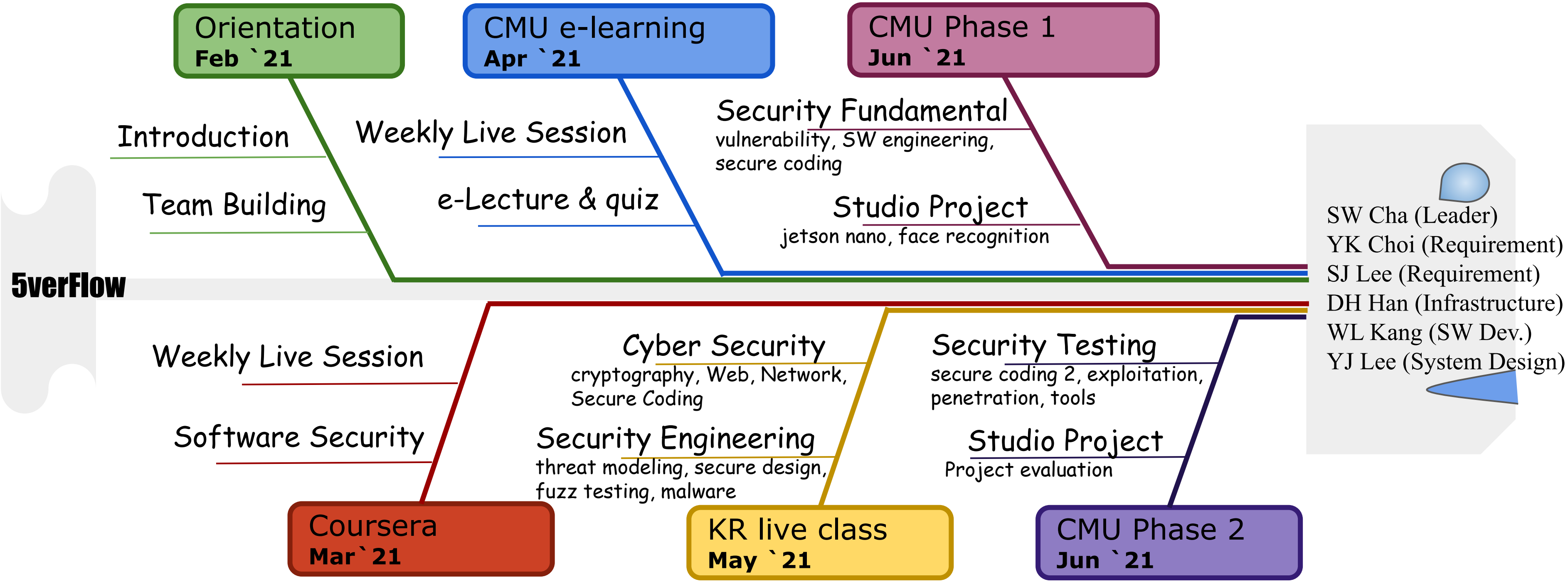
# LGE Security Specialist
# Studio Project Phase 1

Team 5. 5verFlow

# History

The road we passed



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

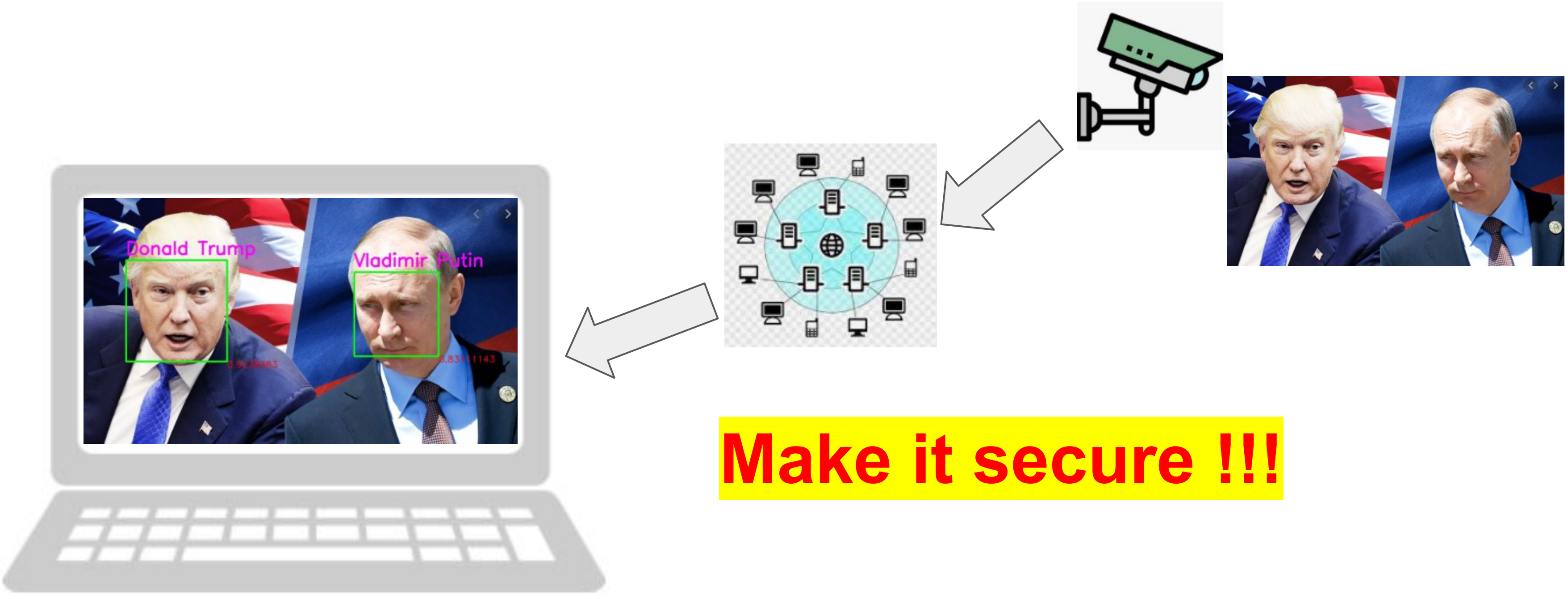
CMU Phase 2
**Jun `21**

- Security Testing
  secure coding 2, exploitation, penetration, tools
- Studio Project
  Project evaluation

SW Cha (Leader)
YK Choi (Requirement)
SJ Lee (Requirement)
DH Han (Infrastructure)
WL Kang (SW Dev.)
YJ Lee (System Design)

# Secure Face ID

- The original program is a face recognition system for video camera and video file.
- but it did not consider security, so we tried to make it secure with learnt knowledge.



**Make it secure !!!**

# Security Development Life-cycle

The way we did

Our activity :
User Requirement
Project Requirement
Software Requirement
Security Requirement

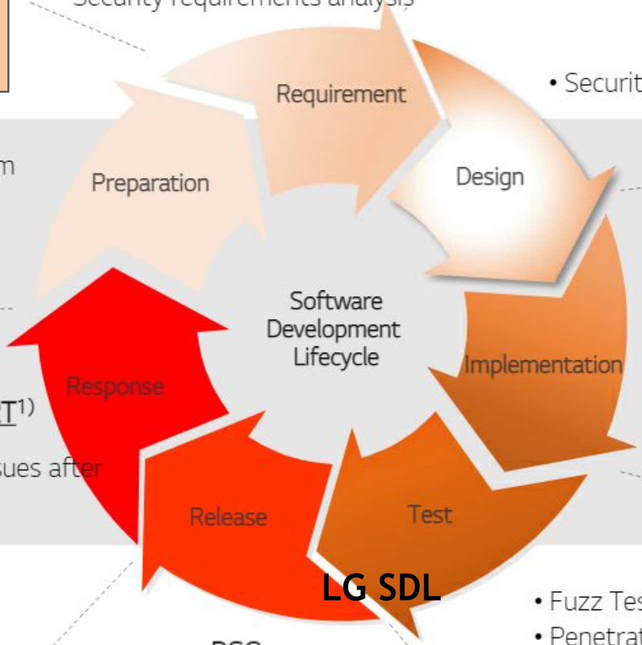
- Define product security level
- the LG-SDL Planning
- Security requirements analysis

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

PSRT[1)]

- Responding to security issues after product launch

- Fuzz Testing
- Penetration Testing

Phase 2

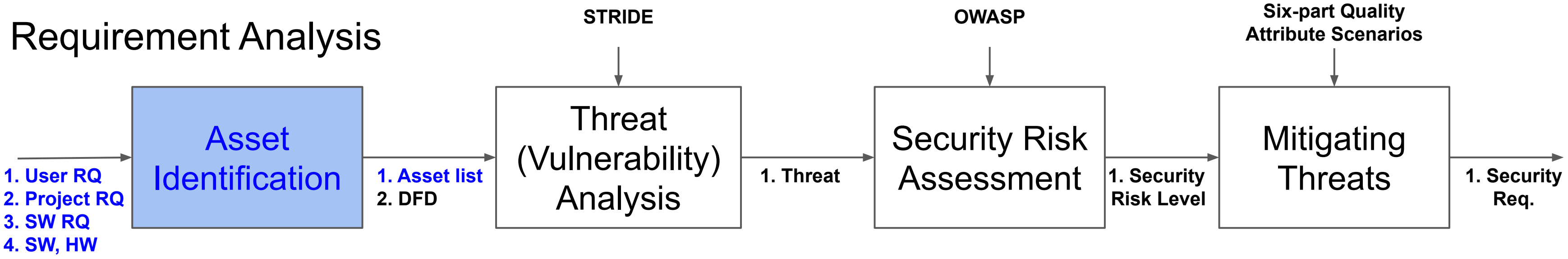
PSC

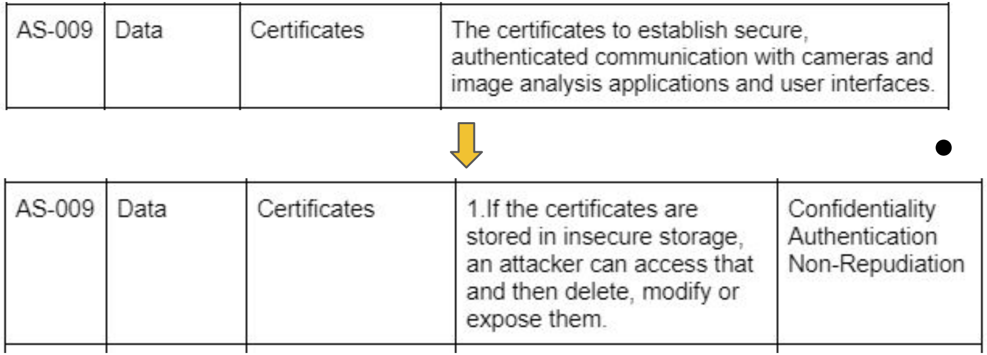
- LG Electronics Product Security Certification

# Asset Identification

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<br>Authentication<br>Non-Repudiation |

**<Asset List>**

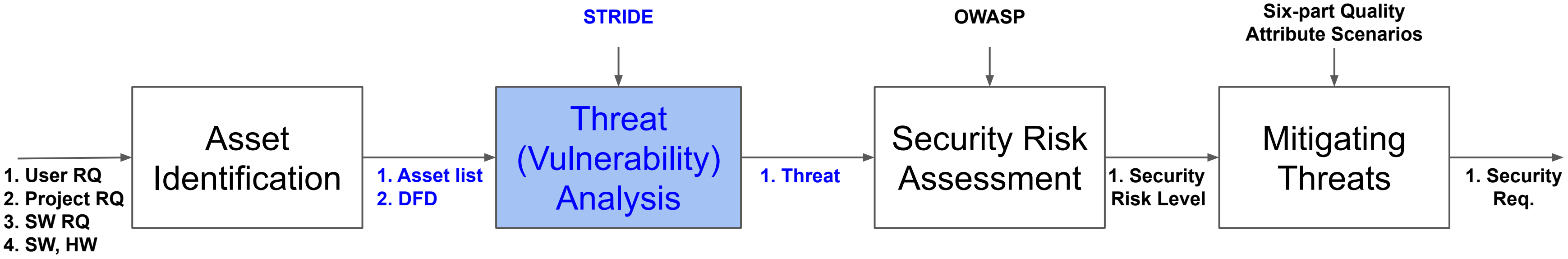
- Asset identification was possible based on functions derived through system definition, scenario creation, and requirement analysis.

- Assets to protect from threats
  - ○ Jetson Nano : server application
  - ○ User display & System control application
  - ○ Data (video frame, meta data, picture, certificate..)
  - ○ Network interface
  - ○ HW : Camera

…

**Asset List** : https://docs.google.com/document/d/17hBksXkO3t7uZDN-yFIo5x02YWFBjmVHrCzRcF1x-1A

# Threat Analysis



STRIDE

OWASP

Six-part Quality Attribute Scenarios

1. User RQ
2. Project RQ
3. SW RQ
4. SW, HW

Asset Identification

1. Asset list
2. DFD

Threat (Vulnerability) Analysis

1. Threat

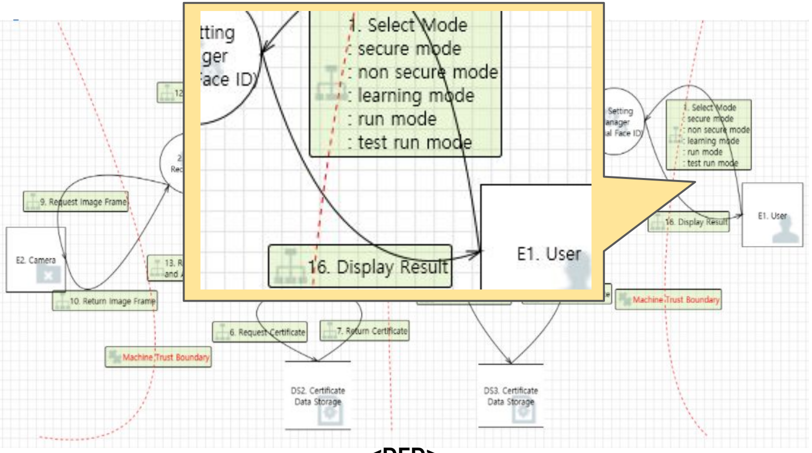
Security Risk Assessment

1. Security Risk Level

Mitigating Threats

1. Security Req.

- We mainly focused on boundary area of DFD and extracted following risk items especially data exchanges between entities.
  - HMI; Invalid data could cause buffer overflow or connecting to unauthorized system
  - Network Communication; data sniffing
  - Data Consistency; user data corruption

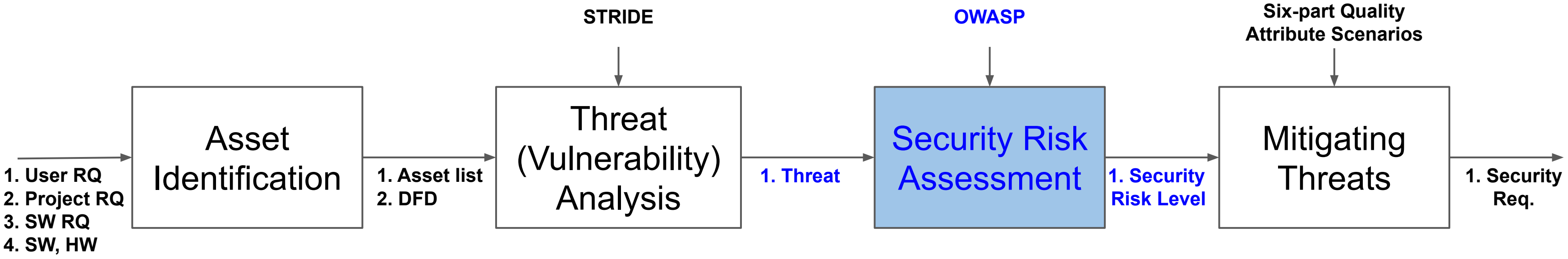


<DFD>

**Security Requirement Analysis working sheet** : https://docs.google.com/spreadsheets/d/1Mx0e_mOhwO24hP5NUkUQ4YFxJ2HmPbjVOljCh0Gsd7E

# Risk Assessment

Requirement


STRIDE

OWASP

Six-part Quality Attribute Scenarios

1. User RQ
2. Project RQ
3. SW RQ
4. SW, HW

Asset Identification

1. Asset list
2. DFD

Threat (Vulnerability) Analysis

1. Threat

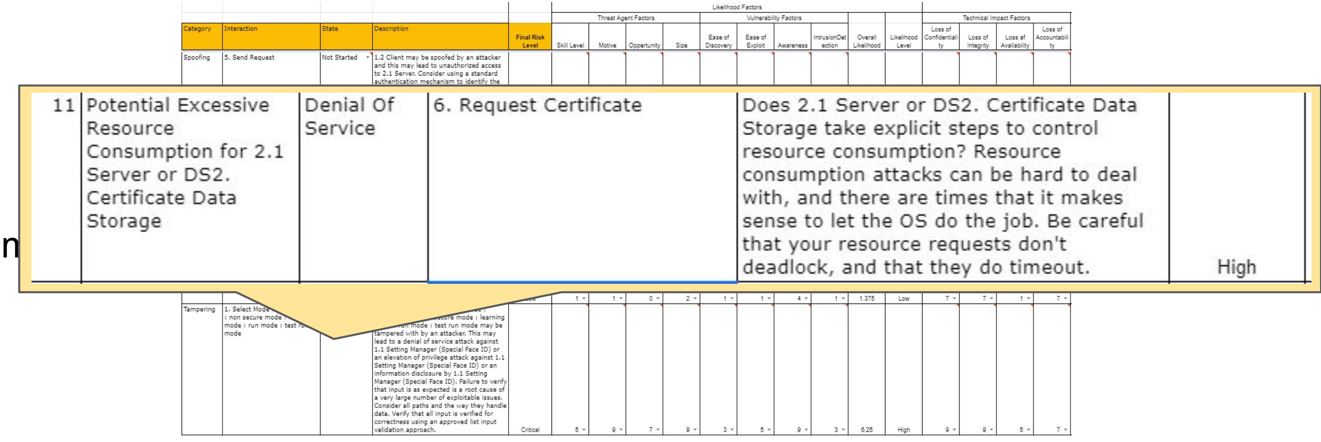
Security Risk Assessment

1. Security Risk Level

Mitigating Threats

1. Security Req.

- **19** threats / **5** category
    - Input Validation
    - Secure Data Transn
    - Authentication
    - Secure Data Store
    - Logging



| | | | | | 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 | | | | | | | | | | | | | | | |
| | | | | [illegible] | 1 | 1 | 0 | 2 | 1 | 1 | 4 | 1 | 1.375 | Low | 7 | 7 | 1 | 7 |
| Tampering | 1. Select Mode [illegible] : non secure mode : [illegible] mode : run mode : test ru[illegible] mode | [illegible] | [illegible] cure mode : learning [illegible] 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 |

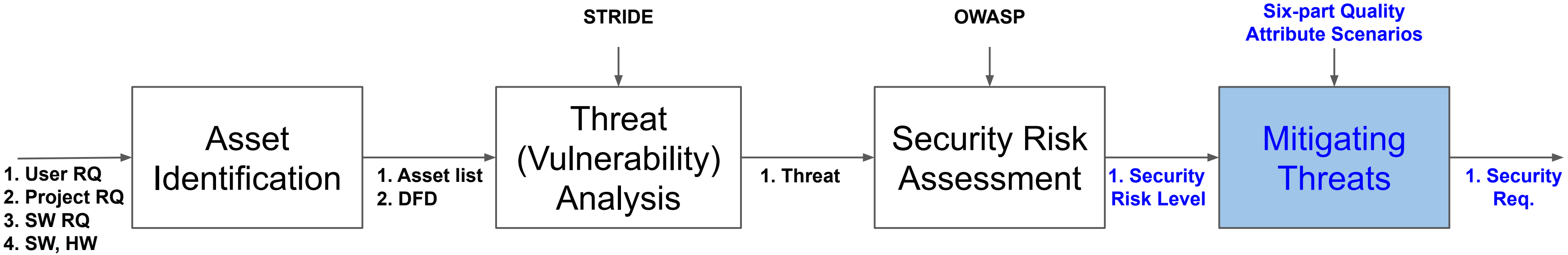
| | | | | | |
|---|---|---|---|---|---|
| 11 | Potential Excessive Resource Consumption for 2.1 Server or DS2. Certificate Data Storage | Denial Of Service | 6. Request Certificate | Does 2.1 Server or DS2. Certificate Data Storage take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout. | High |

**<Security Risk Level>**

**Security Requirement Analysis working sheet** : https://docs.google.com/spreadsheets/d/1Mx0e_mOhwO24hP5NUkUQ4YFxJ2HmPbjVOljCh0Gsd7E

# Mitigating Threats

Requirement



STRIDE

OWASP

Six-part Quality Attribute Scenarios

1. User RQ
2. Project RQ
3. SW RQ
4. SW, HW

Asset Identification

1. Asset list
2. DFD

Threat (Vulnerability) Analysis

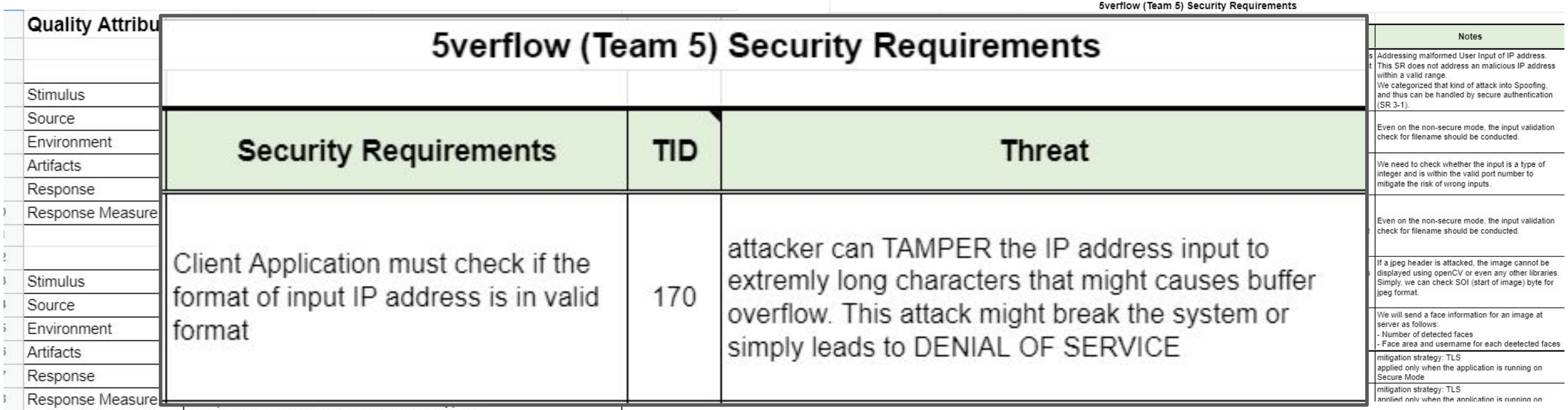
1. Threat

Security Risk Assessment

1. Security Risk Level

Mitigating Threats

1. Security Req.



| Quality Attribu |
|---|
| |
| Stimulus |
| Source |
| Environment |
| Artifacts |
| Response |
| Response Measure |
| |
| |
| Stimulus |
| Source |
| Environment |
| Artifacts |
| Response |
| Response Measure |

5verflow (Team 5) Security Requirements

## 5verflow (Team 5) Security Requirements

| Security Requirements | TID | Threat |
|---|---|---|
| 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 |

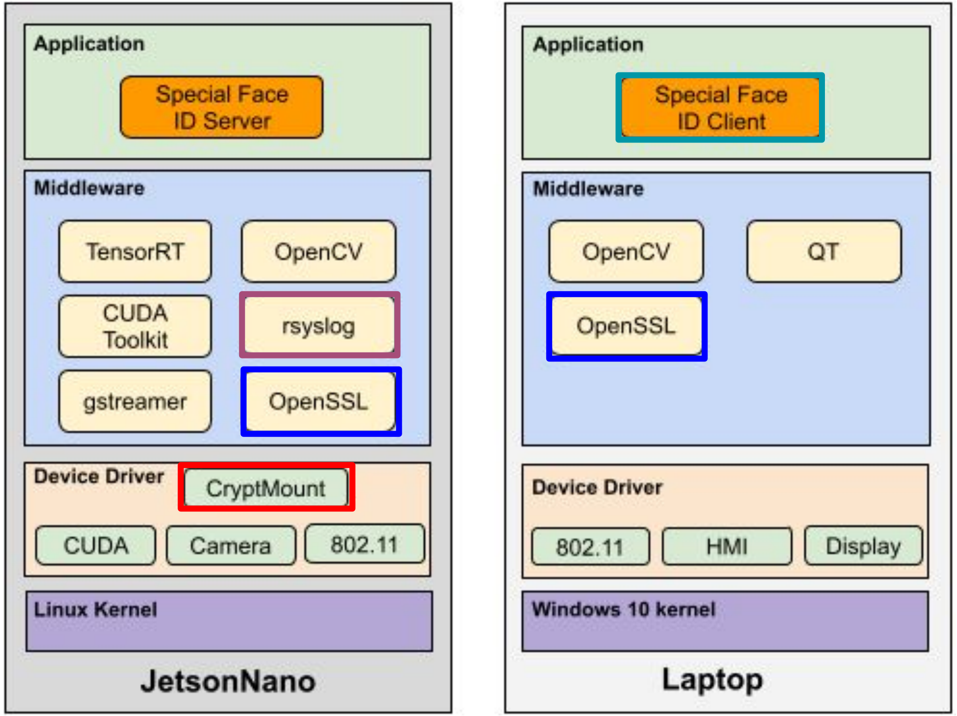
| Notes |
|---|
| Addressing malformed User Input of IP address.<br>This SR does not address an malicious IP address within a valid range.<br>We categorized that kind of attack into Spoofing, and thus can be handled by secure authentication (SR 3-1). |
| Even on the non-secure mode, the input validation check for filename should be conducted. |
| 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. |
| Even on the non-secure mode, the input validation check for filename should be conducted. |
| 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. |
| We will send a face information for an image at server as follows:<br>- Number of detected faces<br>- Face area and username for each deetected faces |
| mitigation strategy: TLS<br>applied only when the application is running on Secure Mode |
| mitigation strategy: TLS<br>applied only when the application is running on |

<Quality Attributes>

<Security Requirement>

**Security Requirement Analysis working sheet** : https://docs.google.com/spreadsheets/d/1Mx0e_mOhwO24hP5NUkUQ4YFxJ2HmPbjVOljCh0Gsd7E

# Security 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

# Input Validation

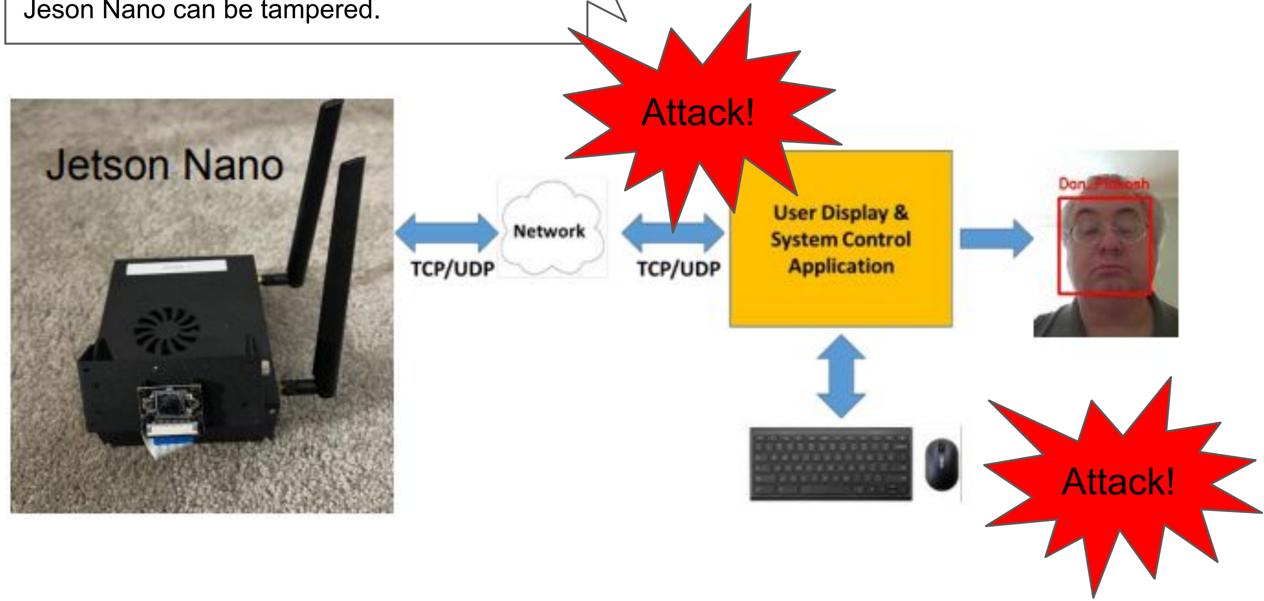
Design

**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 Data Transmission

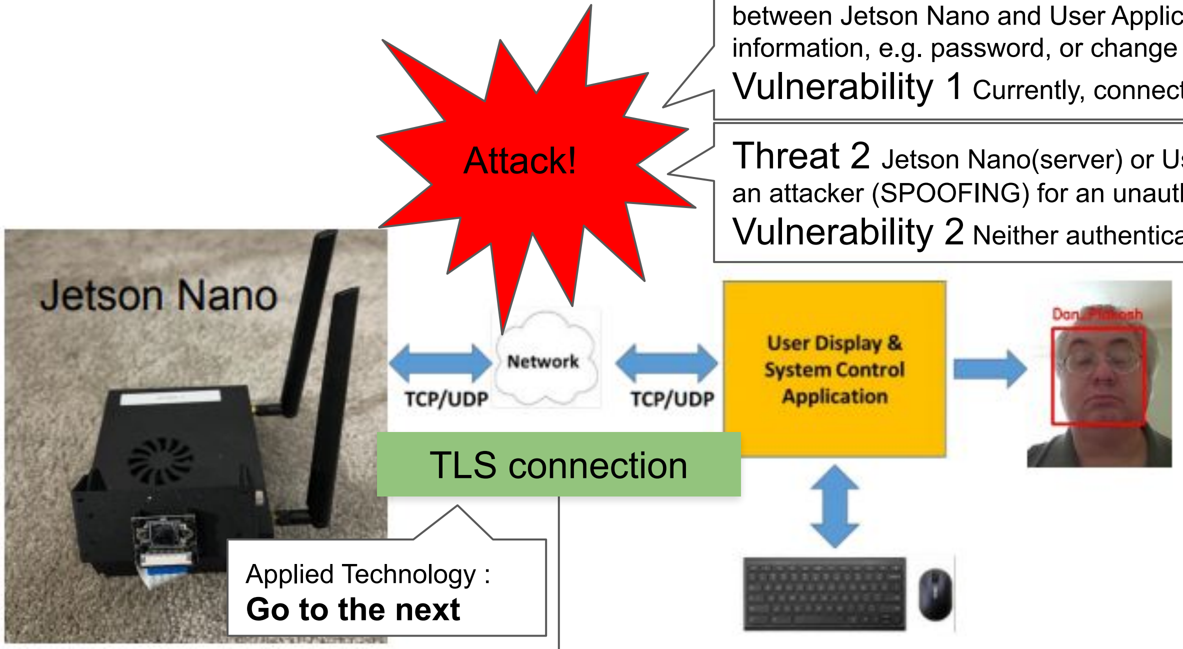
Design

**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

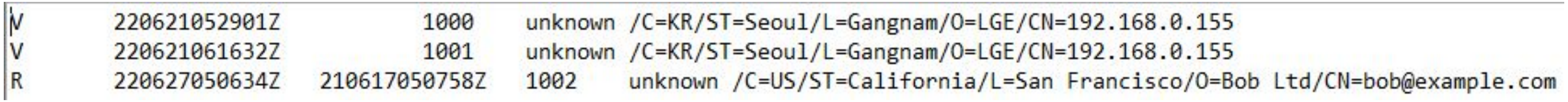
Applied Technology :
**Go to the next**

**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.

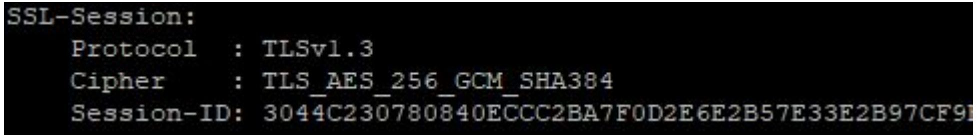
# TLS Implementation

- X.509 Certificate
  - Long key length: 4096 bits, AES-256 encrypted
  - Stored in secure storage (encrypted, not accessible to unauthorized user)
  - Permission to the keys are restricted so that only the owner can read and no one is able to write and execute
  - Certificate status (valid or revoked) are **managed**



```
V       220621052901Z                1000    unknown /C=KR/ST=Seoul/L=Gangnam/O=LGE/CN=192.168.0.155
V       220621061632Z                1001    unknown /C=KR/ST=Seoul/L=Gangnam/O=LGE/CN=192.168.0.155
R       220627050634Z   210617050758Z   1002    unknown /C=US/ST=California/L=San Francisco/O=Bob Ltd/CN=bob@example.com
```

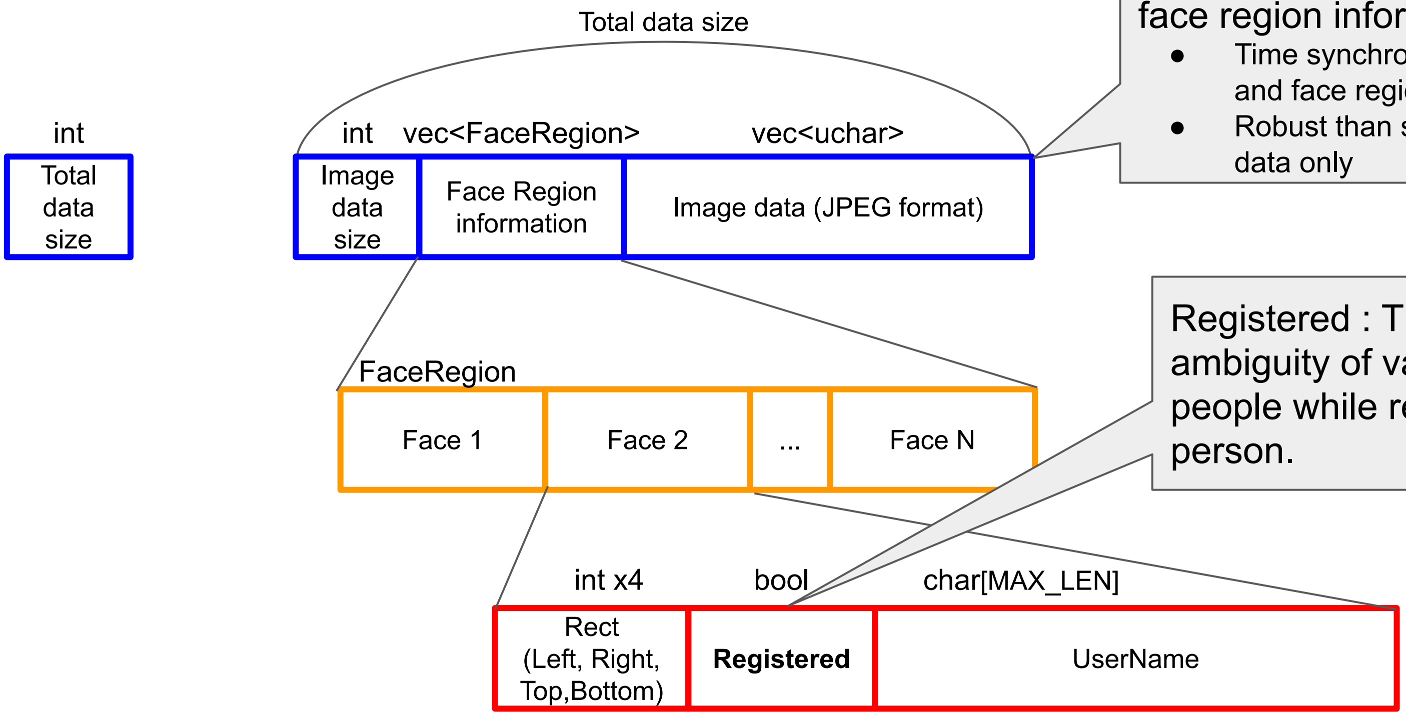
- TLS
  - TLS v1.3: faster handshake and stronger security by removing static key exchanges
  - Cipher Suite: TLS_AES_256_GCM_SHA384
    - AES 256: according to NIST Recommendations
    - GCM: provides both confidentiality and authentication using AAD
    - SHA384: susceptible to length extension attack



```
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: 3044C230780840ECCC2BA7F0D2E6E2B57E33E2B97CF9
```

# Data Validation during transfer

From Server to Client - Data Overview



Total data size

int

Total data size

int

vec<FaceRegion>

vec<uchar>

Image data size

Face Region information

Image data (JPEG format)

FaceRegion

Face 1

Face 2

...

Face N

int x4

bool

char[MAX_LEN]

Rect (Left, Right, Top,Bottom)

**Registered**

UserName

Send a image data combined with face region information

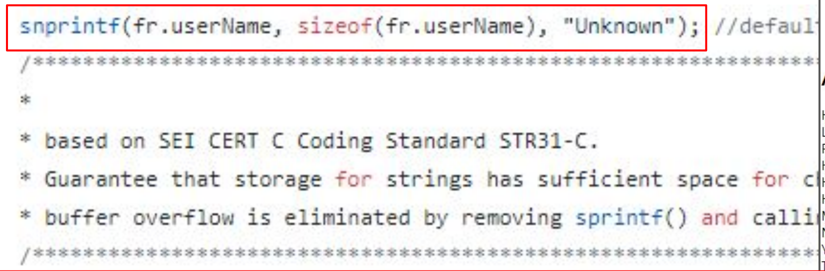
- Time synchronization between image and face region info
- Robust than sending simple structured data only

Registered : This field prevent ambiguity of various unknown people while registering a new person.

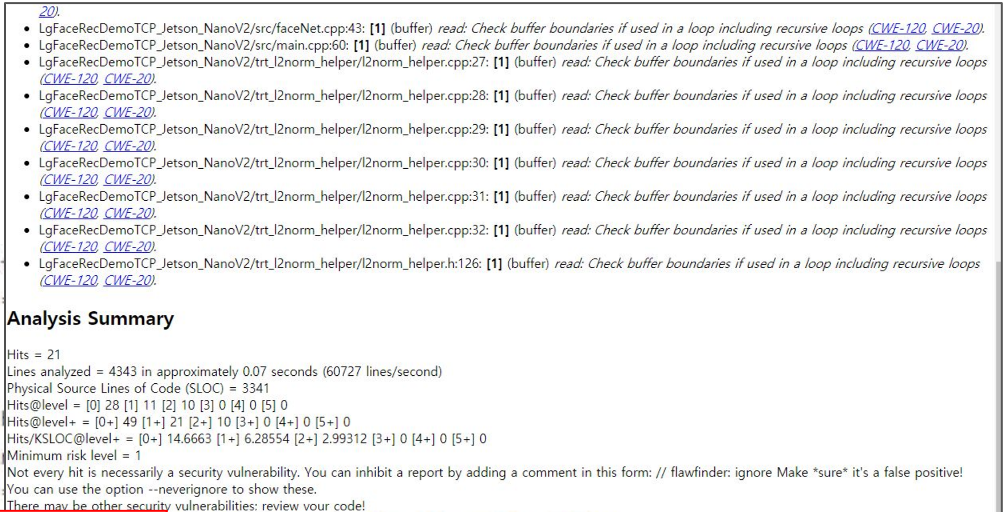
# Secure Coding

Implementation

- Static Analysis
  - FlawFinder : total 21 hits
    - ✓ fixed !!



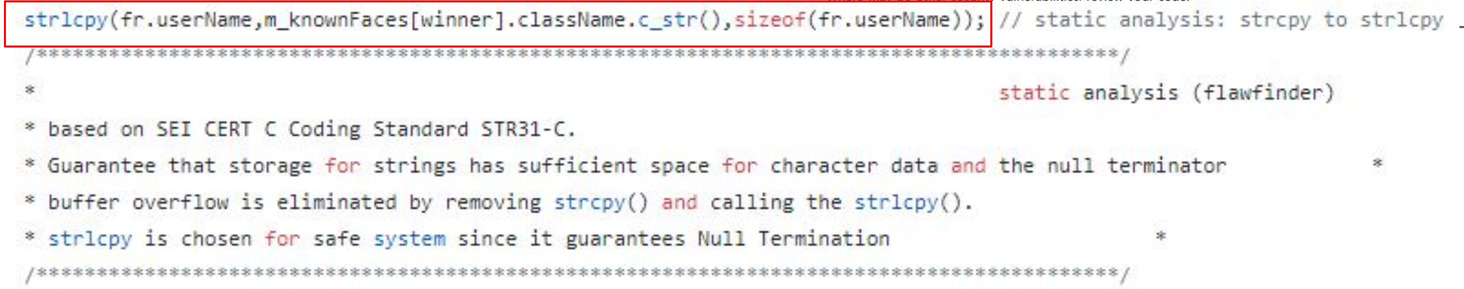
```
snprintf(fr.userName, sizeof(fr.userName), "Unknown"); //defaul
/*****************************************************************
*
* based on SEI CERT C Coding Standard STR31-C.
* Guarantee that storage for strings has sufficient space for c
* buffer overflow is eliminated by removing sprintf() and calli
/*****************************************************************
```



- 20).
- LgFaceRecDemoTCP_Jetson_NanoV2/src/faceNet.cpp:43: **[1]** (buffer) *read: Check buffer boundaries if used in a loop including recursive loops (CWE-120, CWE-20).*
- LgFaceRecDemoTCP_Jetson_NanoV2/src/main.cpp:60: **[1]** (buffer) *read: Check buffer boundaries if used in a loop including recursive loops (CWE-120, CWE-20).*
- LgFaceRecDemoTCP_Jetson_NanoV2/trt_l2norm_helper/l2norm_helper.cpp:27: **[1]** (buffer) *read: Check buffer boundaries if used in a loop including recursive loops (CWE-120, CWE-20).*
- LgFaceRecDemoTCP_Jetson_NanoV2/trt_l2norm_helper/l2norm_helper.cpp:28: **[1]** (buffer) *read: Check buffer boundaries if used in a loop including recursive loops (CWE-120, CWE-20).*
- LgFaceRecDemoTCP_Jetson_NanoV2/trt_l2norm_helper/l2norm_helper.cpp:29: **[1]** (buffer) *read: Check buffer boundaries if used in a loop including recursive loops (CWE-120, CWE-20).*
- LgFaceRecDemoTCP_Jetson_NanoV2/trt_l2norm_helper/l2norm_helper.cpp:30: **[1]** (buffer) *read: Check buffer boundaries if used in a loop including recursive loops (CWE-120, CWE-20).*
- LgFaceRecDemoTCP_Jetson_NanoV2/trt_l2norm_helper/l2norm_helper.cpp:31: **[1]** (buffer) *read: Check buffer boundaries if used in a loop including recursive loops (CWE-120, CWE-20).*
- LgFaceRecDemoTCP_Jetson_NanoV2/trt_l2norm_helper/l2norm_helper.cpp:32: **[1]** (buffer) *read: Check buffer boundaries if used in a loop including recursive loops (CWE-120, CWE-20).*
- LgFaceRecDemoTCP_Jetson_NanoV2/trt_l2norm_helper/l2norm_helper.h:126: **[1]** (buffer) *read: Check buffer boundaries if used in a loop including recursive loops (CWE-120, CWE-20).*

**Analysis Summary**

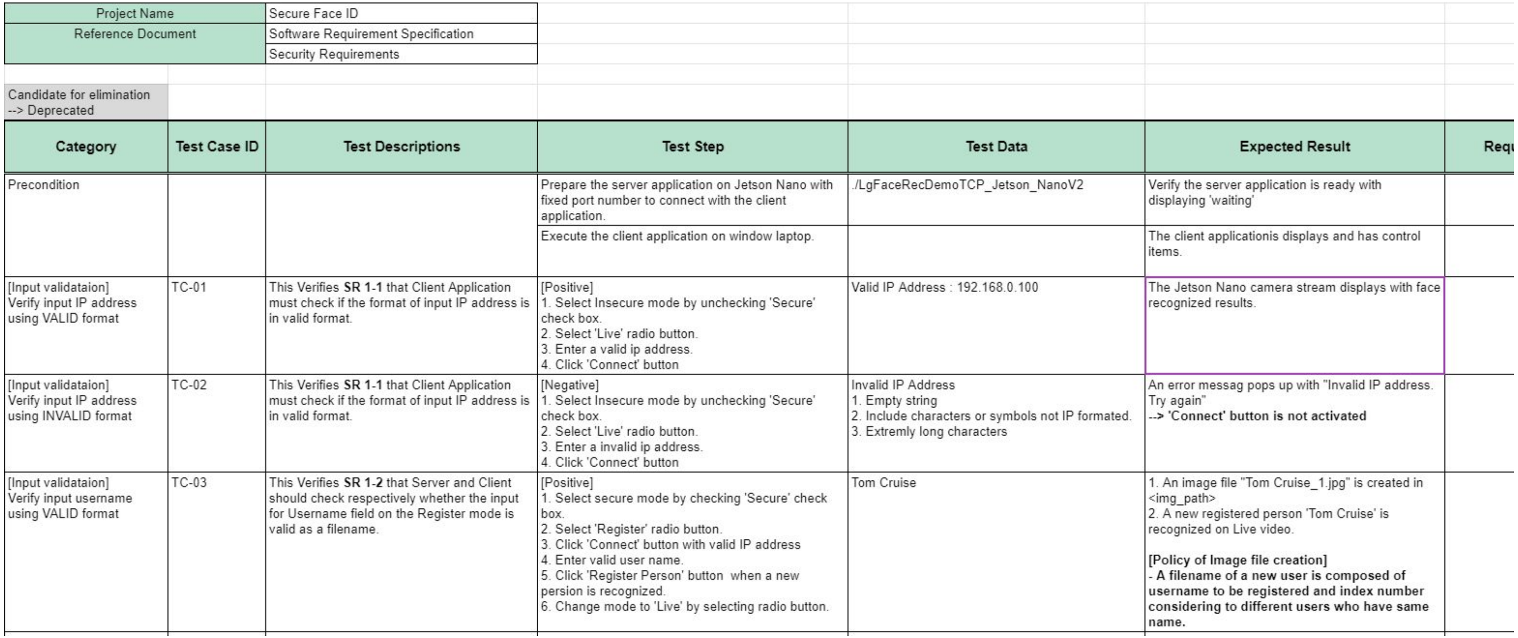
```
Hits = 21
Lines analyzed = 4343 in approximately 0.07 seconds (60727 lines/second)
Physical Source Lines of Code (SLOC) = 3341
Hits@level = [0] 28 [1] 11 [2] 10 [3] 0 [4] 0 [5] 0
Hits@level+ = [0+] 49 [1+] 21 [2+] 10 [3+] 0 [4+] 0 [5+] 0
Hits/KSLOC@level+ = [0+] 14.6663 [1+] 6.28554 [2+] 2.99312 [3+] 0 [4+] 0 [5+] 0
Minimum risk level = 1
Not every hit is necessarily a security vulnerability. You can inhibit a report by adding a comment in this form: // flawfinder: ignore Make *sure* it's a false positive!
You can use the option --neverignore to show these.
There may be other security vulnerabilities; review your code!
```



```
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                                  *
/**************************************************************************************************/
```

# Verification - Test cases

- 20 test cases for 5 categories



| | |
|---|---|
| Project Name | Secure Face ID |
| Reference Document | Software Requirement Specification<br>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]<br>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]<br>1. Select Insecure mode by unchecking 'Secure' check box.<br>2. Select 'Live' radio button.<br>3. Enter a valid ip address.<br>4. Click 'Connect' button | Valid IP Address : 192.168.0.100 | The Jetson Nano camera stream displays with face recognized results. | |
| [Input validataion]<br>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]<br>1. Select Insecure mode by unchecking 'Secure' check box.<br>2. Select 'Live' radio button.<br>3. Enter a invalid ip address.<br>4. Click 'Connect' button | Invalid IP Address<br>1. Empty string<br>2. Include characters or symbols not IP formated.<br>3. Extremly long characters | An error messag pops up with "Invalid IP address. Try again"<br>**--> 'Connect' button is not activated** | |
| [Input validataion]<br>Verify input username using VALID format | TC-03 | This Verifies **SR 1-2** that Server and Client should check respectively whether the input for Username field on the Register mode is valid as a filename. | [Positive]<br>1. Select secure mode by checking 'Secure' check box.<br>2. Select 'Register' radio button.<br>3. Click 'Connect' button with valid IP address<br>4. Enter valid user name.<br>5. Click 'Register Person' button when a new persion is recognized.<br>6. Change mode to 'Live' by selecting radio button. | Tom Cruise | 1. An image file "Tom Cruise_1.jpg" is created in <img_path><br>2. A new registered person 'Tom Cruise' is recognized on Live video.<br>**[Policy of Image file creation]**<br>**- A filename of a new user is composed of username to be registered and index number considering to different users who have same name.** | |

https://docs.google.com/spreadsheets/d/1v_cauZ085o0E29nCD0ZCVTvivRbOVXEpa1OFGBs_ujs

# Demonstration
# & Thank You