exploit Test
trigger	DoS attack via metasploit's rpcombscript
problem	The server failed to send image information of the camera to the client due to DoS attack, resulting in image disconnection on the client side.
severity	High
date	25/Jun/2021
reporter	Daesik Kim

### 5.2.4 OperationMode tampering

description	Shared operationMode global variable can be modified by mode change of insecure mode
method	Architecture / Code review
artifact	Code
trigger	When there is a user connecting in secure mode and a mode change packet is transmitted with insecure mode
problem	A video stream transmitted to a user connected with secure mode can be tampered with a playback video stream
severity	High
date	24/Jun/2021
reporter	Hanil Jang

### 5.2.5 Openssl vulnerability

description	Using an older version(v1.1.1) with vulnerabilities
method	4.2.2 Service/Library analysis used in server/client program

	4.4.2 OWASP Top 10 Vulnerabilities
artifact	Architecture
trigger	When an attacker attacks a vulnerability that has not been fixed
problem	Exploit is possible according to vulnerability (dos, crash, etc.)
severity	Medium
date	25/Jun/2021
reporter	Dahee Jung

### 5.2.6 Insufficient Logging & Monitoring

description	According to Target system requirements, a fault/error must be reported and each client and server must leave a log of the connection status (SR 6-1), but it is currently implemented to leave only connection records for the server. It does not appear to be enough logarithm to be mitigated against the deniability threat for each action.
method	4.4.2 OWASP Top 10 Vulnerabilities
artifact	<a href="#">Not Mitigated Vulnerabilities</a>
trigger	fault/error is occurred, connection is disconnected, connection is tried with invalid authentication.
problem	It is possible to deny because connected status logs are left only, even if system requirement and SR6-1 is defined that error and fault logs should be left.
severity	Low
date	24/Jun/2021
reporter	Hanil Jang

### 5.2.7 Uncontrolled socket port due to invalid socket close handling

description	When a socket connection is attempted from the client to the server, the port is not closed normally due to abnormal close of the socket port
method	4.1 Fuzz Testing

artifact	fuzz_testing_report.pdf
trigger	Consecutive attempts to connect to a port open by the server
problem	If the connection attempt attack continues due to not closing the port opened by the server, the server can no longer process the client's connection request and enters a DoS state.
severity	High
date	25/Jun/2021
reporter	Kyungsik Lee

## 5.3 Exploits

A possible attack was implemented through the discovered vulnerabilities.

### 5.3.1 OperationMode tampering

A video stream transmitted to a user connected with secure mode can be tampered with a playback video stream.

Misuse cases are as follows.

Misuse case1:

Precondition

- The user is connected to the server with secure mode and viewing the live stream.

Flow

- The attacker connects to the server with insecure mode.
- The attacker sends a packet to change to playback mode.

Postcondition

- The user receives a playback image (Friends), not a Live Stream.

Demo: vulm\_demo.mp4

Attack client: sfid-attack-client.zip

### 5.3.2 Buffer overrun by sending manipulated data to control socket.

As the result of code review with the issue found by the static analysis tool which is covered in '4.3 Static Analysis', we found the possibility of an attack on Issue 19 and attempted to exploit it.

- Using python script, connected to TLS control socket (port 6000) with proper private key and certificate.
- It needs to send 4 bytes which contain msgType and dataLen first.



```
typedef struct {
    uint16_t    msgType;
    uint16_t    dataLen;
    //char    data[];
} TClientMsg;
```

- To make an overflow situation, generate the msg beginning with '\x00\x03\xff\xff' which contains E\_MSG\_ADD\_USER as msgType and \xffff (which could be  $256^2 - 1 = 65535$ ) as dataLen, then the rest of the message fills in dummy characters.
- The script connects to control sockets using the proper key and certificate for the client. After the connection was established, it sent the generated msg.

```
➔ 0625_py_client git:(main) x python3 -i ./client.py
READY...
>>> msg = b'\x00\x03\xff\xff'
>>> msg = fillme(msg)
>>> msg
b'\x00\x03\xff\xffAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBB
CCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAA
AABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCC
DDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBB
BBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDDAAAABBBBCCCCDDDD\x00\x00\x00\x00'
>>> sock = connect()
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('0.0.0.0',
0)>
<ssl.SSLSocket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('0.0.0.0',
0)>
None
>>> sock.write(msg)
512
>>>
```

- As observed by using GDB, global variable `nameToRegister` gets overflowed and the data is written to the following global variables which are `_ConnCli[]` and `_ProcImg[]`.

[illegible]

<Before>

[illegible]

<After>

## 5.4 Recommendations

We suggest an alternative to mitigation of the discovered vulnerabilities. They are listed in ascending order of severity.

### 5.4.1 Client Certificate is exposed

If the certificate used for authentication is stolen, the spoofing defense, sniffing defense, and tempering defense are neutralized, so the certificate must be kept secure.

#### Solutions

Alternatives to keeping the certificate are as follows.

1. hide certificate file(public key, private key)
2. Put the certificate file in the .ssh directory and set the file access permission so that only the user can access it.
3. It is stored in the windows certificate store and retrieved from the monitoring system application through cryptoAPI.

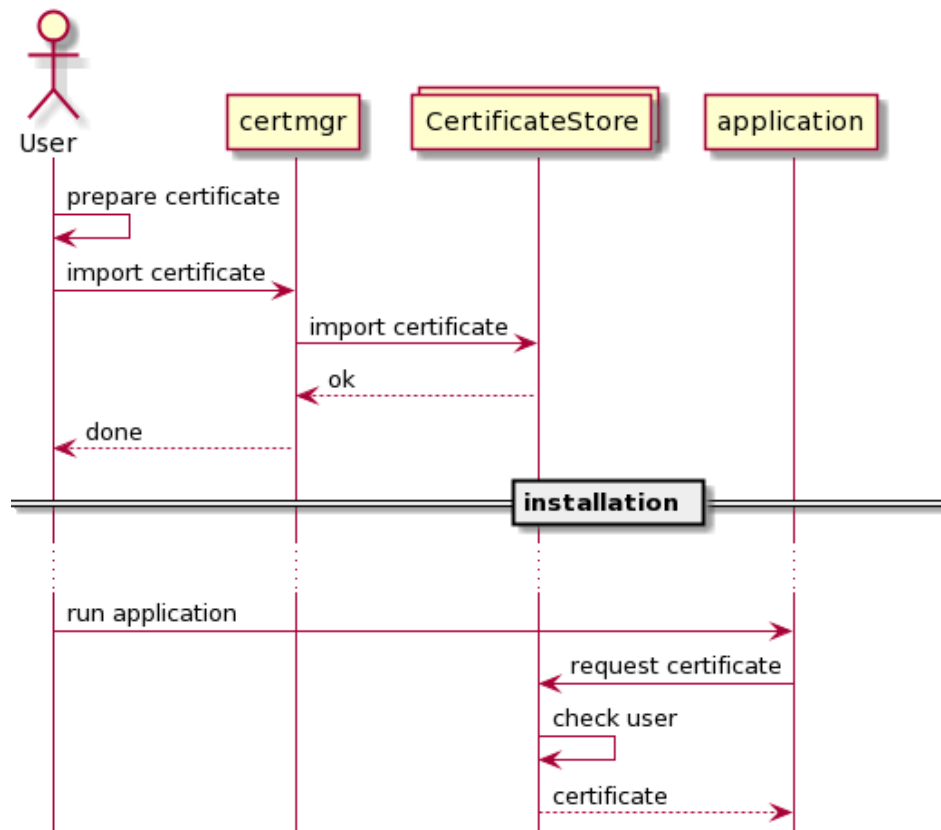
The pros and cons of applying each alternative to the current situation are as follows.

solution	pros.	cons.
1	Implementation is the simplest	Because hidden locations are inevitably exposed, it is difficult to prevent hijacking.
2	Implementation is simple. It can be used only by that user, so it can be protected to some extent from stealing by other users without permission.	If the user's account is hijacked, the certificate is leaked. It is difficult to prevent being hijacked by malware or other attacks.
3	This is the most secure way to defend against the vulnerabilities of other solutions above.	The procedure for importing a certificate into the windows certificate store should still be protected.

#### Recommended solution

To use the most effective countermeasures rather than considering trade-offs, solution3. has chosen To use this solution, the process of importing the certificate must be carried out securely.

The process of using the windows certificate store is as follows.



Consideration: import certificate

In this sequence, the part where the certificate is prepared before the user imports the certificate can be a vulnerable section where the certificate can be stolen.

#### 5.4.2 buffer overrun is possible for the `nameToRegister` global variable.

The vulnerability that we have found was caused by lack of input validation and bugs in the code. Here is recommendations to mitigate this vulnerability.

1. **Perform bound check in server side also.**

There is an input validation code on the client side, but it should apply to both, as it can directly access the server to transmit maliciously generated data.

2. **Use heap instead of statically-sized arrays.**

Statically-sized array has a risk of overflowing data to other variables. Using heap is much more flexible for various lengths of required buffer size.

### 5.4.3 Dos attack to opened port of rpcbind

Because dos attacks are possible through rpcbind, commercial products should need to disable the rpcbind service and additionally set up firewalls on the network to prevent dos attacks.

### 5.4.4 OperationMode control vulnerability

One of the following alternatives can be applied to avoid problems arising from this vulnerability.

1. The In-secure mode feature is used only within the development cycle and allows only secure mode connections when installed and operated as a product.
2. When connected to secure mode, close the socket for the open in-secure mode.
3. Use control value separately in secure mode and in-secure mode.

### 5.4.5 Openssl version vulnerability

Upgrading the openssl library to Openssl 1.1.1k can mitigate currently known vulnerabilities.

### 5.4.6 Insufficient Logging & Monitoring

It adds records of mode changes, disconnection, authentication failures, other errors, and user registration in the current log.

### 5.4.7 Uncontrolled socket port due to invalid socket close handling

SSL\_accept() waits for a TLS/SSL client to initiate the TLS/SSL handshake. Add logic to close socket fd when SSL\_accept fails.

```
@@ -599,6 +601,10 @@ int main(int argc, char *argv[])
    SSL_set_fd(p_ssl[0], TlsConnectedPort->ConnectedFd);      /* set connection
socket to SSL state */
    if( SSL_accept(p_ssl[0]) < 1 ) {
        ERR_print_errors_fp(stderr);
+       close(TlsConnectedPort->ConnectedFd);
+       delete TlsConnectedPort;
    } else {
```

```
if(tls_data_user_num == 0) {
```

## 6. Conclusions

### 6.1 Assessment summary of team5

Team potential performed a penetration test on the final artifact of Team5. Team 5 artifacts provide a streaming and face recognition service using a camera.

The purpose of this security assessment is to review the design documents, codes and outputs of the team5 and to find out whether there are sufficient mitigations to the security threat or no potential security threat.

Overall, Team 5 derived and applied mitigation measures against various security threats to the product. Team5 considered mutual authentication, encryption of communication intervals, and safe storage of data.

However, the final product has secure mode and insecure mode. it should have been considered so that each other's modes are not affected by other modes. Currently, there is a problem that affects secure mode connections as attacks through development mode are allowed during secure mode operation.

**In order to achieve the security goal, it is recommended to apply the proposed mitigation to the vulnerabilities found as having a high severity.**

Additionally, as a result of code analysis, memory overruns have been found, so improvements are needed.

### 6.2 Lessons & Learned

- Do not ignore static analysis issues, it may come with critical vulnerability.
- The same vulnerability also seems to vary in severity depending on the use scenario. While it is important to determine and apply threats and security measures, it is also important to define clear product functional requirements, usage scenarios, and operating environments.
- As it is very difficult to visualize the completeness of the system while analyzing the test cases, I felt that establishing and agreeing on metrics and criteria is very important.
- It's important to provide well-organized documentation from a security standpoint for the product's specifications, installation instructions, and usage guides.

- The fuzzing test can be useful for tests based on a large number of random inputs that are difficult for humans to do, but it is important for humans to well define the seeds of the input values in order to produce more meaningful test results.
- Even well-known open sources may have vulnerabilities, so it is important to have a habit of checking whether vulnerabilities exist before using them.