LGE Security Specialist
Studio Project

Team 5 - 5verFlow

Contents

- Team Members
- Roadmap
- Phase 1 Review
- Phase 2 Security Analysis of Classmate System (Team 6)
- Lessons Learned

Team Members

Team 5 - 5verflow

SeungWook Cha
(Team Leader)

SungJun Lee

DongHyuk Han

WooLam Kang

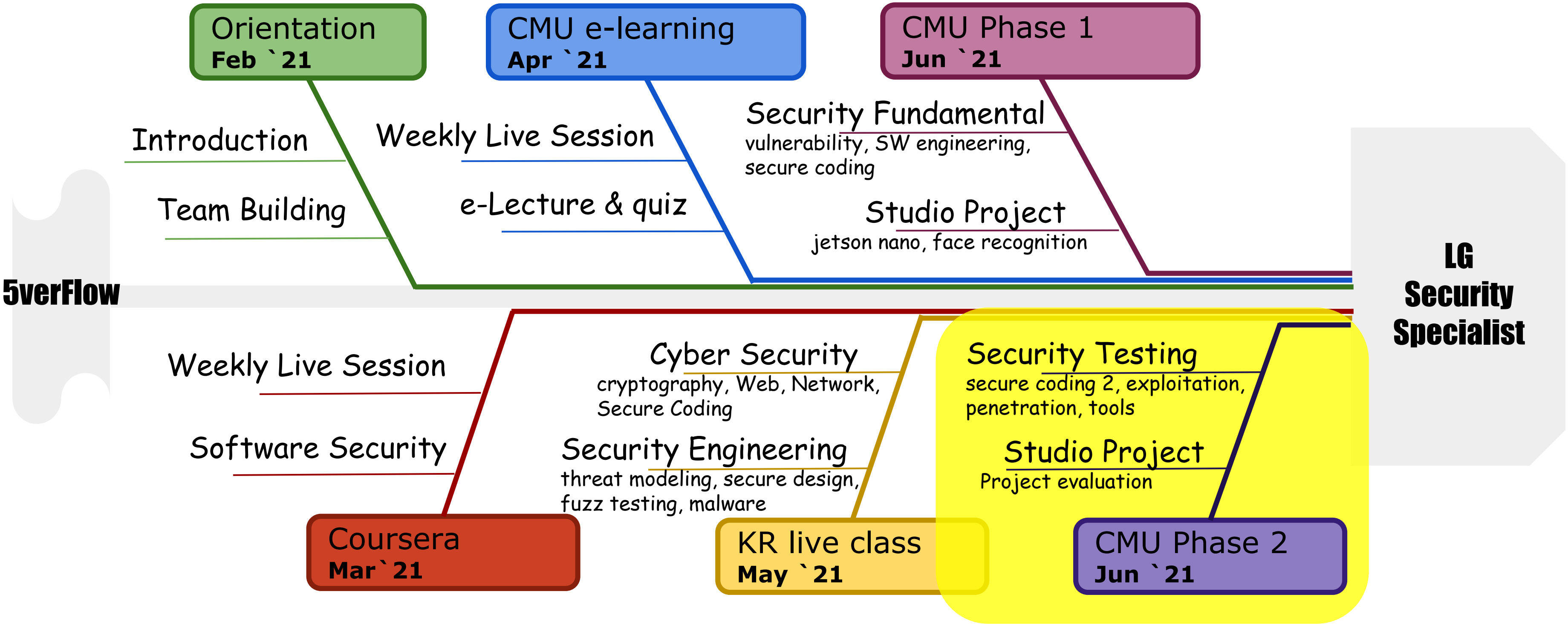
YooKyoung Choi

YoungJinn Lee

Bradley Schmerl
(Mentor)

Roadmap

To the LG security specialist



5verFlow

Orientation
Feb `21

Introduction

Team Building

CMU e-learning
Apr `21

Weekly Live Session

e-Lecture & quiz

CMU Phase 1
Jun `21

Security Fundamental
vulnerability, SW engineering, secure coding

Studio Project
jetson nano, face recognition

Coursera
Mar`21

Weekly Live Session

Software Security

KR live class
May `21

Cyber Security
cryptography, Web, Network, Secure Coding

Security Engineering
threat modeling, secure design, fuzz testing, malware

CMU Phase 2
Jun `21

Security Testing
secure coding 2, exploitation, penetration, tools

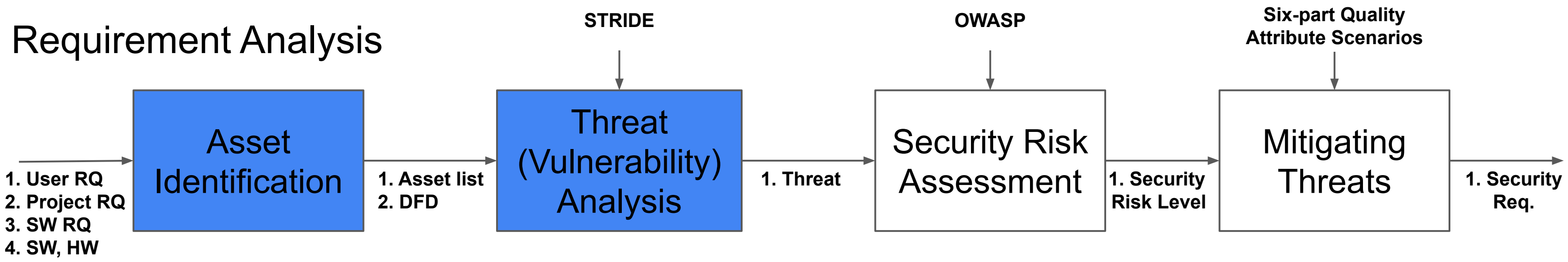
Studio Project
Project evaluation

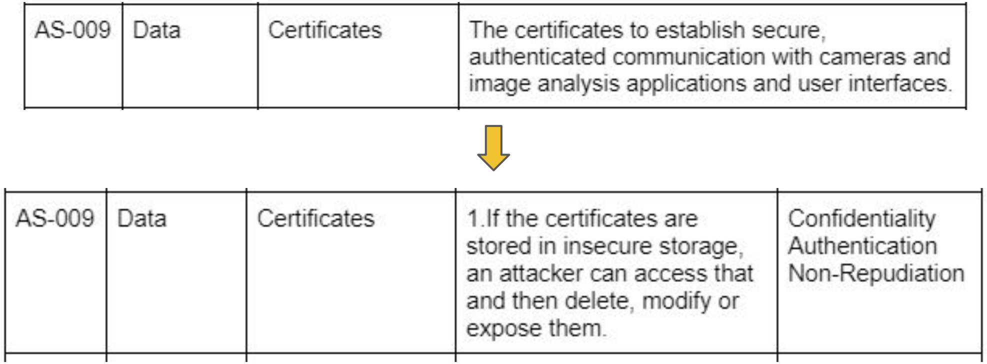
LG Security Specialist

Phase 1 Review

Requirement

Requirement Analysis

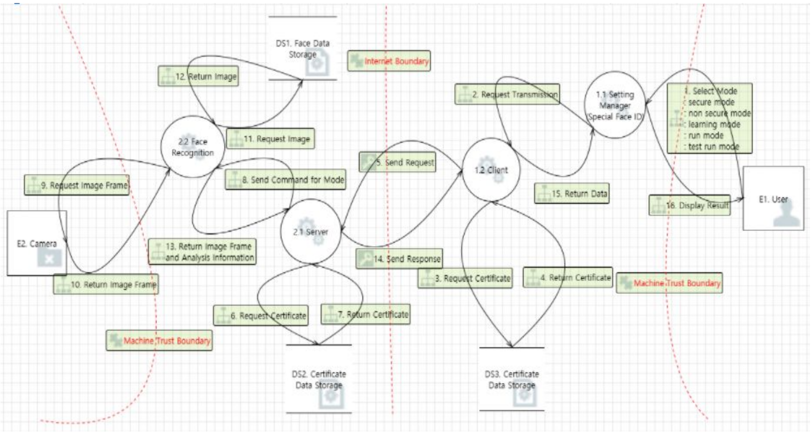




AS-009	Data	Certificates	The certificates to establish secure, authenticated communication with cameras and image analysis applications and user interfaces.

AS-009	Data	Certificates	1.If the certificates are stored in insecure storage, an attacker can access that and then delete, modify or expose them.	Confidentiality Authentication Non-Repudiation

<Asset List>

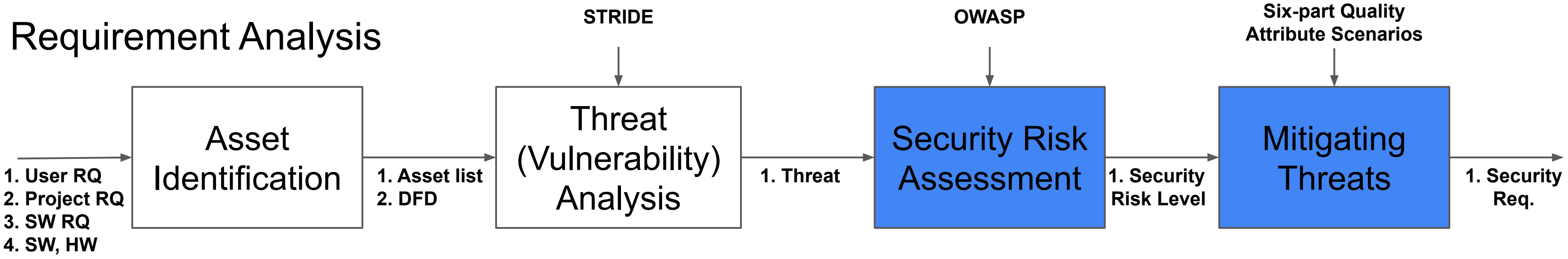


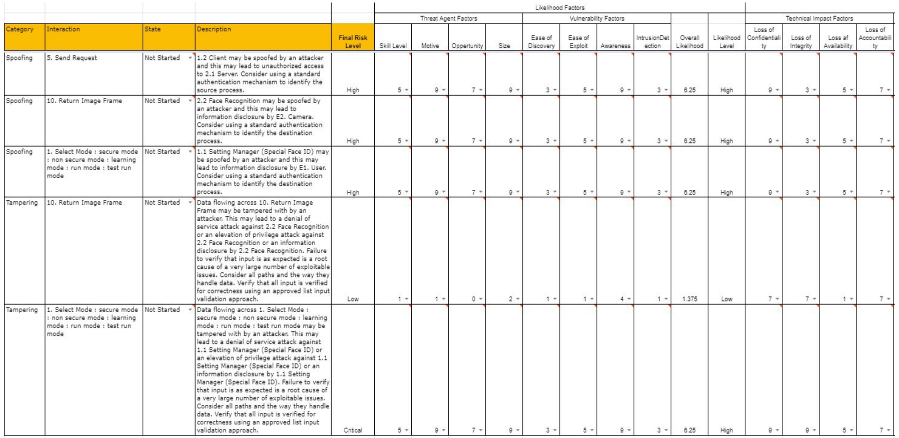
<DFD>

Requirement

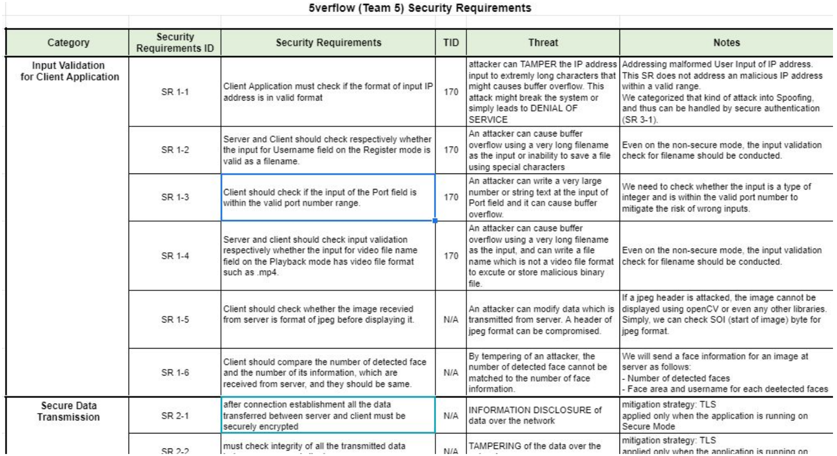
Phase 1 Review

Requirement Analysis





					Likelihood Factors													
					Threat Agent Factors				Vulnerability Factors						Technical Impact Factors			
Category	Interaction	State	Description	Final Risk Level	Skill Level	Motive	Oppertunity	Size	Ease of Discovery	Ease of Exploit	Awareness	IntrusionDetection	Overall Likelihood	Likelihood Level	Loss of Confidentiality	Loss of Integrity	Loss af Availability	Loss of Accountability
Spoofing	5. Send Request	Not Started	1.2 Client may be spoofed by an attacker and this may lead to unauthorized access to 2.1 Server. Consider using a standard authentication mechanism to identify the source process.	High	5	9	7	9	3	5	9	3	6.25	High	9	3	5	7
Spoofing	10. Return Image Frame	Not Started	2.2 Face Recognition may be spoofed by an attacker and this may lead to information disclosure by E2. Camera. Consider using a standard authentication mechanism to identify the destination process.	High	5	9	7	9	3	5	9	3	6.25	High	9	3	5	7
Spoofing	1. Select Mode : secure mode : non secure mode : learning mode : run mode : test run mode	Not Started	1.1 Setting Manager (Special Face ID) may be spoofed by an attacker and this may lead to information disclosure by E1. User. Consider using a standard authentication mechanism to identify the destination process.	High	5	9	7	9	3	5	9	3	6.25	High	9	3	5	7
Tampering	10. Return Image Frame	Not Started	Data flowing across 10. Return Image Frame may be tampered with by an attacker. This may lead to a denial of service attack against 2.2 Face Recognition or an elevation of privilege attack against 2.2 Face Recognition or an information disclosure by 2.2 Face Recognition. 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.	Low	1	1	0	2	1	1	4	1	1.375	Low	7	7	1	7
Tampering	1. Select Mode : secure mode : non secure mode : learning mode : run mode : test run mode	Not Started	Data flowing across 1. Select Mode : secure mode : non secure mode : learning mode : run mode : test run mode may be tampered with by an attacker. This may lead to a denial of service attack against 1.1 Setting Manager (Special Face ID) or an elevation of privilege attack against 1.1 Setting Manager (Special Face ID) or an information disclosure by 1.1 Setting Manager (Special Face ID). 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.	Critical	5	9	7	9	3	5	9	3	6.25	High	9	9	5	7



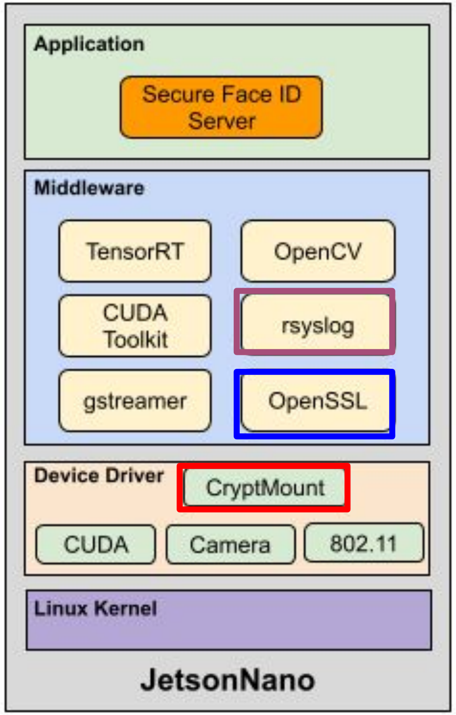
5verflow (Team 5) Security Requirements

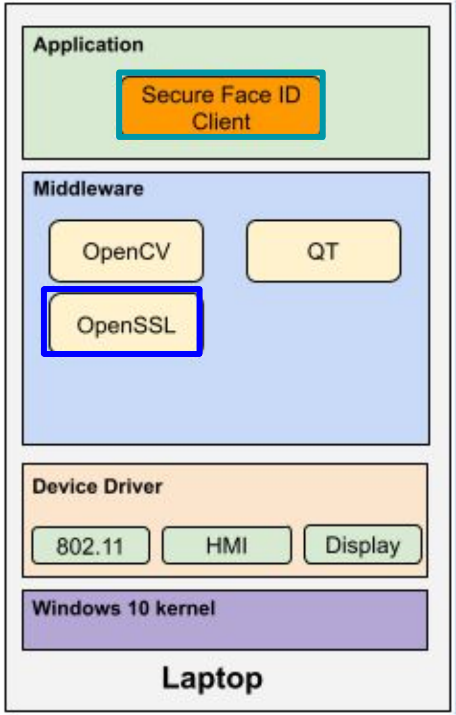
Category	Security Requirements ID	Security Requirements	TID	Threat	Notes
Input Validation for Client Application	SR 1-1	Client Application must check if the format of input IP address is in valid format	170	attacker can TAMPER the IP address input to extremly long characters that might causes buffer overflow. This attack might break the system or simply leads to DENIAL OF SERVICE	Addressing malformed User Input of IP address. This SR does not address an malicious IP address within a valid range. We categorized that kind of attack into Spoofing, and thus can be handled by secure authentication (SR 3-1).
	SR 1-2	Server and Client should check respectively whether the input for Username field on the Register mode is valid as a filename.	170	An attacker can cause buffer overflow using a very long filename as the input or inability to save a file using special characters	Even on the non-secure mode, the input validation check for filename should be conducted.
	SR 1-3	Client should check if the input of the Port field is within the valid port number range.	170	An attacker can write a very large number or string text at the input of Port field and it can cause buffer overflow.	We need to check whether the input is a type of integer and is within the valid port number to mitigate the risk of wrong inputs.
	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.	170	An attacker can cause buffer overflow using a very long filename as the input, and can write a file name which is not a video file format to excute or store malicious binary file.	Even on the non-secure mode, the input validation check for filename should be conducted.
	SR 1-5	Client should check whether the image recevied from server is format of jpeg before displaying it.	N/A	An attacker can modify data which is transmitted from server. A header of jpeg format can be compromised.	If a jpeg header is attacked, the image cannot be displayed using openCV or even any other libraries. Simply, we can check SOI (start of image) byte for jpeg format.
	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.	N/A	By tempering of an attacker, the number of detected face cannot be matched to the number of face information.	We will send a face information for an image at server as follows: - Number of detected faces - Face area and username for each deetected faces
Secure Data Transmission	SR 2-1	after connection establishment all the data transferred between server and client must be securely encrypted	N/A	INFORMATION DISCLOSURE of data over the network	mitigation strategy: TLS applied only when the application is running on Secure Mode
	SR 2-2	must check integrity of all the transmitted data	N/A	TAMPERING of the data over the	mitigation strategy: TLS applied only when the application is running on

<Security Risk Level>

<Security Requirement>

Secure Design





🟧 Write invalid form of IP address (ex. 123.456.789)
- ➡ Input Validation : Input Data Verifier

🟦 Sniffing data on network between JetsonNano and user laptop
- ➡ Data Encryption : OpenSSL v1.1

🟦 Connection from unknown client
- ➡ Authentication : Key from trusted certificate authority (JetsonNano)

🟥 Secure Storage
- ➡ cryptmount

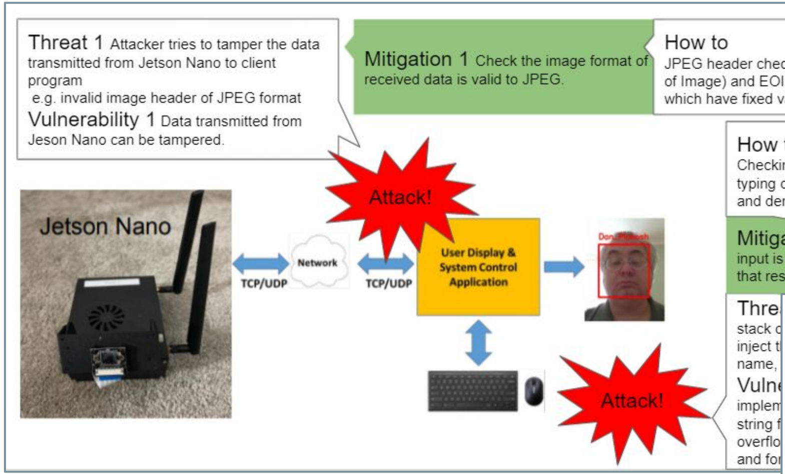
🟪 Logging
- ➡ rsyslog

Software Architect Design Document : https://docs.google.com/document/d/1NJaph-tavyyWxa13gYBciuc1nuE-puGsy08pcuRcfRk

Secure Design

Phase 1 Review

Input Validation



Threat 1 Attacker tries to tamper the data transmitted from Jetson Nano to client program
e.g. invalid image header of JPEG format

Vulnerability 1 Data transmitted from Jeson Nano can be tampered.

Mitigation 1 Check the image format of received data is valid to JPEG.

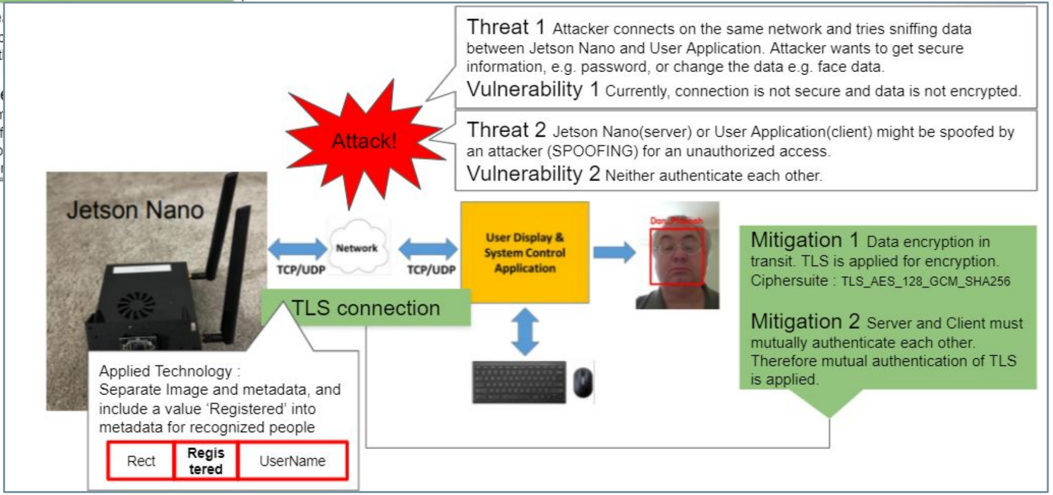
How to
JPEG header check by parsing SOI (Start of Image) and EOI (End Of Image) bytes which have fixed values.

How to
Checking the input validity while in typing on the edit box of client program and deny input when violate rules

Mitigation 2 Application checks if input is valid or not and use functions that restrict the number of bytes

Threa
stack o
inject t
name,
Vulne
implem
string f
overflo
and fo

Data Transmission



Threat 1 Attacker connects on the same network and tries sniffing data between Jetson Nano and User Application. Attacker wants to get secure information, e.g. password, or change the data e.g. face data.

Vulnerability 1 Currently, connection is not secure and data is not encrypted.

Threat 2 Jetson Nano(server) or User Application(client) might be spoofed by an attacker (SPOOFING) for an unauthorized access.

Vulnerability 2 Neither authenticate each other.

Mitigation 1 Data encryption in transit. TLS is applied for encryption.
Ciphersuite : TLS_AES_128_GCM_SHA256

Mitigation 2 Server and Client must mutually authenticate each other. Therefore mutual authentication of TLS is applied.

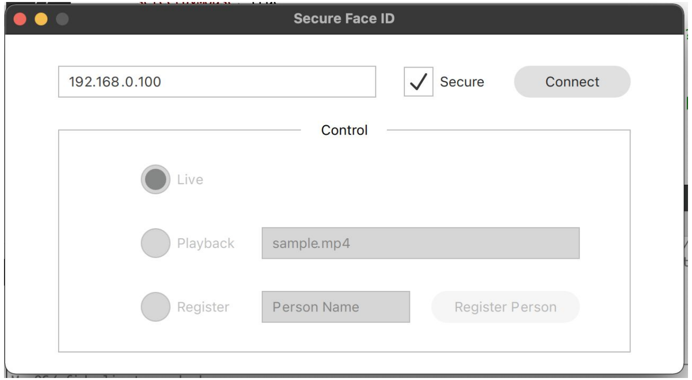
Applied Technology :
Separate Image and metadata, and include a value 'Registered' into metadata for recognized people

Rect	Registered	UserName

Implementation & Verification

Phase 1 Review

<Client>



Secure Face ID

192.168.0.100 ☑ Secure

Connect

Control

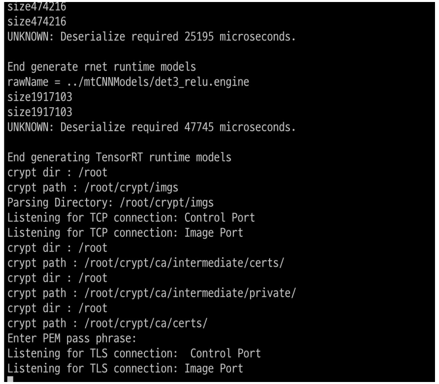
● Live

○ Playback sample.mp4

○ Register Person Name

Register Person

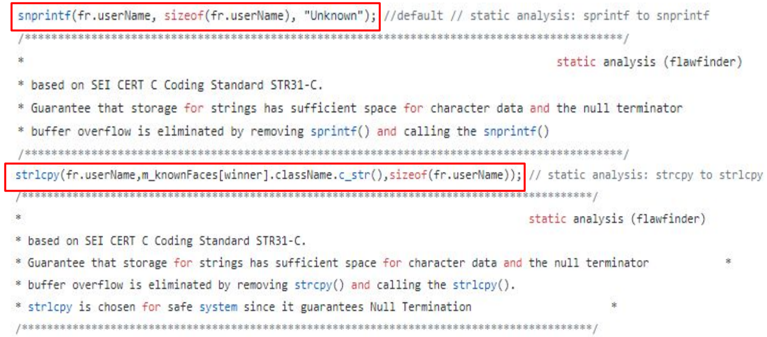
<Server>



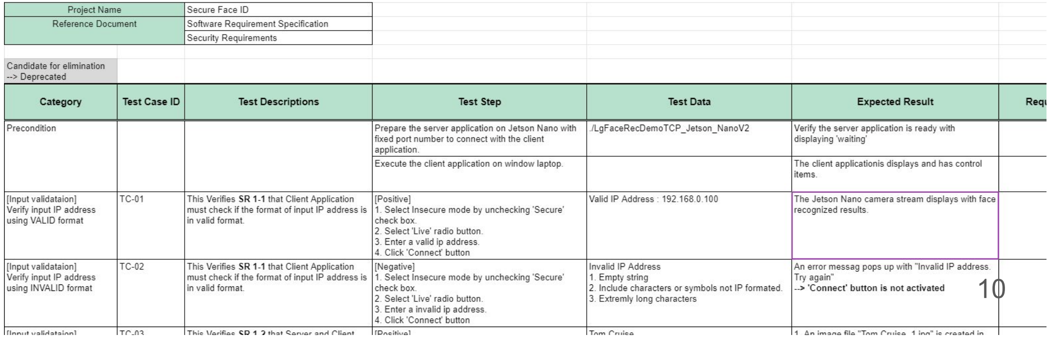
```
size474216
size474216
UNKNOWN: Deserialize required 25195 microseconds.

End generate rnet runtime models
rawName = ../mtCNNModels/det3_relu.engine
size1917103
size1917103
UNKNOWN: Deserialize required 47745 microseconds.

End generating TensorRT runtime models
crypt dir : /root
crypt path : /root/crypt/imgs
Parsing Directory: /root/crypt/imgs
Listening for TCP connection: Control Port
Listening for TCP connection: Image Port
crypt dir : /root
crypt path : /root/crypt/ca/intermediate/certs/
crypt dir : /root
crypt path : /root/crypt/ca/intermediate/private/
crypt dir : /root
crypt path : /root/crypt/ca/certs/
Enter PEM pass phrase:
Listening for TLS connection:  Control Port
Listening for TLS connection: Image Port
```

- Secure Coding w/ Static Analysis
  - FlawFinder : 21 issues found -> fixed!

- Verifications w/ 20 Test cases



```
snprintf(fr.userName, sizeof(fr.userName), "Unknown"); //default // static analysis: sprintf to snprintf
/*******************************************************************************************/
*                                                                  static analysis (flawfinder)
* based on SEI CERT C Coding Standard STR31-C.
* Guarantee that storage for strings has sufficient space for character data and the null terminator
* buffer overflow is eliminated by removing sprintf() and calling the snprintf()
/*******************************************************************************************/
strlcpy(fr.userName,m_knownFaces[winner].className.c_str(),sizeof(fr.userName)); // static analysis: strcpy to strlcpy
/*******************************************************************************************/
*                                                              static analysis (flawfinder)
* based on SEI CERT C Coding Standard STR31-C.
* Guarantee that storage for strings has sufficient space for character data and the null terminator        *
* buffer overflow is eliminated by removing strcpy() and calling the strlcpy().
* strlcpy is chosen for safe system since it guarantees Null Termination                  *
/*******************************************************************************************/
```



Project Name	Secure Face ID
Reference Document	Software Requirement Specification
	Security Requirements

Candidate for elimination
--> Deprecated

Category	Test Case ID	Test Descriptions	Test Step	Test Data	Expected Result	Requ
Precondition			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'	
			Execute the client application on window laptop.		The client applicationis displays and has control items.	
[Input validataion] Verify input IP address using VALID format	TC-01	This Verifies **SR 1-1** that Client Application must check if the format of input IP address is in valid format.	[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.	
[Input validataion] Verify input IP address using INVALID format	TC-02	This Verifies **SR 1-1** that Client Application must check if the format of input IP address is in valid format.	[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 formated. 3. Extremly long characters	An error messag pops up with "Invalid IP address. Try again" **--> 'Connect' button is not activated**	
[Input validataion]	TC-03	This Verifies **SR 1.2** that Server and Client	[Positive]	Tom Cruise	1. An image file "Tom Cruise_1.jpg" is created in	

Phase 2

Security Analysis of Classmate System

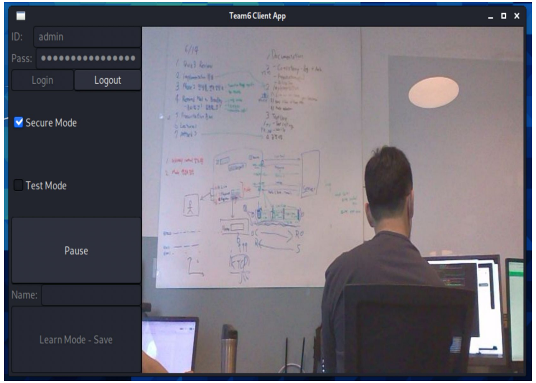
Introduction of Team 6’s System

Phase 2

Our review from the presentation of phase 1 and the given artifacts

- Assumes that only one ID/PW is allowed and faces for ID are pre-registered.
- 2 factor authentication is used to log in (ID/PW, face)
- All the faces are stored with encryption. even the filename is hashed value as well.
- Communicate using Encrypted channel with TLS protocol.

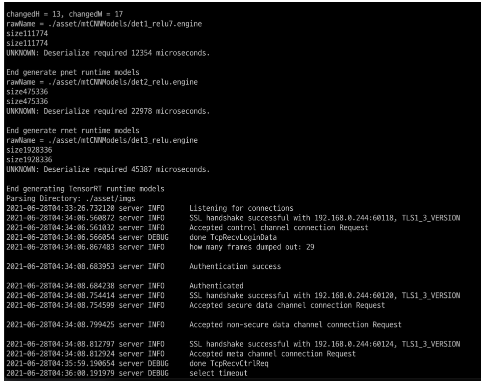
Team 6’s Client UI



ID: admin
Pass: ●●●●●●●●●●●●●●●●
☑ Secure Mode
□ Test Mode
Name:

<Client>

Server is always printing the log



```
changedH = 13, changedW = 17
rawName = ./asset/mtCNNModels/det1_relu7.engine
size111774
size111774
UNKNOWN: Deserialize required 12354 microseconds.

End generate pnet runtime models
rawName = ./asset/mtCNNModels/det2_relu.engine
size475336
size475336
UNKNOWN: Deserialize required 22978 microseconds.

End generate rnet runtime models
rawName = ./asset/mtCNNModels/det3_relu.engine
size1928336
size1928336
UNKNOWN: Deserialize required 45387 microseconds.

End generating TensorRT runtime models
Parsing Directory: ./asset/imgs
2021-06-28T04:33:26.732120 server INFO      Listening for connections
2021-06-28T04:34:06.560872 server INFO      SSL handshake successful with 192.168.0.244:60118, TLS1_3_VERSION
2021-06-28T04:34:06.561032 server INFO      Accepted control channel connection Request
2021-06-28T04:34:06.566054 server DEBUG     done TcpRecvLoginData
2021-06-28T04:34:06.867483 server INFO      how many frames dumped out: 29

2021-06-28T04:34:08.683953 server INFO      Authentication success

2021-06-28T04:34:08.684238 server INFO      Authenticated
2021-06-28T04:34:08.754414 server INFO      SSL handshake successful with 192.168.0.244:60120, TLS1_3_VERSION
2021-06-28T04:34:08.754599 server INFO      Accepted secure data channel connection Request

2021-06-28T04:34:08.799425 server INFO      Accepted non-secure data channel connection Request

2021-06-28T04:34:08.812797 server INFO      SSL handshake successful with 192.168.0.244:60124, TLS1_3_VERSION
2021-06-28T04:34:08.812924 server INFO      Accepted meta channel connection Request
2021-06-28T04:35:59.190654 server DEBUG     done TcpRecvCtrlReq
2021-06-28T04:36:00.191979 server DEBUG     select timeout
```

<Server>

Found Vulnerabilities - Categorized by Methods

Phase 2

Method	# of founding	Takeaway
Manual Code Review	17	Source code has full and correct information about the software Good to find security holes for attackers accustomed to programming languages Risks on this method are: ● Illegible code might be almost impossible to read ● Takes time to understand without documents
Documents	8	Well-organized information to the program Security-related documents are useful to find out what have been neglected. Risks on this method are: ● If a document is not written well or not updated for the latest commit of source code, it might misguide the reader.
Tests	4	Developers have already run several tests - using another tools is recommended Requires knowledge and environment setups for testing Risks on this method are: ● Takes time for preparations
Total	29	

Found Vulnerabilities - Categorized by CIA triad

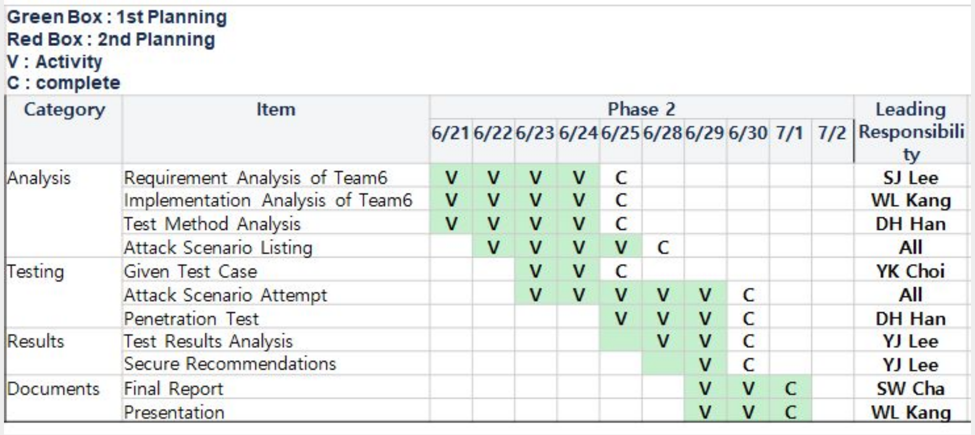
Phase 2

CIA	# of founding	Takeaway
Availability	16	Most of the exploitable vulnerabilities are to harm availability as ● Just breaking a software does not require full understanding of it ● The attacks are out of coverage the application handles (OS or router’s role), from our experience in phase 1
Integrity	5	Found some vulnerabilities, but most of them failed since ● Data on transmission is protected by TLS ● Data in filesystem is protected by encryption (hashed)
Complex	5	Some of the vulnerabilities affect multiple components. E.g., weak SSH ID and password leads to penetration that enables numerous attacks
Confidentiality	3	It was hard to find out exploitable vulnerabilities due to ● TLS covers many vulnerabilities related to confidentiality ● two-factor authentication
Total	29	

Red Team Activities

Phase 2

Schedule



Green Box : 1st Planning
Red Box : 2nd Planning
V : Activity
C : complete

Category	Item	Phase 2										Leading Responsibility
		6/21	6/22	6/23	6/24	6/25	6/28	6/29	6/30	7/1	7/2	
Analysis	Requirement Analysis of Team6	V	V	V	V	C						SJ Lee
	Implementation Analysis of Team6	V	V	V	V	C						WL Kang
	Test Method Analysis	V	V	V	V	C						DH Han
	Attack Scenario Listing		V	V	V	V	C					All
Testing	Given Test Case			V	V	C						YK Choi
	Attack Scenario Attempt			V	V	V	V	V	C			All
	Penetration Test					V	V	V	C			DH Han
Results	Test Results Analysis						V	V	C			YJ Lee
	Secure Recommendations							V	C			YJ Lee
Documents	Final Report							V	V	C		SW Cha
	Presentation							V	V	C		WL Kang

Team Organization

Name	Role (Phase 2)
SeungWook Cha	Team Lead
SungJun Lee	Document Analysis
DongHyuk Han	Document Analysis & Secure Testing
WooLam Kang	Implement Analysis & Secure Testing
YooKyoung Choi	Document Analysis & Secure Testing
YoungJinn Lee	Implement Analysis & Secure Testing

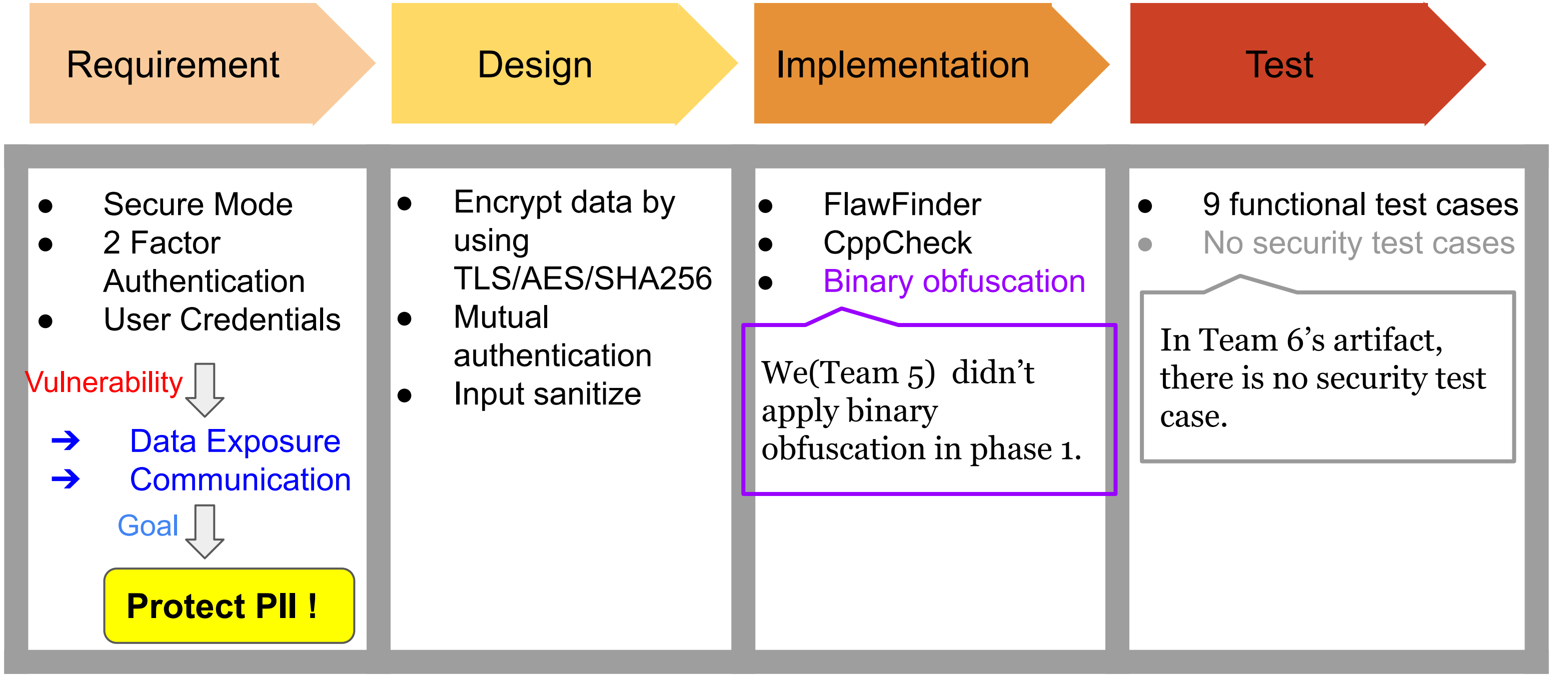
Activity

1. Analyze documents
   a. Threat modeling
   b. Security requirement
   c. Design document
   d. Static analysis result
2. Conduct test cases
   a. Given test cases
   b. Additional exploring test
3. Code reviews
   a. Write sequence diagram
   b. Secure perspectives
4. Discuss attack scenario (Periodically)
5. Do attack
   a. based on our vulnerability hypothesis
6. Wrap Up

Analysis : Team 6’s Development Process

Phase 2

Security items found in Team 6’s docs → Discussion points



Requirement

- Secure Mode
- 2 Factor Authentication
- User Credentials

Vulnerability ⇩

→ Data Exposure

→ Communication

Goal ⇩

Protect PII !

Design

- Encrypt data by using TLS/AES/SHA256
- Mutual authentication
- Input sanitize

Implementation

- FlawFinder
- CppCheck
- Binary obfuscation

We(Team 5) didn’t apply binary obfuscation in phase 1.

Test

- 9 functional test cases
- No security test cases

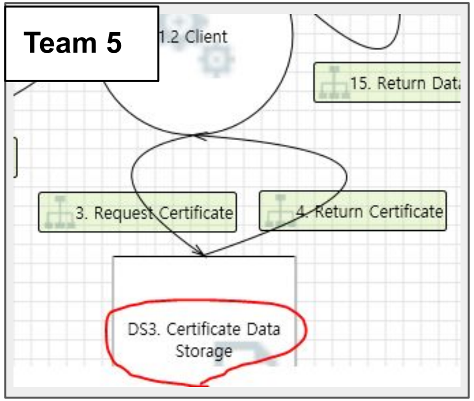
In Team 6’s artifact, there is no security test case.

* PII : Personally Identifiable Information

+ Violet : Team 5’s missed item
- Gray : Team 6’s missed item

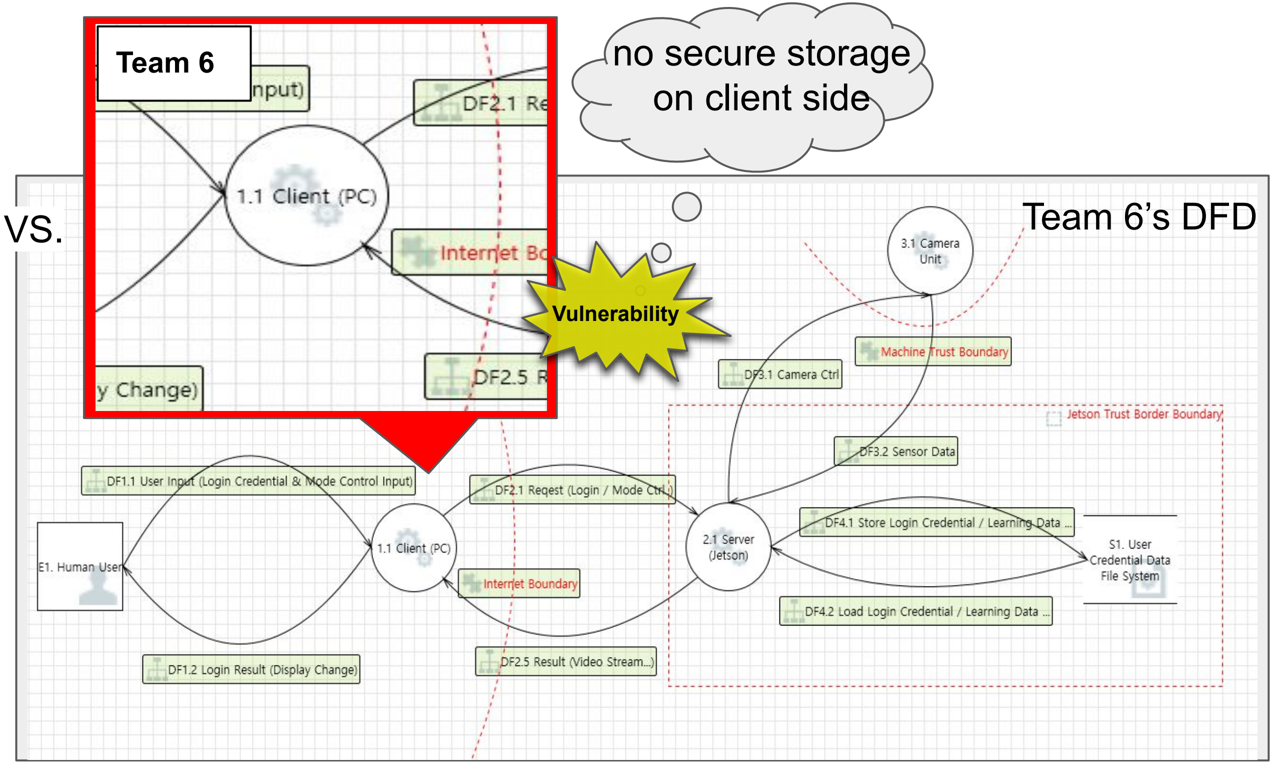
Analysis : Team 6’s Threat Model

Vulnerability : On the client side, secure storage is not considered.



[This DFD considered the secure storage]

VS.


Team 6
nput)
DF2.1 Re
1.1 Client (PC)
Internet Bo
y Change)
DF2.5 R
Vulnerability
3.1 Camera Unit
Machine Trust Boundary
DF3.1 Camera Ctrl
Jetson Trust Border Boundary
DF3.2 Sensor Data
DF1.1 User Input (Login Credential & Mode Control Input)
DF2.1 Reqest (Login / Mode Ctrl)
DF4.1 Store Login Credential / Learning Data ...
E1. Human User
1.1 Client (PC)
2.1 Server (Jetson)
S1. User Credential Data File System
Internet Boundary
DF4.2 Load Login Credential / Learning Data ...
DF1.2 Login Result (Display Change)
DF2.5 Result (Video Stream...)

no secure storage on client side

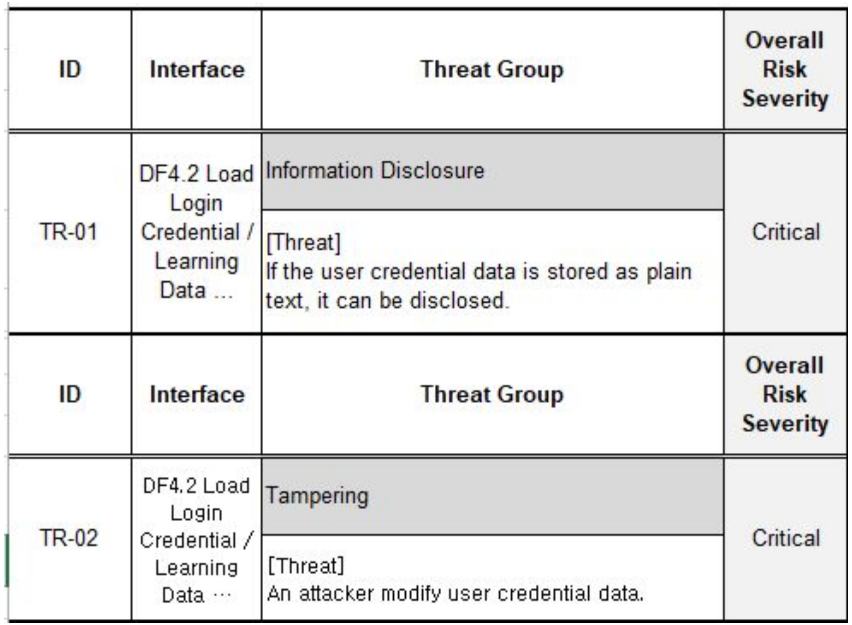
Team 6’s DFD

Analysis : Team 6’s Risk Assessment

Phase 2

‘Input Validation’ was most critical in our case, however, Team 6 Assessment shows ‘User Credential’ as critical and we reviewed focused on it and found that mitigation was applied well.

Next, we’d tried looking for vulnerabilities in the Input Validation area but there was no remarkable result. Input Validation was marked as High in the Team 6 assessment and it looks like well mitigated.



ID	Interface	Threat Group	Overall Risk Severity
TR-01	DF4.2 Load Login Credential / Learning Data ...	Information Disclosure	Critical
		[Threat] If the user credential data is stored as plain text, it can be disclosed.	

ID	Interface	Threat Group	Overall Risk Severity
TR-02	DF4.2 Load Login Credential / Learning Data ···	Tampering	Critical
		[Threat] An attacker modify user credential data.	

Analysis : Static Analysis

Phase 2

FlawFinder ID	Source code path (line)	Target	Vulnerability code	Analysis of Team 5
FF-01	./common/TcpSendRecv.cpp:124	(buffer) memcpy	CWE-120	need mitigation - alloc size of dst
FF-04	./server/src/faceNet.cpp:122	(misc) open	CWE-362	Follow the principle of least privilege when assigning access rights to entities in a software system. Denying access to a file can prevent an attacker from replacing that file with a link to a sensitive file.
FF-08	./server/src/main.cpp:163	(buffer) memcpy	CWE-120	false alarm
FF-09	./common/TcpSendRecv.cpp:99	(buffer) strlen	CWE-126	false alarm the parameter userid((const gchar*) is called with c_str() which always contains null termination

CppCheck ID	Sourcecode path (line)	Type	Analysis of Team 5
CC-01	server/src/main.cpp:196	style [unreadVariable]	false positive
CC-07	server/src/videoStreamer.cpp:35	warning [noCopyConstructor]	false positive - use openCV library
CC-08	server/src/videoStreamer.cpp:35	warning [noOperatorEq]	false positive - use openCV library
CC-09	server/src/videoStreamer.cpp:60	style [unusedFunction]	unused. if not in use, delete it.
CC-13	common/Logger.cpp:124	style [unusedFunction]	false positive

Top 5 violations

Violated Rules	counts	Rule
cppcheck:misra_c_2012_15_01	119	The goto statement should not be used
cppcheck:misra_c_2012_14_04	88	The controlling expression of an if-statement and the controlling expression of an iteration-statement shall have essentially Boolean type
cppcheck:misra_c_2012_15_05	86	A function should have a single point of exit at the end
cppcheck:misra_c_2012_15_06	65	The body of an iteration-statement or a selection-statement shall be a compound statement
cppcheck:misra_c_2012_12_01	59	The precedence of operators within expressions should be made explicit

Team 5 did

1. Run static analysis and Find vulnerabilities.
2. Analyze and Evaluate each item.
3. Suggest mitigations for vulnerabilities.
4. Try to attack !!
   - Encoding data without null character, then restart server. ⇒ Terminated abnormally.

FlawFinder can find

uses of risky functions, buffer overflow (strcpy()), format string ([v][f]printf()), race conditions (access(), chown(), and mktemp()), shell metacharacters (exec()), and poor random numbers (random()).

CppCheck can find

pointer to a variable that goes out of scope, bounds, classes (missing constructors, unused private functions, etc.), exception safety, memory leaks, invalid STL usage, overlapping data in sprintf, division by zero, null pointer dereference, unused struct member, passing parameter by value

Code x-ray

```
if (i){ // non compliant
}
if (i != 0){ // compliant
}
```

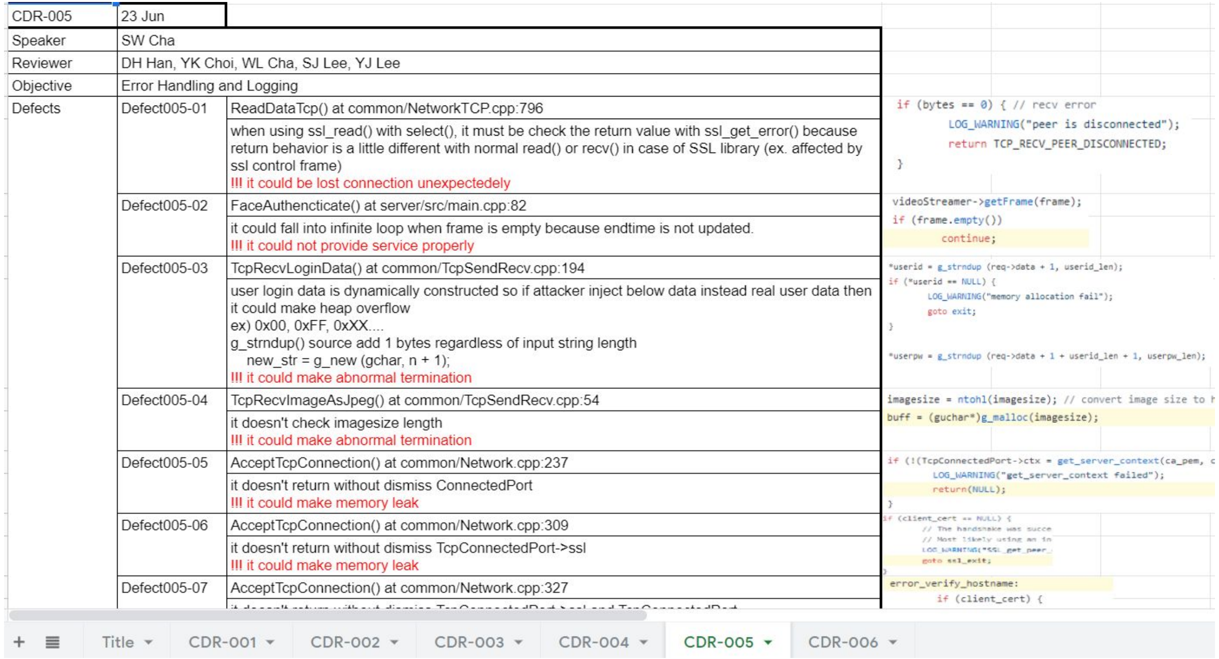
** LG's internal tool (MISRA C 2012)*
** Supports the detection of security vulnerabilities.*
** Compatible with security standards such as CERT, CWE, OWASP, SANS Top 25, OWASP Top 10, and more.*

Analysis : Manual Code Review

Phase 2

We'd conducted a code review on each module categorized by objective.

- File Management
- Input Validation
- Authentication and Password Management
- Session Management
- Error Handling and Logging
- Communication Security



CDR-005	23 Jun		
Speaker	SW Cha		
Reviewer	DH Han, YK Choi, WL Cha, SJ Lee, YJ Lee		
Objective	Error Handling and Logging		

Defects	Defect005-01	ReadDataTcp() at common/NetworkTCP.cpp:796	if (bytes == 0) { // recv error
			LOG_WARNING("peer is disconnected");
			return TCP_RECV_PEER_DISCONNECTED;
			}

		when using ssl_read() with select(), it must be check the return value with ssl_get_error() because return behavior is a little different with normal read() or recv() in case of SSL library (ex. affected by ssl control frame) !!! it could be lost connection unexpectedely	

	Defect005-02	FaceAuthencticate() at server/src/main.cpp:82	videoStreamer->getFrame(frame);
			if (frame.empty())
			continue;

		it could fall into infinite loop when frame is empty because endtime is not updated. !!! it could not provide service properly	

	Defect005-03	TcpRecvLoginData() at common/TcpSendRecv.cpp:194	*userid = g_strndup (req->data + 1, userid_len);
			if (*userid == NULL) {
			LOG_WARNING("memory allocation fail");
			goto exit;
			}
			*userpw = g_strndup (req->data + 1 + userid_len + 1, userpw_len);

		user login data is dynamically constructed so if attacker inject below data instead real user data then it could make heap overflow ex) 0x00, 0xFF, 0xXX.... g_strndup() source add 1 bytes regardless of input string length new_str = g_new (gchar, n + 1); !!! it could make abnormal termination	

	Defect005-04	TcpRecvImageAsJpeg() at common/TcpSendRecv.cpp:54	imagesize = ntohl(imagesize); // convert image size to h
			buff = (guchar*)g_malloc(imagesize);

		it doesn't check imagesize length !!! it could make abnormal termination	

	Defect005-05	AcceptTcpConnection() at common/Network.cpp:237	if (!(TcpConnectedPort->ctx = get_server_context(ca_pem, c
			LOG_WARNING("get_server_context failed");
			return(NULL);
			}

		it doesn't return without dismiss ConnectedPort !!! it could make memory leak	

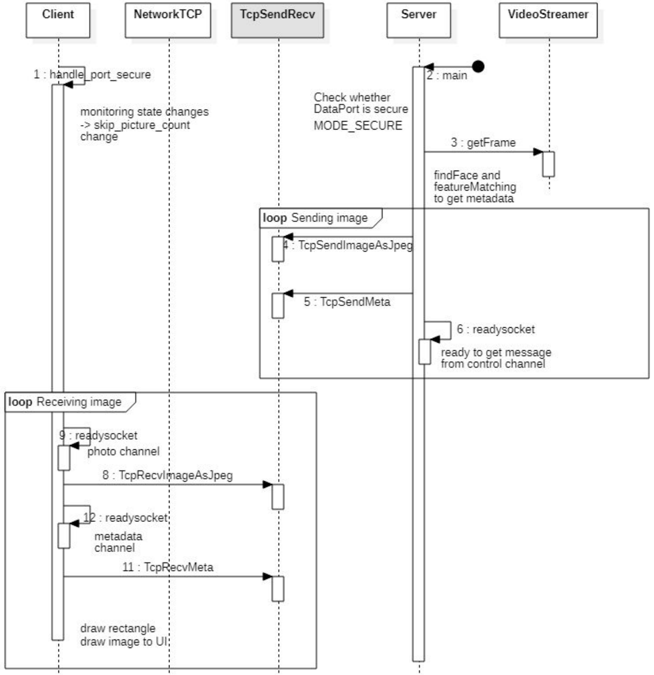
	Defect005-06	AcceptTcpConnection() at common/Network.cpp:309	if (client_cert == NULL) {
			// The handshake was succe
			// Most likely using an in
			LOG_WARNING("SSL_get_peer_
			goto ssl_exit;
			}

		it doesn't return without dismiss TcpConnectedPort->ssl !!! it could make memory leak	
	Defect005-07	AcceptTcpConnection() at common/Network.cpp:327	error_verify_hostname: if (client_cert) {
		it doesn't return without dismiss TcpConnectedPort->ssl and TcpConnectedPort	

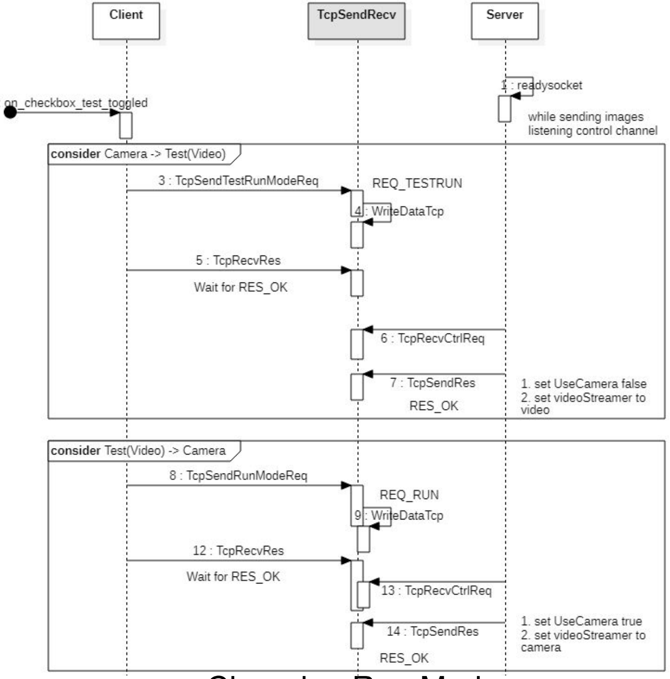
https://docs.google.com/spreadsheets/d/1sCAtsdGnupTEm-SJymMjXAJZx7G2cgQ5aBmfEt2TdWU/edit#gid=1988357498

Analysis : Manual Code Review (Sequence Diagram)

Deep analysis of code in sequence diagrams helped find vulnerabilities. e.g. attack case 1~4



<Data Communications>



<Changing Run Mode>

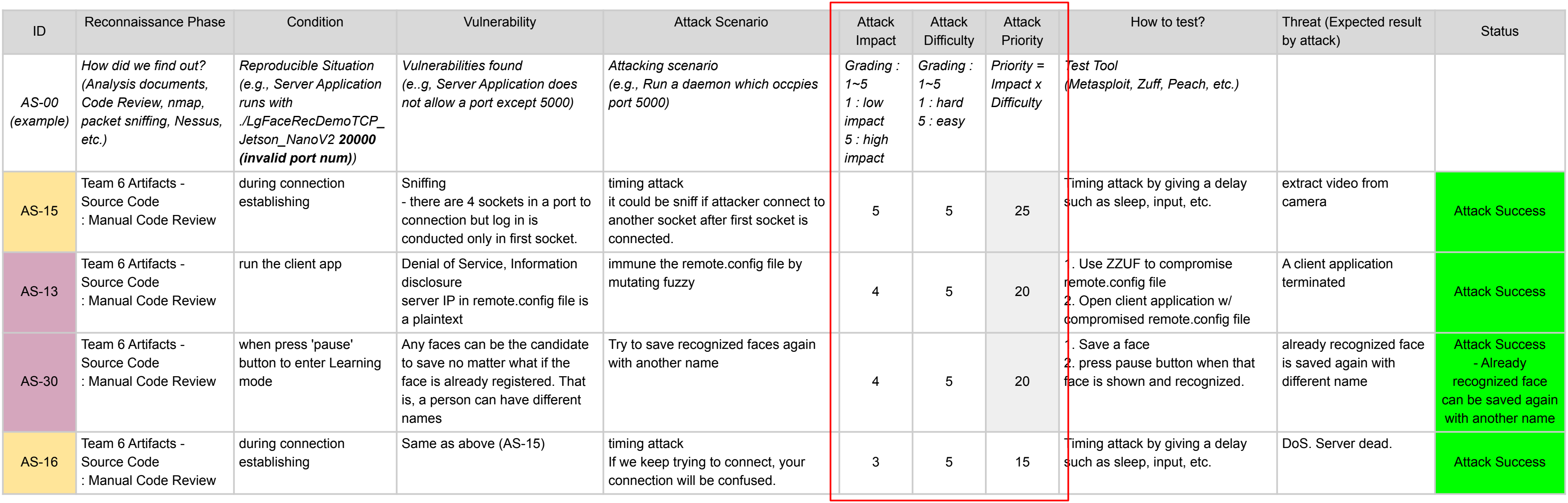
Attack Scenario

Phase 2

Attack Scenario - 11 of 29 scenarios are tested

Assessment using two factors

- Attack Impact - Private info. is the most important
- Attack Difficulty - How easy to try

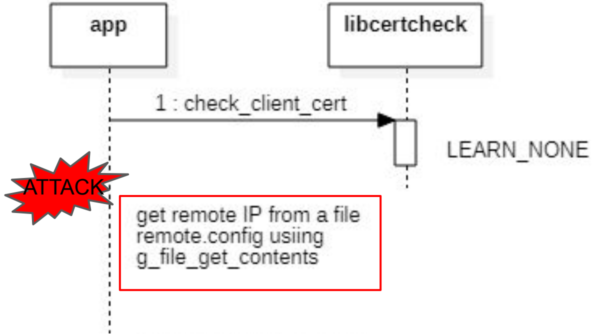


ID	Reconnaissance Phase	Condition	Vulnerability	Attack Scenario	Attack Impact	Attack Difficulty	Attack Priority	How to test?	Threat (Expected result by attack)	Status
AS-00 (example)	*How did we find out? (Analysis documents, Code Review, nmap, packet sniffing, Nessus, etc.)*	*Reproducible Situation (e.g., Server Application runs with ./LgFaceRecDemoTCP_Jetson_NanoV2* ***20000 (invalid port num)****)*	*Vulnerabilities found (e..g, Server Application does not allow a port except 5000)*	*Attacking scenario (e.g., Run a daemon which occpies port 5000)*	*Grading : 1~5 1 : low impact 5 : high impact*	*Grading : 1~5 1 : hard 5 : easy*	*Priority = Impact x Difficulty*	*Test Tool (Metasploit, Zuff, Peach, etc.)*		
AS-15	Team 6 Artifacts - Source Code : Manual Code Review	during connection establishing	Sniffing - there are 4 sockets in a port to connection but log in is conducted only in first socket.	timing attack it could be sniff if attacker connect to another socket after first socket is connected.	5	5	25	Timing attack by giving a delay such as sleep, input, etc.	extract video from camera	Attack Success
AS-13	Team 6 Artifacts - Source Code : Manual Code Review	run the client app	Denial of Service, Information disclosure server IP in remote.config file is a plaintext	immune the remote.config file by mutating fuzzy	4	5	20	1. Use ZZUF to compromise remote.config file 2. Open client application w/ compromised remote.config file	A client application terminated	Attack Success
AS-30	Team 6 Artifacts - Source Code : Manual Code Review	when press 'pause' button to enter Learning mode	Any faces can be the candidate to save no matter what if the face is already registered. That is, a person can have different names	Try to save recognized faces again with another name	4	5	20	1. Save a face 2. press pause button when that face is shown and recognized.	already recognized face is saved again with different name	Attack Success - Already recognized face can be saved again with another name
AS-16	Team 6 Artifacts - Source Code : Manual Code Review	during connection establishing	Same as above (AS-15)	timing attack If we keep trying to connect, your connection will be confused.	3	5	15	Timing attack by giving a delay such as sleep, input, etc.	DoS. Server dead.	Attack Success

Attack Scenario 1 - Server IP Address stored in Plaintext

Vulnerability:

The server IP written in the remote.config file is plaintext. This file is read during client app initialization.


app
libcertcheck
1 : check_client_cert
LEARN_NONE
ATTACK
get remote IP from a file remote.config usiing g_file_get_contents

Attack Scenario (how to test):

Compromising remote.config file

1. Use ZZUF to compromise remote.config file

```
$ cat remote.config
192.168.0.100
$ ./zzuf -r0.05 cat remote.config
182.16|.4�1p
```

2. Open client application with compromised remote.config file

Attack Result:

Client blocked connections by input validation for IP address (`g_hostname_is_ip_address`)

```
$ ./client
2021-06-24T19:49:35.029507 client WARNING   not valid ipaddr in
remote.config file 182.16|.4\2561p\021
```

Recommended Mitigation:

- Store config file in secure storage
- Lock server ip address with encryption

Attack Scenario 2 - Video Sniffing without Authentication

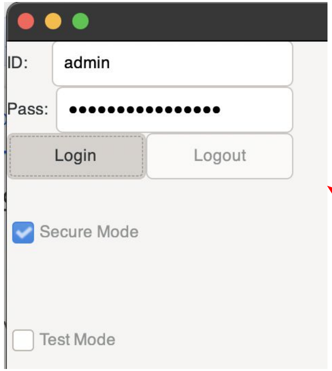
Phase 2

Vulnerability:
Though 4 socket connections are used, the only first connection has the authentication process.

Attack Scenario (how to test):
Timing Attack
(Immediately after normal login of control socket,
 try to connect 3 socket except for control socket)

Attack Result:
The attacker can sniff the video

Recommended Mitigation:
Use same session key over 4 connection
Authenticate every socket connection

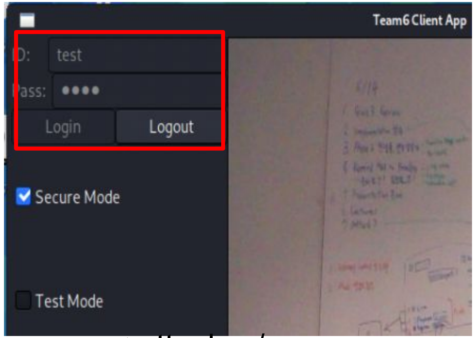


ID: admin
Pass: ••••••••••••••••

☑ Secure Mode

□ Test Mode

< normal login >

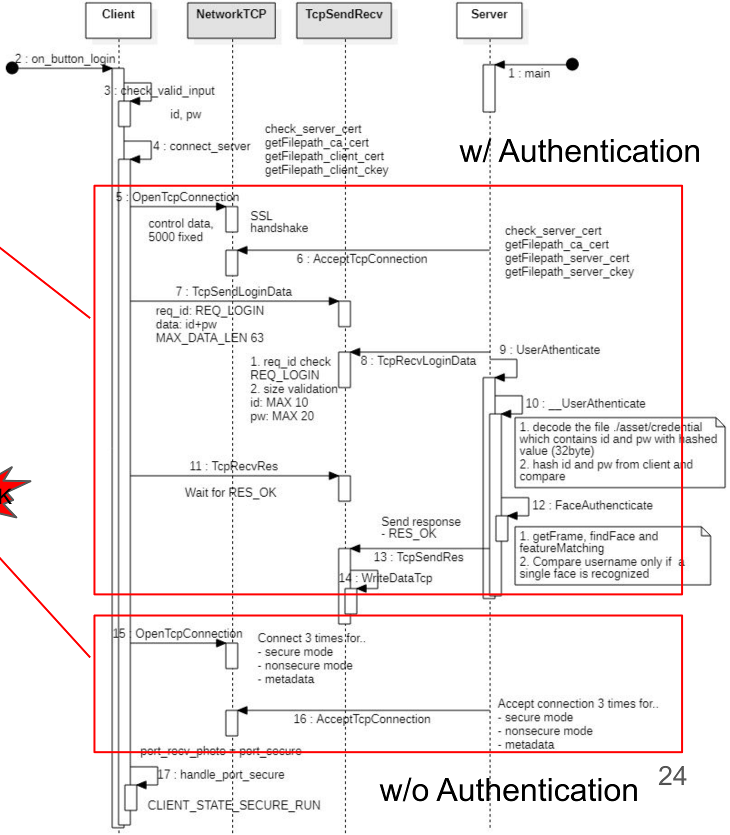


ID: test
Pass: ••••

☑ Secure Mode

□ Test Mode

< attack w/o authentication >



w/ Authentication

w/o Authentication

Attack Scenario 3 - A face with Various Names

Phase 2

Vulnerability:

A person can have different names on learning mode

Attack Scenario (how to test):

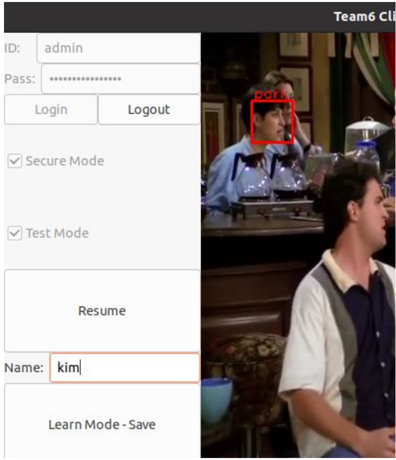
1. Save a face to a new name ‘park’
2. Press pause button when that face is shown and recognized.
3. Save it with another name ‘kim’

Attack Result:

The same face is recognized alternately with different names ‘park’ and ‘kim’ depending on the conditions such as angle, lighting.

Recommended Mitigation:

Add a logic NOT to allow another name for already recognized face



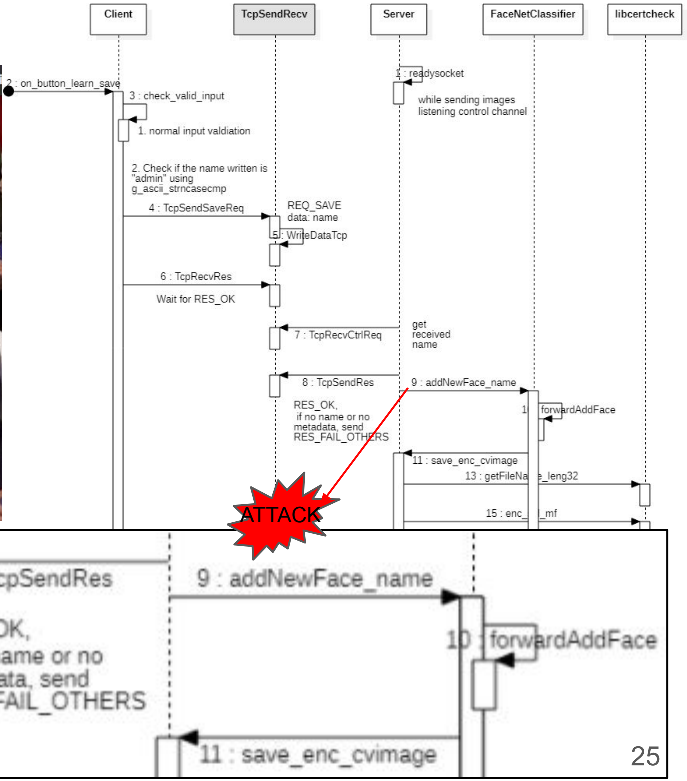
ID: admin

Pass: ••••••••••••••••

☑ Secure Mode

☑ Test Mode

Name: kim



Attack Scenario 4 - Authentication Bypassing

Vulnerability:
ID/PW are checked only one time,
so one-block detour enables to avoid authentication

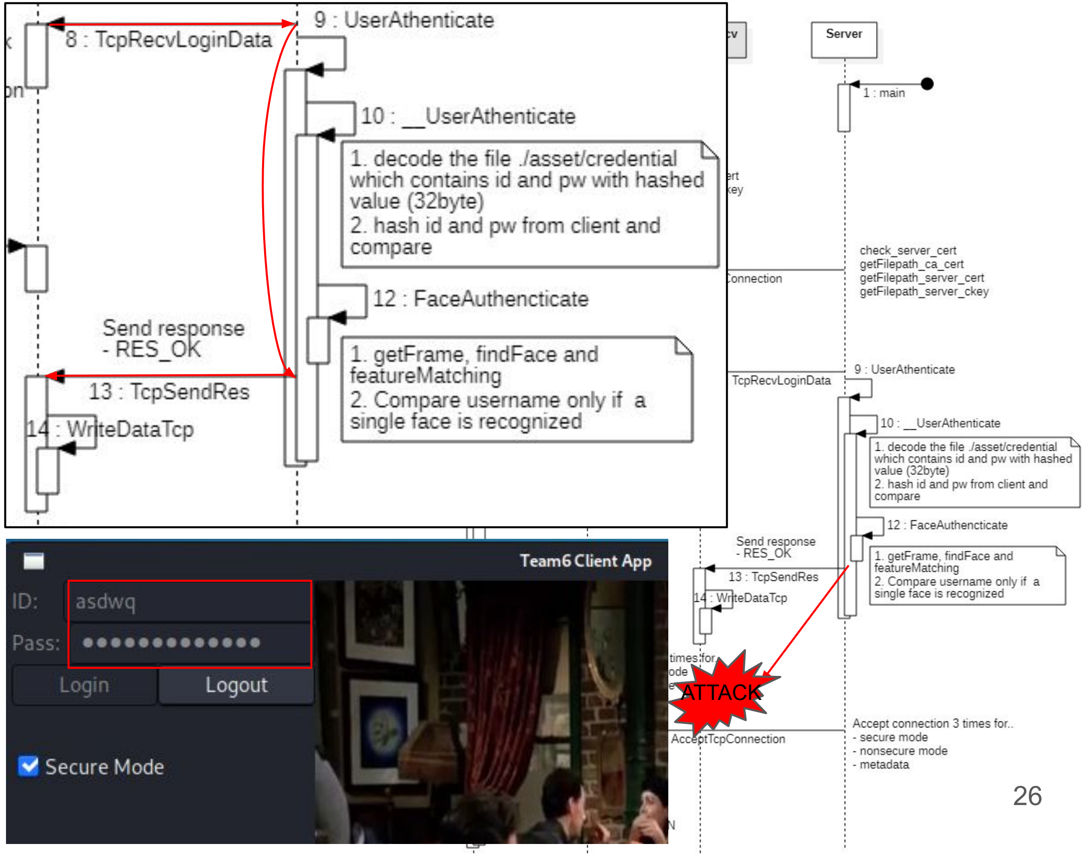
Attack Scenario (how to test):

1. Disassemble the server software using rizin
2. Find out the authentication function and modify it to always return true

Attack Result:
Unauthorized users are able to access the system

Recommended Mitigation:

1. Repeatedly use authentication credentials (e.g., use hashed ID and PW as authentication token and server keeps requesting it for every functionality)



ID: asdwq
Pass: ●●●●●●●●●●●●●●

☑ Secure Mode

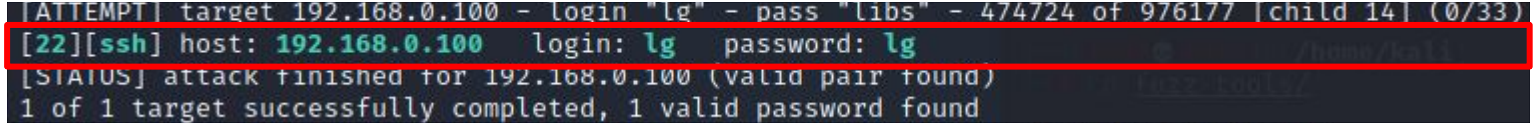
Penetration Test

Phase 2

Vulnerability: Too short and simple User ID and password are easily exploitable

Penetration Method:

1. Metasploit (FAIL)
   a. Able to exploit Rpcbomb to Rpcbind service (DoS)
2. Brute-Force SSH Credentials (SUCCESS)
   a. Take up too much time
   b. Proper dictionary would save a lot of time (success probability ↔ running time trade-off)
   - made assumption that lower-case letters only and short length (20,469 words → 988 words)



```
[ATTEMPT] target 192.168.0.100 - login "lg" - pass "libs" - 474724 of 976177 [child 14] (0/33)
[22][ssh] host: 192.168.0.100   login: lg   password: lg
[STATUS] attack finished for 192.168.0.100 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
```

Potential risks in consequence of penetration are:
1. Confidentiality - access to private data (user images and certificates)
2. Integrity - modify log or program by recompiling or reverse engineering
3. Availability - break the program of file system

Summary

Phase 2

- Analysis of Team 6's security activities
  - Are there any missing threats? -> **Some threats were not derived.**
  - Has the threat been mitigated? -> All derived threats are mitigated.
  - Did new vulnerabilities arise because of mitigation? -> No.
  - Did Team 6 mitigate well-known vulnerabilities? -> Team 6 assumed that known vulnerabilities were mitigated due to the tight schedule and focused on looking for new vulnerabilities.
- Vulnerability Assessment & Evaluation
  - We summarized the vulnerabilities we additionally discovered.
  - We derive the attack scenario, try to attack, and suggest mitigation methods.
- Lesson Learned
  - It is necessary to do these activity to improve security of our system as well.

Lessons Learned

Lessons Learned

1. Proactive security considerations improve security and accelerate the development period.
2. Communicating the ownership and responsibilities of security processes is essential.
3. Collaboration between development and security (blue team and red team) results in higher value work.
4. Various techniques should be utilized to find, evaluate, and mitigate vulnerabilities.

Thank You!
(Q&A)

Contents

- Team Members
- Roadmap
- Phase 1 Review
  - Security Development Life-cycle
  - Requirement
  - Secure Design
  - Implementation
- Phase 2 Security Evaluation of Classmate’s system (Team 6)
  - Found Vulnerabilities
  - Plan
  - Red Team Activities
  - Development Process Overview
  - Analysis of Threat Model
  - Static Analysis
  - Sequence Diagram
  - Attack Scenario
  - Attack Cases
  - Penetration Test
  - Summary
- Lessons Learned

Security Development Life-cycle

Our activity :
User Requirement
Project Requirement
Software Requirement
Security Requirement

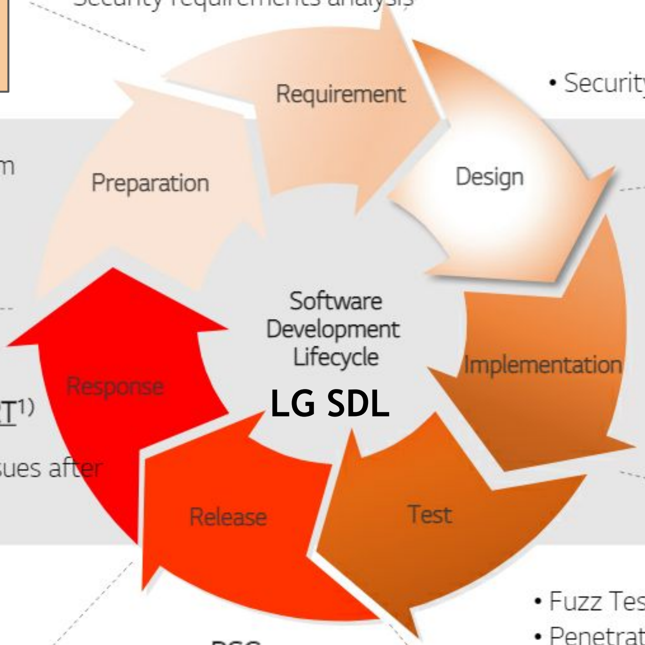
- Define product security level
- the LG-SDL Planning
- Security requirements analysis

Our activity :
Software Architect Design
Quality Attribute
Security Design Review

- Security Design Review

Our activity :
(Education)
Cousera
MOOC
CMU e-lecture
CMU Live Zoom

- Security Education
- LG Certificate system for security experts



- Secure Implementation
- Open Source Vulnerability Analysis
- Security Static Analysis
- Security Functional Testing

Our activity :
Secure Coding
Static Analysis

PSRT1)

- Responding to security issues after product launch

- Fuzz Testing
- Penetration Testing

Phase 2 :
Evaluation of Team 6’s System

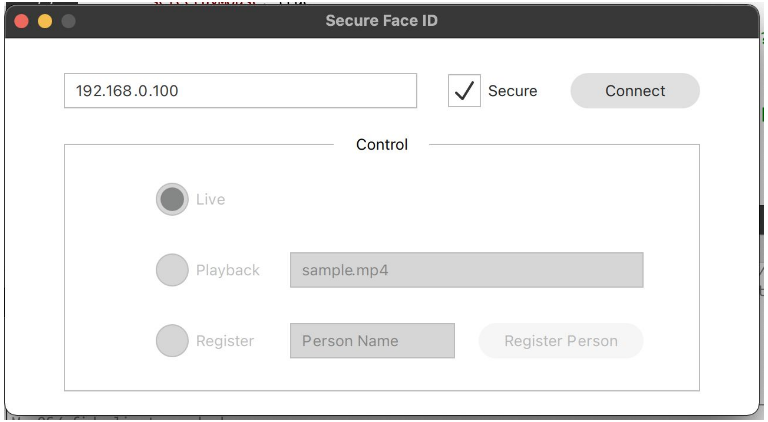
PSC

- LG Electronics Product Security Certification

Implementation

- Client
  - Qt framework
  - OpenSSL

Client UI



Secure Face ID

192.168.0.100 ☑ Secure

Control

● Live

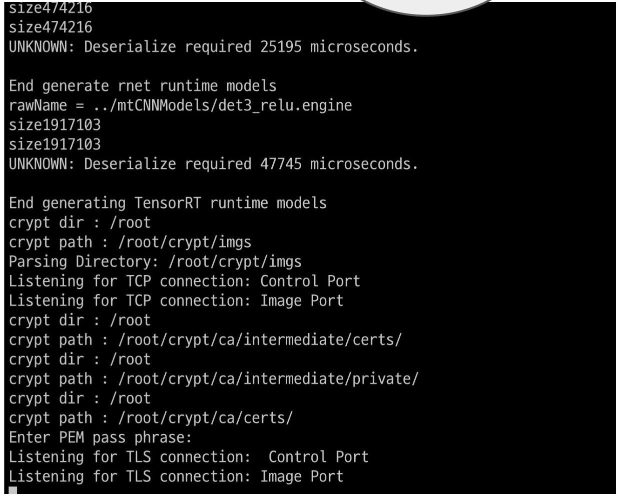
○ Playback sample.mp4

○ Register Person Name

<Client>

- Server
  - OpenSSL
  - rsyslog

Server is always printing the log



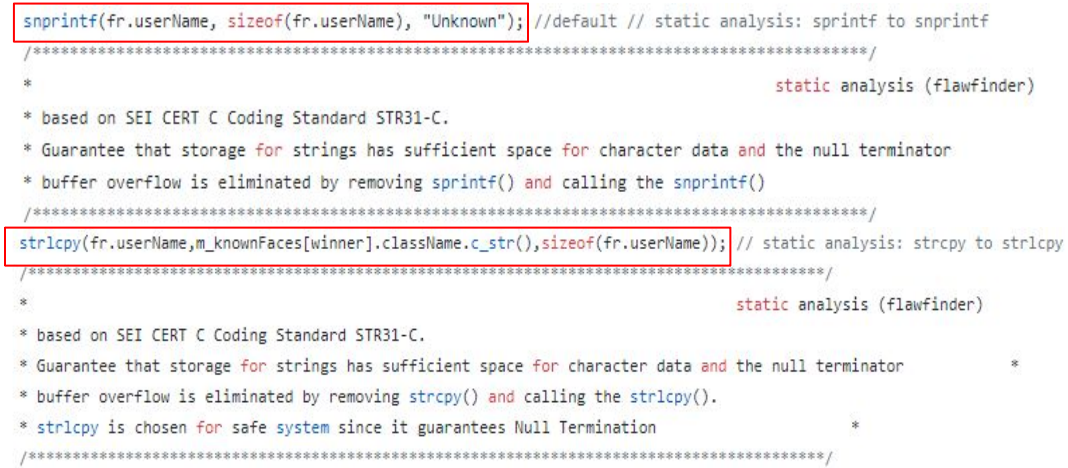
```
size474216
size474216
UNKNOWN: Deserialize required 25195 microseconds.

End generate rnet runtime models
rawName = ../mtCNNModels/det3_relu.engine
size1917103
size1917103
UNKNOWN: Deserialize required 47745 microseconds.

End generating TensorRT runtime models
crypt dir : /root
crypt path : /root/crypt/imgs
Parsing Directory: /root/crypt/imgs
Listening for TCP connection: Control Port
Listening for TCP connection: Image Port
crypt dir : /root
crypt path : /root/crypt/ca/intermediate/certs/
crypt dir : /root
crypt path : /root/crypt/ca/intermediate/private/
crypt dir : /root
crypt path : /root/crypt/ca/certs/
Enter PEM pass phrase:
Listening for TLS connection:  Control Port
Listening for TLS connection: Image Port
```

<Server>

Implementation & Verification

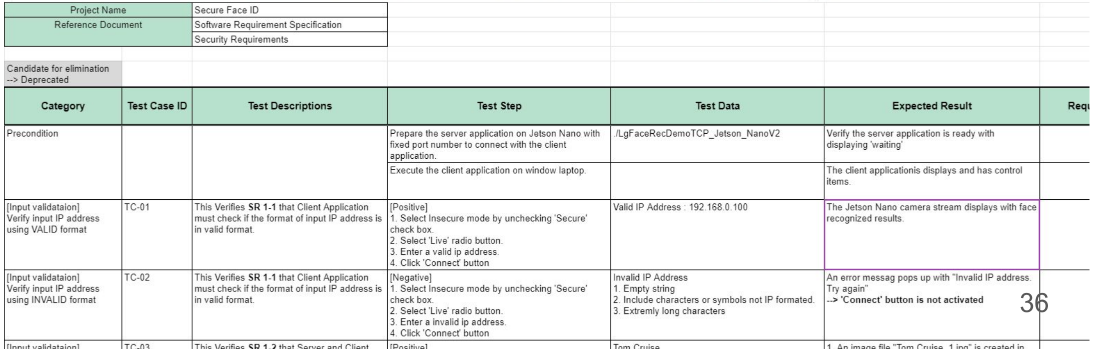
Phase 1 Review

- Secure Coding w/ Static Analysis
  - FlawFinder : 21 issues found
    - ✓ fixed!



```
snprintf(fr.userName, sizeof(fr.userName), "Unknown"); //default // static analysis: sprintf to snprintf
/*****************************************************************************************/
*                                                                 static analysis (flawfinder)
* based on SEI CERT C Coding Standard STR31-C.
* Guarantee that storage for strings has sufficient space for character data and the null terminator
* buffer overflow is eliminated by removing sprintf() and calling the snprintf()
/*****************************************************************************************/
strlcpy(fr.userName,m_knownFaces[winner].className.c_str(),sizeof(fr.userName)); // static analysis: strcpy to strlcpy
/*****************************************************************************************/
*                                                                 static analysis (flawfinder)
* based on SEI CERT C Coding Standard STR31-C.
* Guarantee that storage for strings has sufficient space for character data and the null terminator        *
* buffer overflow is eliminated by removing strcpy() and calling the strlcpy().
* strlcpy is chosen for safe system since it guarantees Null Termination                *
/*****************************************************************************************/
```

- Verifications w/ 20 Test cases



Project Name	Secure Face ID
Reference Document	Software Requirement Specification
	Security Requirements

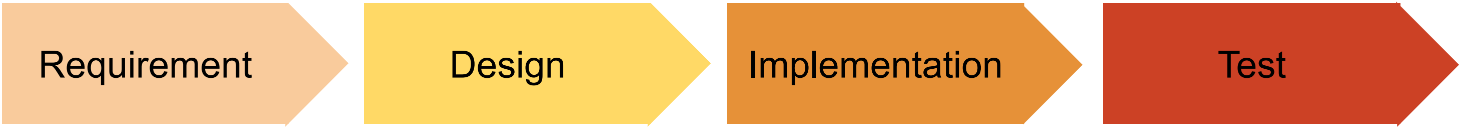
Candidate for elimination
--> Deprecated

Category	Test Case ID	Test Descriptions	Test Step	Test Data	Expected Result	Requ
Precondition			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'	
			Execute the client application on window laptop.		The client applicationis displays and has control items.	
[Input validataion] Verify input IP address using VALID format	TC-01	This Verifies **SR 1-1** that Client Application must check if the format of input IP address is in valid format.	[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.	
[Input validataion] Verify input IP address using INVALID format	TC-02	This Verifies **SR 1-1** that Client Application must check if the format of input IP address is in valid format.	[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 formated. 3. Extremly long characters	An error messag pops up with "Invalid IP address. Try again" **--> 'Connect' button is not activated**	
[Input validataion]	TC-03	This Verifies **SR 1-2** that Server and Client	[Positive]	Tom Cruise	1. An image file "Tom Cruise_1.jpg" is created in	

Team 6 Development Process Overview

Phase 2

Analyzed from Documents



- Secure Mode
- 2 Factor Authentication
- User Credentials

Vulnerability

➢ Data Exposure
➢ Communication

Goal

Protect PII !

- Encrypt data by using TLS/AES/SHA256
- Mutual authentication
- Input sanitize

- Code obfuscation
- FlawFinder
- CppCheck

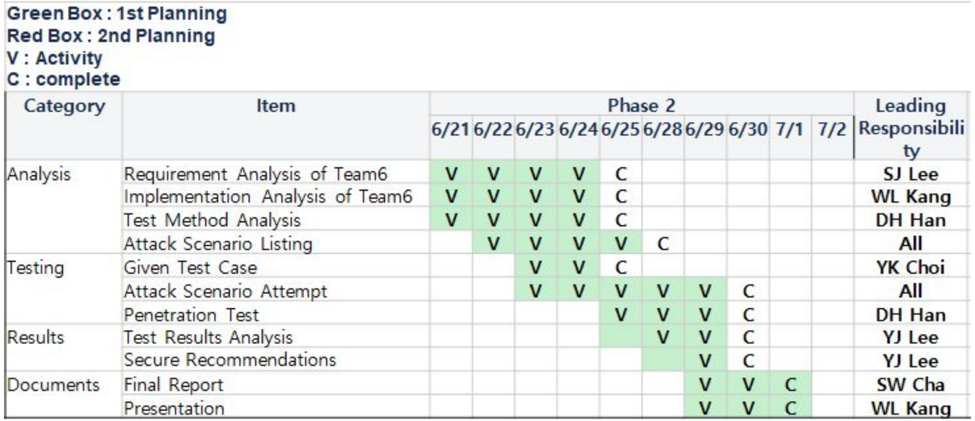
- 9 functional test cases
- secure test cases

Comparison with our artifacts
+ Code obfuscation we didn't
- Secure test cases team 6 didn't

Plan of Red Team Activities

Phase 2

Schedule



Green Box : 1st Planning
Red Box : 2nd Planning
V : Activity
C : complete

Category	Item	Phase 2										Leading Responsibility
		6/21	6/22	6/23	6/24	6/25	6/28	6/29	6/30	7/1	7/2	
Analysis	Requirement Analysis of Team6	V	V	V	V	C						SJ Lee
	Implementation Analysis of Team6	V	V	V	V	C						WL Kang
	Test Method Analysis	V	V	V	V	C						DH Han
	Attack Scenario Listing		V	V	V	V	C					All
Testing	Given Test Case			V	V	C						YK Choi
	Attack Scenario Attempt			V	V	V	V	V	C			All
	Penetration Test					V	V	V	C			DH Han
Results	Test Results Analysis						V	V	C			YJ Lee
	Secure Recommendations							V	C			YJ Lee
Documents	Final Report							V	V	C		SW Cha
	Presentation							V	V	C		WL Kang

Team Organization

Name	Role (Phase 2)
SeungWook Cha	Team Lead
SungJun Lee	Doc. Anal.
DongHyuk Han	Doc. Anal. & Test
WooLam Kang	Impl. Anal. & Test
YooKyoung Choi	Doc. Anal. & Test
YoungJinn Lee	Impl. Anal. & Test

1. Analyze documents
   a. Threat Modeling
   b. Security Requirement
   c. Design Document
2. Conduct test cases
   a. Given test cases
   b. Additional exploring test
3. Code reviews
   a. Write Sequence Diagram
   b. Secure Perspectives
4. Discuss Attack scenario
5. Do Attack
   a. based on our vulnerability hypothesis
6. Wrap Up

Static Analysis

Analysis to evaluate and recommend mitigations

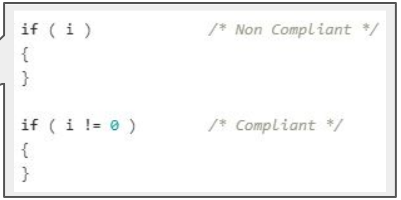
FlawFinder ID	Sourcecode path (line)	Target	Vulnerability code	Analysis of Team 5
FF-01	./common/TcpSendRecv.cpp:124	(buffer) memcpy	CWE-120	need mitigation - alloc size of dst
FF-02	./common/TcpSendRecv.cpp:129	(buffer) memcpy	CWE-120	need mitigation - alloc size of dst
FF-03	./common/TcpSendRecv.cpp:466	(buffer) memcpy	CWE-120	need mitigation - alloc size of dst
FF-04	./server/src/faceNet.cpp:122	(misc) open	CWE-362	Follow the principle of least privilege when assigning access rights to entities in a software system. Denying access to a file can prevent an attacker from replacing that file with a link to a sensitive file.
...	...	...	...	...
FF-08	./server/src/main.cpp:163	(buffer) memcpy	CWE-120	false alarm
FF-09	./common/TcpSendRecv.cpp:99	(buffer) strlen	CWE-126	false alarm the parameter userid((const gchar*) is called with c_str() which always contains null termination
...	...	...	...	...

<FlawFinder>

CppCheck ID	Sourcecode path (line)	Type	Analysis of Team 5
CC-01	server/src/main.cpp:196	style [unreadVariable]	false positive
CC-02	server/src/main.cpp:201	style [unreadVariable]	false positive
CC-03	server/src/main.cpp:209	style [unreadVariable]	false positive
...	...	...	...
CC-07	server/src/videoStreamer.cpp:35	warning [noCopyConstructor]	false positive - use openCV library
CC-08	server/src/videoStreamer.cpp:35	warning [noOperatorEq]	false positive - use openCV library
CC-09	server/src/videoStreamer.cpp:60	style [unusedFunction]	unused. if not in use, delete it.
CC-10	server/src/common.cpp:22	style [unusedFunction]	unused. if not in use, delete it.
CC-11	server/src/faceNet.cpp:414	style [unusedFunction]	unused. if not in use, delete it.
CC-12	common/Logger.cpp:110	style [unusedFunction]	unused. if not in use, delete it.
CC-13	common/Logger.cpp:124	style [unusedFunction]	false positive
CC-16	server/src/videoStreamer.cpp:43	style [unusedFunction]	unused. if not in use, delete it.
...	...	...	...

<CppCheck>

Top 5 violations		
Violated Rules	counts	Rule
cppcheck:misra_c_2012_15_01	119	The goto statement should not be used
cppcheck:misra_c_2012_14_04	88	The controlling expression of an if-statement and the controlling expression of an iteration-statement shall have essentially Boolean type
cppcheck:misra_c_2012_15_05	86	A function should have a single point of exit at the end
cppcheck:misra_c_2012_15_06	65	The body of an iteration-statement or a selection-statement shall be a compound statement
cppcheck:misra_c_2012_12_01	59	The precedence of operators within expressions should be made explicit



```
if ( i )            /* Non Compliant */
{
}

if ( i != 0 )       /* Compliant */
{
}
```

<Code x-ray(internal tool in LGE)>

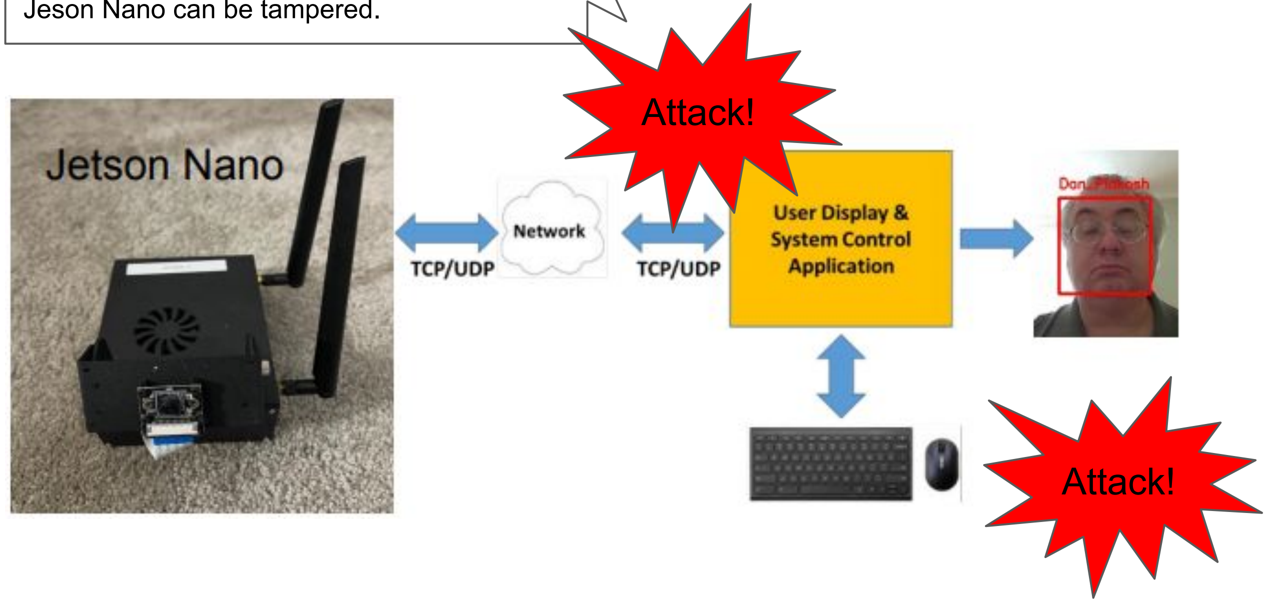
Secure Design - Input Validation

Threat 1 Attacker tries to tamper the data transmitted from Jetson Nano to client program
 e.g. invalid image header of JPEG format

Vulnerability 1 Data transmitted from Jeson Nano can be tampered.

Mitigation 1 Check the image format of received data is valid to JPEG.

How to
JPEG header check by parsing SOI (Start of Image) and EOI (End Of Image) bytes which have fixed values.



How to
Checking the input validity while in typing on the edit box of client program and deny input when violate rules

Mitigation 2 Application checks if input is valid or not and use functions that restrict the number of bytes

Threat 2 Attacker tries to force the stack overflow using invalid input and inject the executable code. e.g. user name, ip address

Vulnerability 2 Application is implemented with C/C++, which has string functions that are vulnerable to overflow, and do not check input size and format.

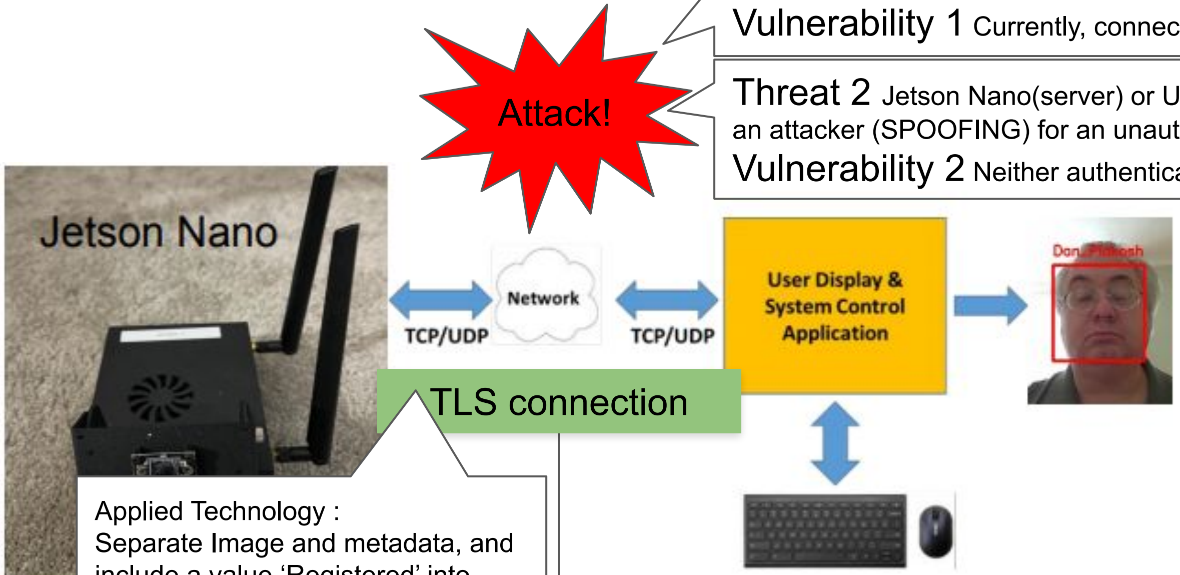
Secure Design - Secure Data Transmission

Threat 1 Attacker connects on the same network and tries sniffing data between Jetson Nano and User Application. Attacker wants to get secure information, e.g. password, or change the data e.g. face data.

Vulnerability 1 Currently, connection is not secure and data is not encrypted.

Threat 2 Jetson Nano(server) or User Application(client) might be spoofed by an attacker (SPOOFING) for an unauthorized access.

Vulnerability 2 Neither authenticate each other.



TLS connection

Applied Technology :
Separate Image and metadata, and include a value ‘Registered’ into metadata for recognized people

Rect	**Registered**	UserName

Mitigation 1 Data encryption in transit. TLS is applied for encryption. Ciphersuite : TLS_AES_128_GCM_SHA256

Mitigation 2 Server and Client must mutually authenticate each other. Therefore mutual authentication of TLS is applied.