

CMU Security Specialist Course

Team Project Phase 1

Six senses Team 6

Seongju Moon (L)

Kyungnam Bae (E)

Jinmo Kim (A)

Jeonghwan Ahn (S)

* Byungchul Park (C)



Mr. Moon

Mr. Park

Mr. Ahn

Mr. Kim

Mr. Bae

Define security goal

Define assets to protect

Do threat modeling

Do risk assessment to prioritize items

Define security requirements

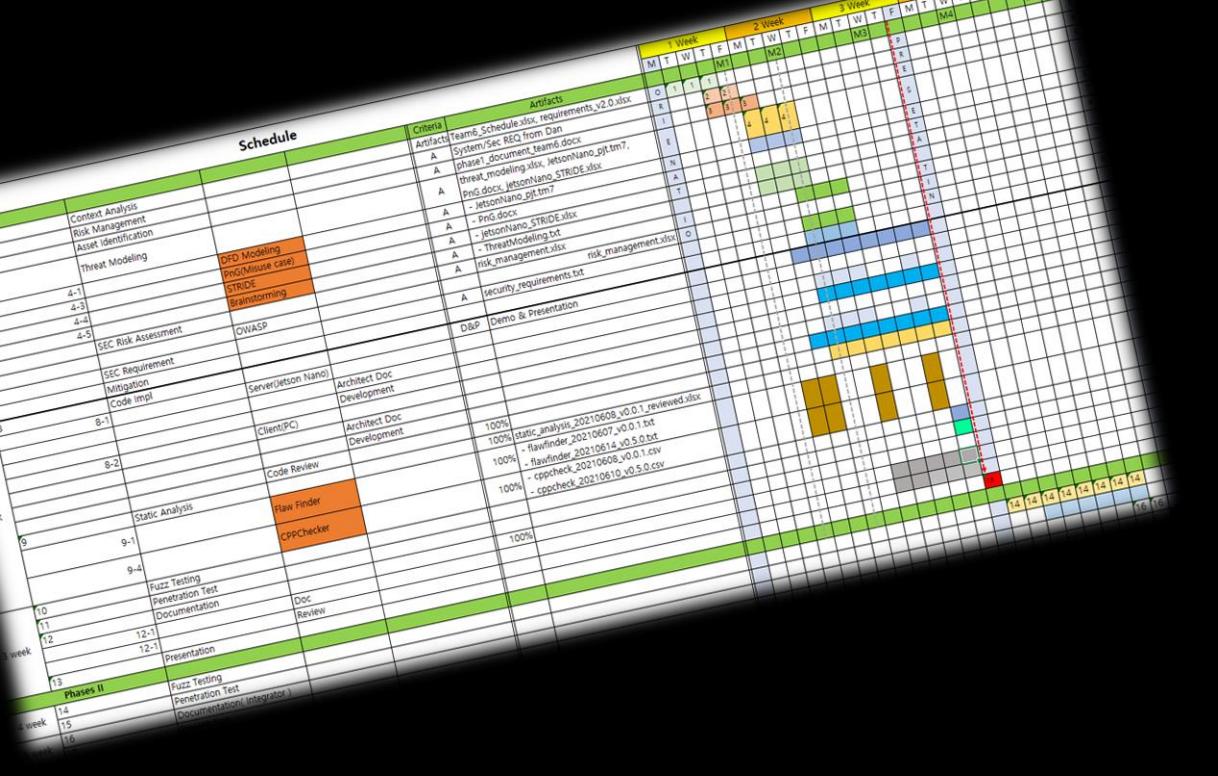
Derive mitigations

Construct architecture

Implement the mitigations

Verify the mitigations

	Schedule	Criteria	Artifacts
Context Analysis			
Risk Management			
Asset Identification			
Threat Modeling			
4-1	DFD Modeling Prod(Minuse case)		
4-3	STRIDE		
4-4	Brainstorming		
4-5			
5	SEC Risk Assessment OWASP		
6	SEC Requirement Mitigation Code Implt		
7	Server(Jetson Nano)	Architect Doc Development	A TeamB_Schedule.xlsx, requirements.v2.0.xlsx A System/Sec Req from Dan A phase1_document_teamd.docx A threat_modelling.xlsx JetsonNano_dkt.tml7 A Prod_dock_jetconlangs_STRIDE.xlsx A - Prod dock A - JetsonNano_STRIDE.xlsx A - ThreatModeling.txt A risk_management.xlsx A security_requirements.txt risk_management.xlsx
8	Client(PC)	Architect Doc Development	
9	Static Analysis	Code Review	100% static_analysis_20210608_v0.0.1_reviewed.xlsx 100% - rawflinder_20210607_v0.0.1.txt 100% - rawflinder_20210614_v0.5.0.txt 100% - rawflinder_20210614_v0.5.0.csv 100% - cppcheck_20210608_v0.0.1.csv 100% - cppcheck_20210610_v0.5.0.csv
9-1	Flaw Finder		
9-2	CPPChecker		
9-4			
10	Fuzz Testing		
11	Penetration Test		
12	Documentation	Doc Review	
12-1	Presentation		
13	Phases II		
14	Fuzz Testing		
15	Penetration Test		
16	Documentation/ Integrator		



Requirements of Secure Coding Training Program, Project Description-1,2,3

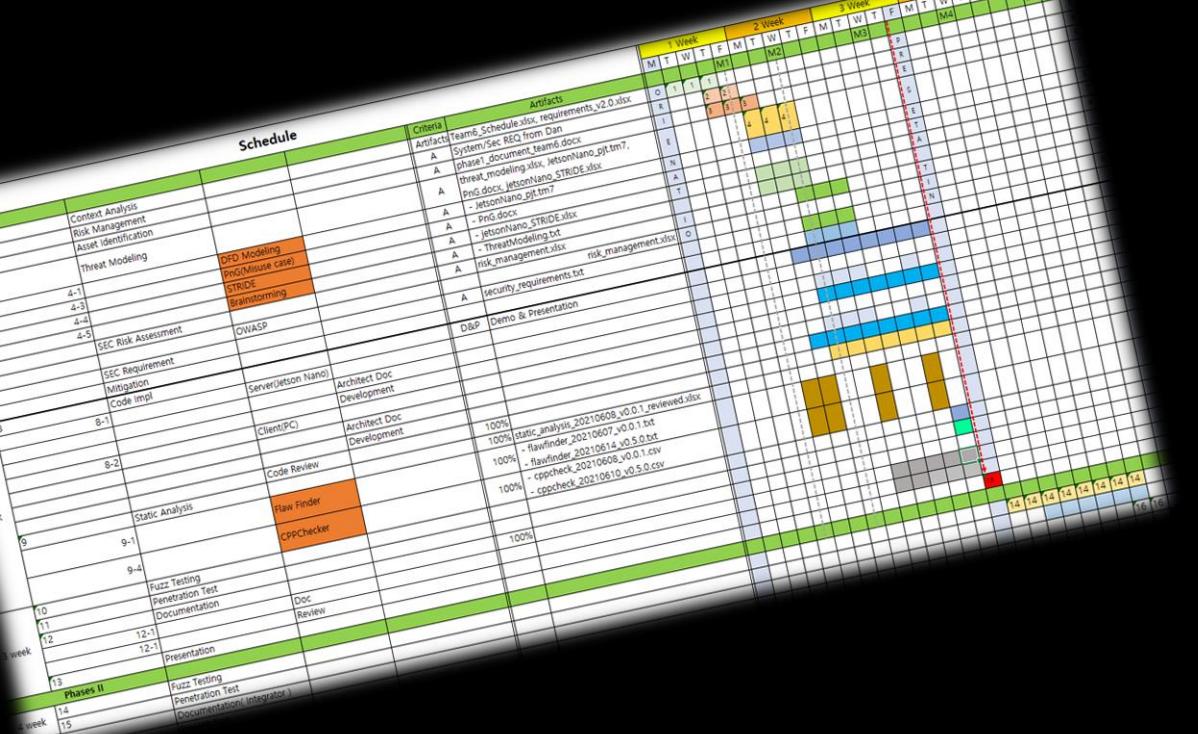
- The user display and system control application is responsible for the following:
- REQ-D-01** Establishing secure and authenticated communication with the camera and image analysis application and user interface when secure mode is selected or requested.
 - REQ-D-02** Provides the user interface to control the system. User Interface shall support the following modes of operation:
 - Secure or non secure mode of communication **REQ-D-03**
 - Learning Mode - User images can be added to the image database. In this mode the interface should query for the name of the person in front of the camera and the number of samples to be collected.
 - Run Mode - System utilizes camera to identify faces and perform facial recognition.
 - Test Run Mode - System utilizes a video file to identify faces and perform facial recognition.
 - REQ-D-04** Communicating with the camera and image analysis application as specified.
 - REQ-D-05** Display image frames and any accompanying amplifying analysis information received from the camera and image analysis application in the format specified.
 - REQ-D-06** Additional consideration
 - REQ-D-07** Communicating with the camera and image analysis application in the format specified.

Project Responsibilities

- REQ-Q-01** Implementing the specified enhancements to the applications **additional consideration**
- REQ-Q-02** Ensuring that all software in both applications are architected and coded to be secure and free of vulnerabilities.
- REQ-D-08** Modifying the implementation so the applications support two modes of communications: 1) a secure mode with all data properly encrypted (including authentication) and 2) a plain text mode without encryption.
- REQ-Q-03** Proper fault/error detection, recovery, and reporting.
- REQ-Q-04** Analyzing the provided initial implementation for vulnerabilities and developing solutions to mitigate.
- REQ-Q-05** Analyze another team's implementation assigned to you for security flaws and vulnerabilities.

Phase 2

* reference : LG May 2021 Lecture Secure Coding Project Intro V1.1.pptx.pdf



Requirements of Secure Coding Training Program, Project Description-1,2,3

The user display and system control application is responsible for the following:

- Establishing secure and authenticated communication with the camera and image analysis application and user interface when secure mode is selected or requested.
- Provides the user interface to control the system. User Interface shall support the following modes of operation:
 - Secure or non secure mode of communication **REQ-D-03**
 - Learning Mode - User images can be added to the image database. In this mode the interface should query for the name of the person in front of the camera and the number of samples to be collected.
 - Run Mode - System utilizes camera to identify faces and perform facial recognition.
 - Test Run Mode - System utilizes a video file to identify faces and perform facial recognition.
- Communicating with the camera and image analysis application as specified.
- Display image frames and any accompanying amplifying analysis information received from the camera and image analysis application in the format specified.

Project Responsibilities

- Implementing the specified enhancements to the applications **additional consideration**
- Ensuring that all software in both applications are architected and coded to be secure and free of vulnerabilities.
- Modifying the implementation so the applications support two modes of communications: 1) a secure mode with all data properly encrypted (including authentication) and 2) a plain text communication.
- Conduct proper fault/error detection, recovery, and reporting.
- Reviewing the provided initial implementation for vulnerabilities and developing solutions to mitigate them.
- Ensuring that another team's implementation assigned to you for review has no major flaws and vulnerabilities.

Phase 2

LG May 2021 Lecture Secure Coding Project Intro V1.1.pptx.pdf

Requirements of Tartan Secure Camera Application

The proposed system has the following basic functional requirements. Note

- REQ-D-10 • A user should be able to initiate a video feed, end a feed.
- REQ-D-11 • A user should be able to end a video feed.
- REQ-D-12 • A user should be able to save a video feed for offline review.
- REQ-D-13 • A user should be able to tune image analysis.

The system also has the following architectural concerns (i.e. quality attributes)

- REQ-Q-06 • Performance: The system must deliver video as close to real time as possible, especially in real-time mode.
- REQ-Q-14 • Authentication: The system must use two factor authentication for sign on and user credentials must be protected. Lost or compromised credentials must be handled in a reasonable way. **REQ-D-16**
- REQ-D-17 • Communication privacy: When in the desired mode the system must ensure that data sent to a user remains private while in transit. No intermediary should be able to snoop or spy on an ongoing video feed.
- REQ-D-18 • Proof of identity (nonrepudiation): Users should be confident that the camera they are using is the one that they believe it is.
- REQ-D-19 • Multi-user privacy: The system must ensure that multiple video feeds remain private between the intended users.
- REQ-Q-20 • Reliability: The system must ensure that video is reliably delivered. The system should recover from networking errors as soon as possible. The goal is to maintain a secure, performant connection at all costs.
- REQ-Q-21 • Maintainability: The system must be easy to maintain and upgrade. This includes providing clear documentation, easy access to source code, and a modular design.

Aside from these requirements, there are a number of basic quality concerns that must be addressed during development.

- REQ-Q-07 1. Ensuring that all software in both applications are architected and coded to be secure and free of vulnerabilities.
- REQ-Q-08 2. Conduct proper fault/error detection, recovery and reporting.
- REQ-Q-09 3. Ensure the developed software adheres to the company coding standard and quality standards.
- REQ-Q-10 4. Implement security measures such as encryption, authentication, and access control.

Schedule

Criteria

Artifacts

Week	TeamB_Schedule.xlsx, requirements.v2.0.xlsx						
	1 Week	2 Week	3 Week	4 Week	5 Week	6 Week	7 Week
1	O	M1	M2	M3	M4	M5	M6
2	R	E	S	F	T	M	
3	A	A	A	A	A	A	
4	A	A	A	A	A	A	
5	A	A	A	A	A	A	
6	A	A	A	A	A	A	
7	A	A	A	A	A	A	
8	D&P	Demo & Presentation					
9	Server(Jetson Nano)	Architect Doc Development					
10	Client(PC)	Architect Doc Development					
11	Code Review						
12	Static Analysis						
13	Raw Finder						
14	CPPChecker						
15	Fuzz Testing						
16	Penetration Test						
17	Documentation	Doc Review					
18	Presentation						

Phases II

3 week

4 week

5 week

6 week

7 week

8 week

9 week

10 week

11 week

12 week

13 week

14 week

15 week

16 week

17 week

18 week

19 week

20 week

21 week

22 week

23 week

24 week

25 week

26 week

27 week

28 week

29 week

30 week

31 week

32 week

33 week

34 week

35 week

36 week

37 week

38 week

39 week

40 week

41 week

42 week

43 week

44 week

45 week

46 week

47 week

48 week

49 week

50 week

51 week

52 week

53 week

54 week

55 week

56 week

57 week

58 week

59 week

60 week

61 week

62 week

63 week

64 week

65 week

66 week

67 week

68 week

69 week

70 week

71 week

72 week

73 week

74 week

75 week

76 week

77 week

78 week

79 week

80 week

81 week

82 week

83 week

84 week

85 week

86 week

87 week

88 week

89 week

90 week

91 week

92 week

93 week

94 week

95 week

96 week

97 week

98 week

99 week

100 week

The image displays a complex software development environment with multiple overlapping windows:

- Top Left:** A Gantt chart titled "Schedule" showing tasks from week 4 to 16. Tasks include Context Analysis, Risk Management, Asset Identification, Threat Modeling, SEC Risk Assessment, SEC Requirement Mitigation, Code Impl, Static Analysis, Fuzz Testing, Penetration Test, Documentation, and Present. The chart includes a legend for artifacts like TeamB_Schedule.xlsx, requirements_v2.0.xlsx, and threat_modeling_and_jetsonnano_qt_tm7.
- Middle Left:** A table titled "Requirements from CMU documents on left" under the heading "DEVELOPMENT". It lists requirements REQ-D-01 through REQ-D-21, each with a reference ID and a detailed description.
- Bottom Left:** A table titled "Requirements of Tartan Secure Camera Application" under the heading "DEVELOPMENT". It lists requirements CMU-REQ-D-01 through CMU-REQ-D-15, each with a status (Mandatory, Excluded) and notes.
- Right Side:** A large callout box containing the following text:

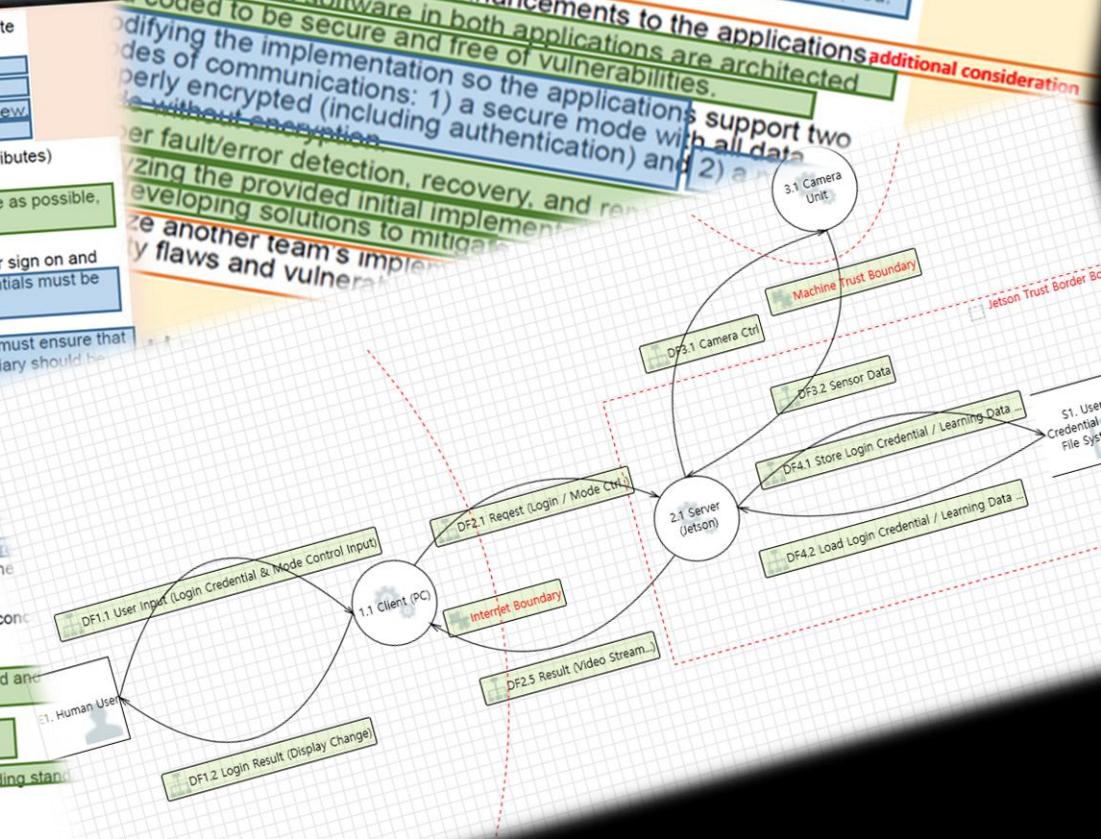
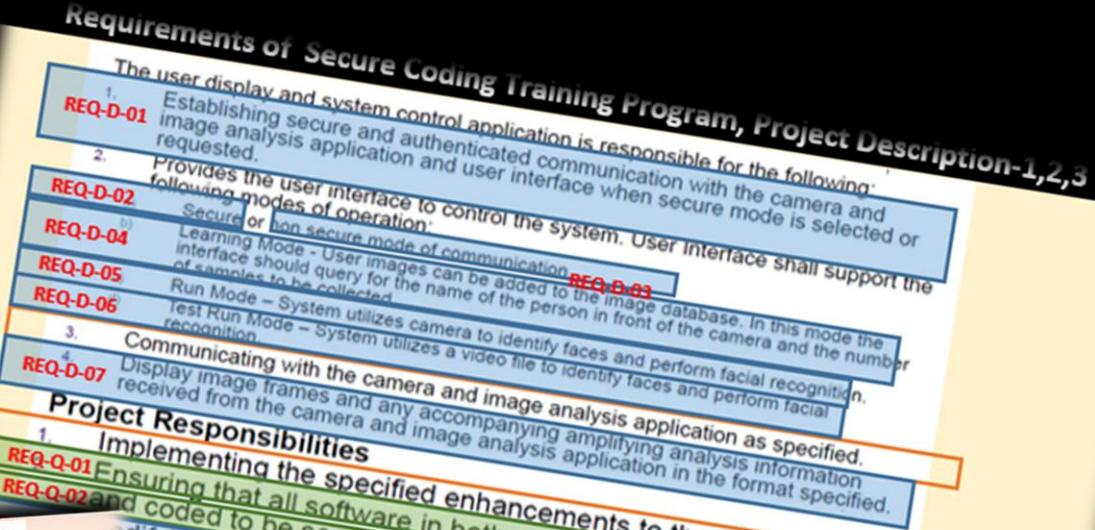
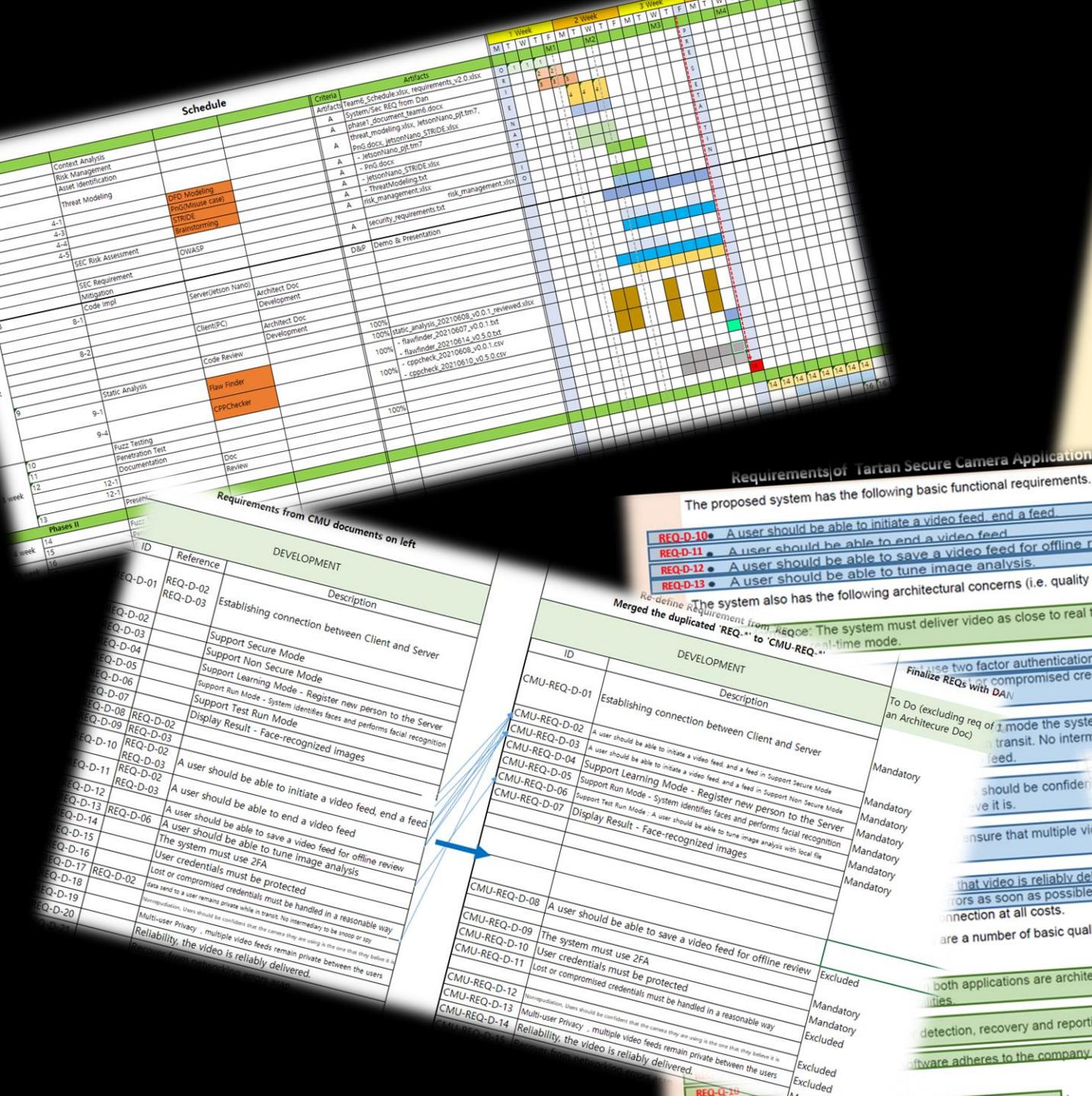
The proposed system has the following basic functional requirements.

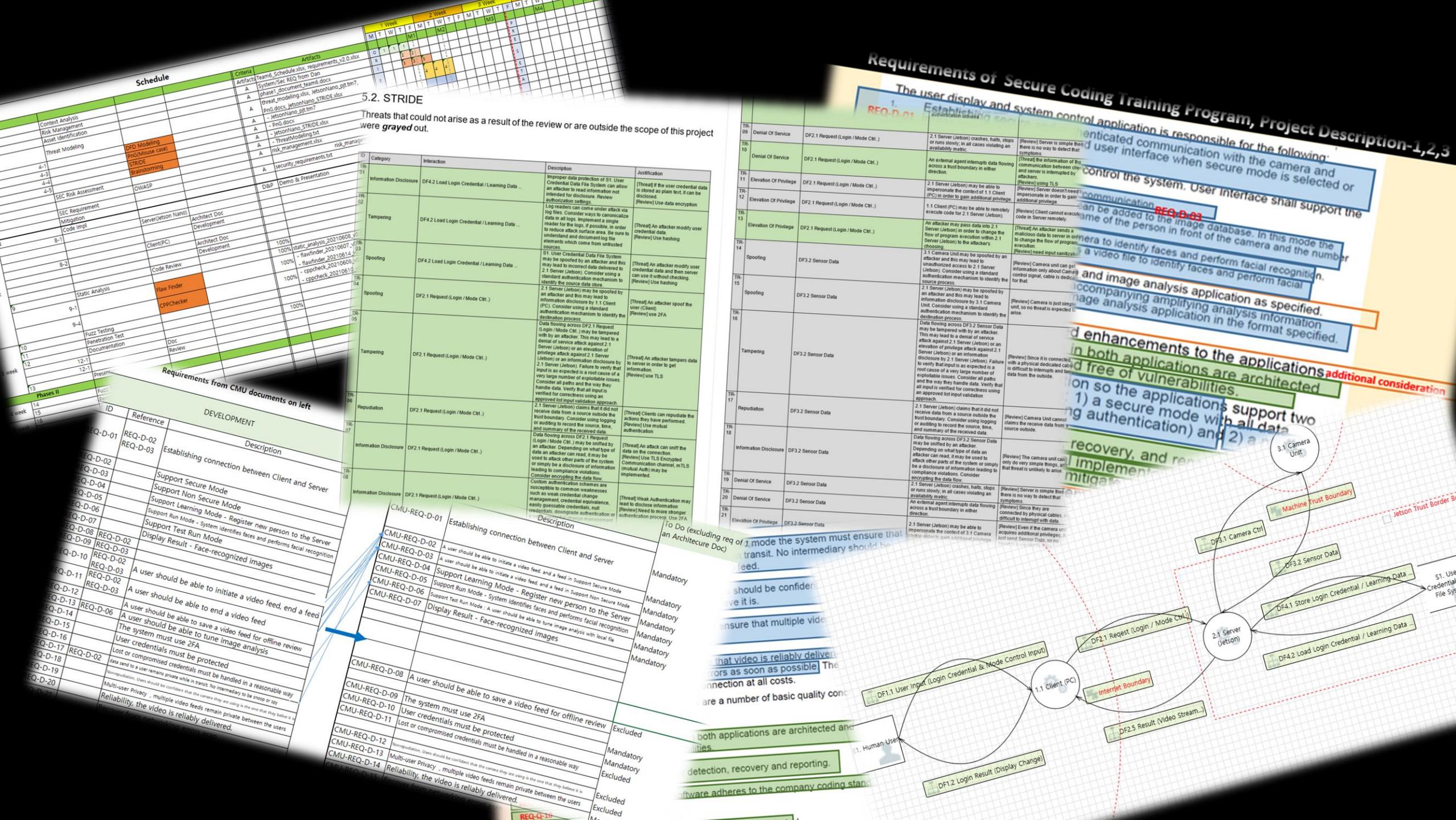
 - REQ-D-10 • A user should be able to initiate a video feed, end a feed
 - REQ-D-11 • A user should be able to end a video feed
 - REQ-D-12 • A user should be able to save a video feed for offline review
 - REQ-D-13 • A user should be able to tune image analysis

The system also has the following architectural concerns (i.e. quality of service):

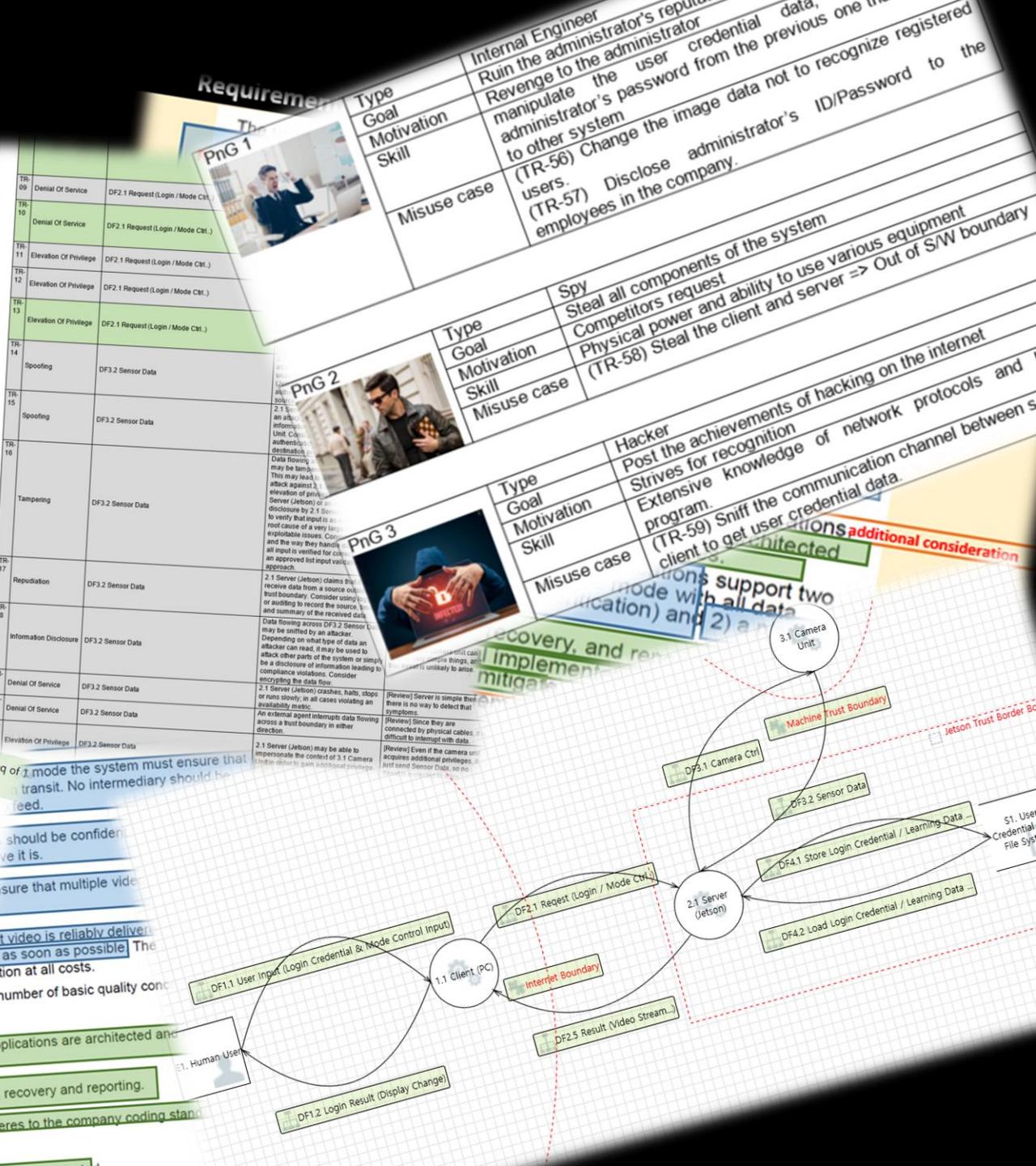
 - Merged the duplicated 'REQ-' to 'CMU-REQ-'.
 - Finalize REQs with DAN.
 - To Do (excluding req of architecture doc)
 - both applications are architected in a similar way.
 - detection, recovery and reporting.
 - software adheres to the company's security policies.

Requirements of Secure Coding Training Program, Project Description-1,2,3	
	The user display and system control application is responsible for the following:
REQ-D-01	1. Establishing secure and authenticated communication with the camera and image analysis application and user interface when secure mode is selected or requested.
REQ-D-02	2. Provides the user interface to control the system. User Interface shall support the following modes of operation:
REQ-D-04	Secure or Non-secure mode of communication.
REQ-D-05	Learning Mode - User images can be added to the image database. In this mode the interface should query for the name of the person in front of the camera and the number of samples to be collected.
REQ-D-06	Run Mode - System utilizes camera to identify faces and perform facial recognition.
REQ-D-07	Test Run Mode - System utilizes a video file to identify faces and perform facial recognition.
	3. Communicating with the camera and image analysis application as specified.
	4. Display image frames and any accompanying amplifying analysis information received from the camera and image analysis application in the format specified.
Project Responsibilities	
REQ-Q-01	1. Implementing the specified enhancements to the applications additional consideration
REQ-Q-02	2. Ensuring that all software in both applications are architected and coded to be secure and free of vulnerabilities.
	3. Modifying the implementation so the applications support two modes of communications: 1) a secure mode with all data properly encrypted (including authentication) and 2) a plain text mode without encryption.
	4. Implementing fault/error detection, recovery, and reporting.
	5. Optimizing the provided initial implementation for vulnerabilities and developing solutions to mitigate them.
	6. Assisting another team's implementation assigned to you for identifying flaws and vulnerabilities.
	REQ-D-09
	Phase 2





Requirements from CMU documents on left							Requirements from right side		
DEVELOPMENT			Description				Actions		
Phases II	ID	Reference	REQ-D-01	Establishing connection between Client and Server	DF2.1 Request (Login / Mode Ctrl.)	2.1 Server (Attack). Failure to properly handle data input as expected is a root cause of a very large number of exploitable issues. Consider the way they handle data. Verify that data is verified for correctness using an agreed list input validation approach.	[Threat] Clients can repudiate the actions they have performed.	[Review] Use TLS authentication	
1 week			REQ-D-02	Support Secure Mode	DF2.1 Request (Login / Mode Ctrl.)	Data flowing across DF2.1 Request (Login / Mode Ctrl.) may be sniffed by an attacker. Consider on what type of data an attacker could gain. It may be used to attack other parts of the system or simply be a disclosure of information leading to further violations. Consider encrypting the data.	[Threat] An attack can sniff the data.	[Review] Use TLS Encrypted communication channel, mTLS (mutual Auth) may be implemented.	
14			REQ-D-03	Support Non Secure Mode	DF2.1 Request (Login / Mode Ctrl.)	Custom authentication schemes are often common weaknesses such as weak credential change management, credential equivalence, easily guessable credentials, null credentials, downgrade authentication or	[Threat] Weak Authentication may lead to disclose information.	[Review] Need to strengthen authentication process. Use 2FA	To Do (excluding re-architecture doc)
15			REQ-D-04	Support Learning Mode - Register new person to the Server	CMU-REQ-D-01	Establishing connection between Client and Server	Description		Mandatory
16			REQ-D-05	Support Run Mode - System identifies faces and performs facial recognition	CMU-REQ-D-02	A user should be able to initiate a video feed, and a feed in Support Secure Mode			Mandatory
EQ-D-01	REQ-D-06	REQ-D-07	REQ-D-08	Support Test Run Mode	CMU-REQ-D-03	A user should be able to initiate a video feed, and a feed in Support Non Secure Mode			Mandatory
EQ-D-02	REQ-D-09	REQ-D-10	REQ-D-11	Display Result - Face-recognized images	CMU-REQ-D-04	Support Learning Mode - Register new person to the Server			Mandatory
EQ-D-03	REQ-D-12	REQ-D-13	REQ-D-14	A user should be able to initiate a video feed	CMU-REQ-D-05	Support Run Mode - System identifies faces and performs facial recognition			Mandatory
EQ-D-04	REQ-D-15	REQ-D-16	REQ-D-17	A user should be able to end a video feed	CMU-REQ-D-06	Support Test Run Mode - A user should be able to tune image analysis with local file			Mandatory
EQ-D-05	REQ-D-18	REQ-D-19	REQ-D-20	A user should be able to save a video feed for offline review	CMU-REQ-D-07	Display Result - Face-recognized images			Mandatory
EQ-D-06	REQ-D-21	REQ-D-22	REQ-D-23	The system must use 2FA	CMU-REQ-D-08	A user should be able to save a video feed for offline review			Mandatory
EQ-D-07	REQ-D-24	REQ-D-25	REQ-D-26	User credentials must be protected	CMU-REQ-D-09	The system must use 2FA		Excluded	Both applications
EQ-D-08	REQ-D-27	REQ-D-28	REQ-D-29	Lost or compromised credentials must be handled in a reasonable way	CMU-REQ-D-10	User credentials must be protected		Mandatory	both applications
EQ-D-09	REQ-D-30	REQ-D-31	REQ-D-32	No reproduction. Users should be confident that the camera they are using is the one that they believe it is	CMU-REQ-D-11	Lost or compromised credentials must be handled in a reasonable way		Excluded	both applications
EQ-D-10	REQ-D-33	REQ-D-34	REQ-D-35	Nonrepudiation. Users should be confident that the camera they are using is the one that they believe it is	CMU-REQ-D-12	Nonrepudiation. Users should be confident that the camera they are using is the one that they believe it is		Excluded	both applications
EQ-D-11	REQ-D-36	REQ-D-37	REQ-D-38	Multi-user Privacy , multiple video feeds remain private between the users	CMU-REQ-D-13	Multi-user Privacy , multiple video feeds remain private between the users		Excluded	both applications
EQ-D-12	REQ-D-39	REQ-D-40	REQ-D-41	Reliability, the video is reliably delivered.	CMU-REQ-D-14	Reliability, the video is reliably delivered.		Excluded	both applications
EQ-D-13	REQ-D-42	REQ-D-43	REQ-D-44	Recover from networking anomalies	CMU-REQ-D-15	Recover from networking anomalies		Excluded	both applications



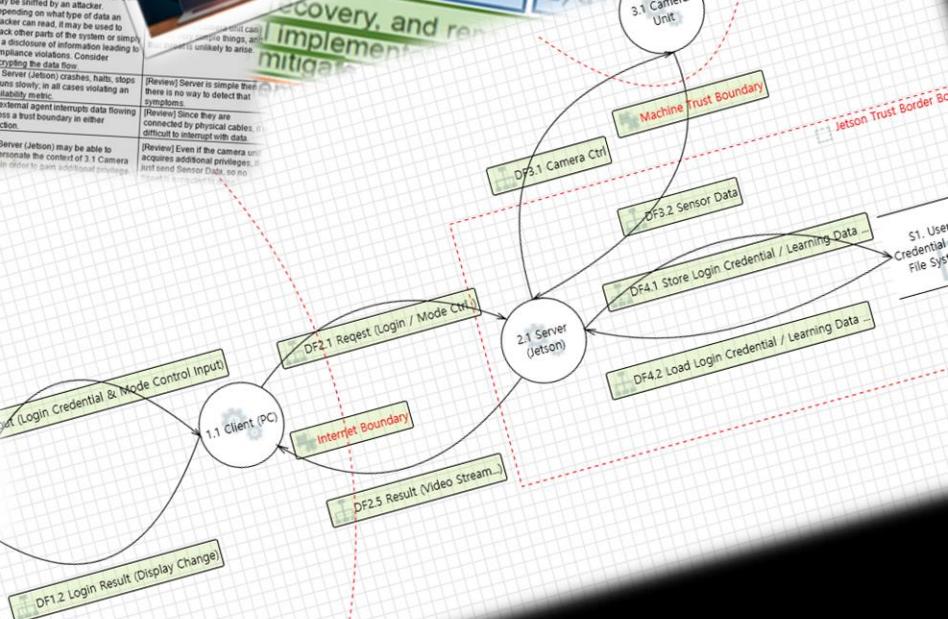
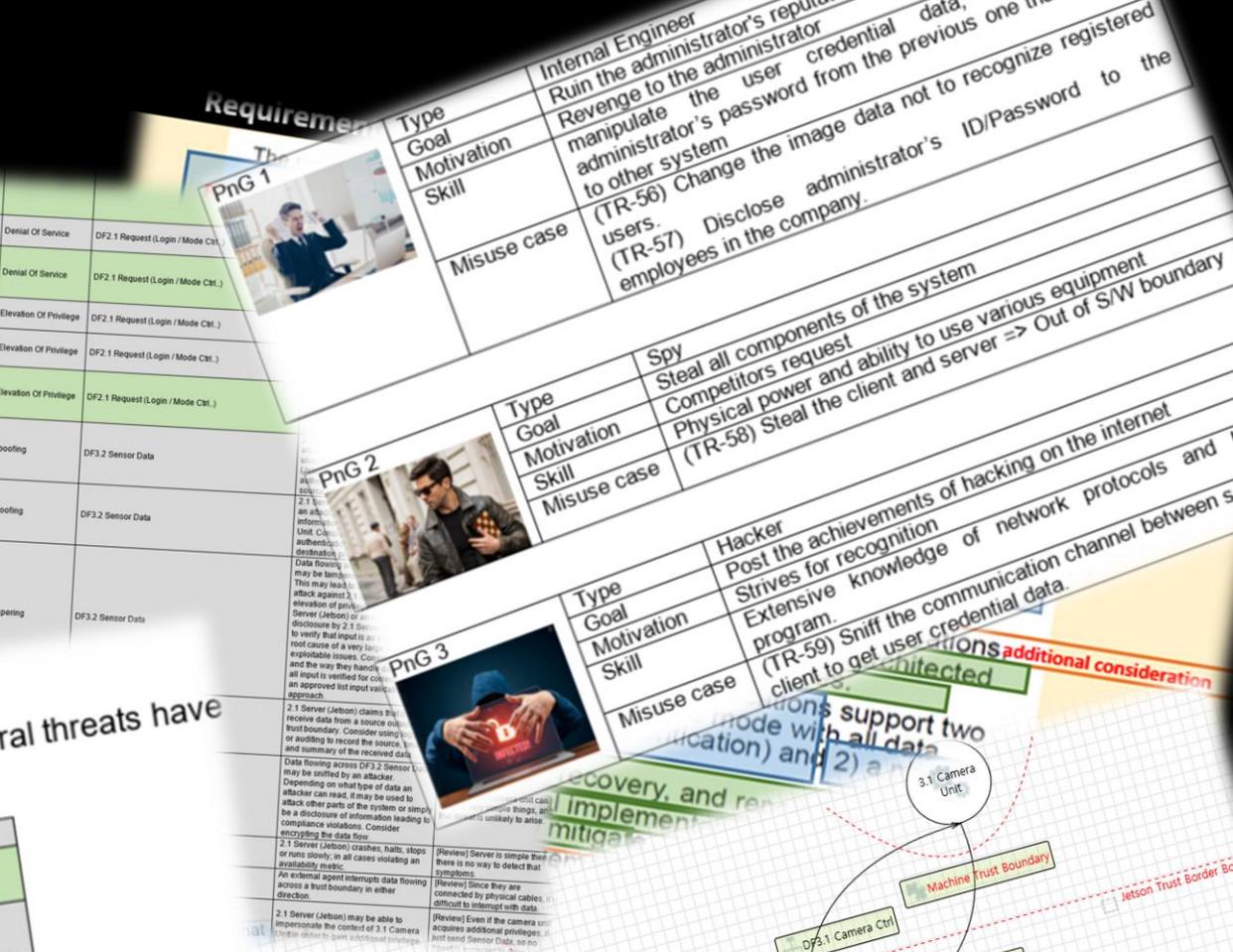
Project Plan & Schedule															
Week	Task	Phase I		Phase II		Phase III		Phase IV		Phase V		Phase VI		Phase VII	
		Start Date	End Date												
1 week	Context Analysis	2023-01-01	2023-01-07												
2 week	Risk Management	2023-01-08	2023-01-14												
3 week	Asset Identification	2023-01-15	2023-01-21												
4 week	Threat Modeling	2023-01-22	2023-01-28	2023-01-29	2023-02-04	2023-02-05	2023-02-11	2023-02-12	2023-02-13	2023-02-14	2023-02-15	2023-02-21	2023-02-22	2023-02-23	2023-02-24
5 week	BFD Modeling	2023-02-25	2023-02-28	2023-02-29	2023-03-03	2023-03-04	2023-03-10	2023-03-11	2023-03-12	2023-03-13	2023-03-14	2023-03-20	2023-03-21	2023-03-22	2023-03-23
6 week	Pro/Minuse case	2023-03-25	2023-03-28	2023-03-29	2023-03-30	2023-03-31	2023-04-06	2023-04-07	2023-04-08	2023-04-09	2023-04-10	2023-04-17	2023-04-18	2023-04-19	2023-04-20
7 week	STRIDE Brainstorming	2023-04-24	2023-04-27	2023-04-28	2023-04-29	2023-04-30	2023-05-06	2023-05-07	2023-05-08	2023-05-09	2023-05-10	2023-05-17	2023-05-18	2023-05-19	2023-05-20
8 week	SEC Risk Assessment	2023-05-22	2023-05-25	2023-05-26	2023-05-27	2023-05-28	2023-05-29	2023-05-30	2023-05-31	2023-06-01	2023-06-02	2023-06-09	2023-06-10	2023-06-11	2023-06-12
9 week	SEC Requirement Mitigation	2023-06-13	2023-06-16	2023-06-17	2023-06-18	2023-06-19	2023-06-25	2023-06-26	2023-06-27	2023-06-28	2023-06-29	2023-07-06	2023-07-07	2023-07-08	2023-07-09
10 week	Code impl.	2023-07-10	2023-07-13	2023-07-14	2023-07-15	2023-07-16	2023-07-22	2023-07-23	2023-07-24	2023-07-25	2023-07-26	2023-07-30	2023-07-31	2023-08-01	2023-08-02
11 week	Server(Jetson Nano)	2023-08-03	2023-08-06	2023-08-07	2023-08-08	2023-08-09	2023-08-15	2023-08-16	2023-08-17	2023-08-18	2023-08-19	2023-08-23	2023-08-24	2023-08-25	2023-08-26
12 week	Client(PC)	2023-08-27	2023-08-30	2023-08-31	2023-09-01	2023-09-02	2023-09-08	2023-09-09	2023-09-10	2023-09-11	2023-09-12	2023-09-16	2023-09-17	2023-09-18	2023-09-19
13 week	Architect Doc	2023-09-20	2023-09-23	2023-09-24	2023-09-25	2023-09-26	2023-09-27	2023-09-28	2023-09-29	2023-09-30	2023-10-01	2023-10-05	2023-10-06	2023-10-07	2023-10-08
14 week	Development	2023-10-09	2023-10-12	2023-10-13	2023-10-14	2023-10-15	2023-10-16	2023-10-17	2023-10-18	2023-10-19	2023-10-20	2023-10-24	2023-10-25	2023-10-26	2023-10-27
15 week	Code Review	2023-10-28	2023-10-31	2023-10-32	2023-10-33	2023-10-34	2023-11-01	2023-11-02	2023-11-03	2023-11-04	2023-11-05	2023-11-09	2023-11-10	2023-11-11	2023-11-12
16 week	Static Analysis	2023-11-13	2023-11-16	2023-11-17	2023-11-18	2023-11-19	2023-11-25	2023-11-26	2023-11-27	2023-11-28	2023-11-29	2023-11-30	2023-12-04	2023-12-05	2023-12-06
17 week	Flaw Finder	2023-12-07	2023-12-10	2023-12-11	2023-12-12	2023-12-13	2023-12-19	2023-12-20	2023-12-21	2023-12-22	2023-12-23	2023-12-27	2023-12-28	2023-12-29	2023-12-30
18 week	CPPChecker	2023-12-31	2023-01-03	2023-01-04	2023-01-05	2023-01-06	2023-01-12	2023-01-13	2023-01-14	2023-01-15	2023-01-16	2023-01-20	2023-01-21	2023-01-22	2023-01-23
19 week	Fuzz Testing	2023-01-24	2023-01-27	2023-01-28	2023-01-29	2023-01-30	2023-02-05	2023-02-06	2023-02-07	2023-02-08	2023-02-09	2023-02-13	2023-02-14	2023-02-15	2023-02-16
20 week	Penetration Test	2023-02-17	2023-02-20	2023-02-21	2023-02-22	2023-02-23	2023-02-29	2023-02-30	2023-02-31	2023-03-01	2023-03-02	2023-03-06	2023-03-07	2023-03-08	2023-03-09
21 week	Documentation	2023-03-10	2023-03-13	2023-03-14	2023-03-15	2023-03-16	2023-03-22	2023-03-23	2023-03-24	2023-03-25	2023-03-26	2023-03-30	2023-03-31	2023-04-01	2023-04-02
22 week	Review	2023-04-03	2023-04-06	2023-04-07	2023-04-08	2023-04-09	2023-04-15	2023-04-16	2023-04-17	2023-04-18	2023-04-19	2023-04-23	2023-04-24	2023-04-25	2023-04-26
23 week	Present	2023-04-27	2023-04-30	2023-05-01	2023-05-02	2023-05-03	2023-05-09	2023-05-10	2023-05-11	2023-05-12	2023-05-13	2023-05-17	2023-05-18	2023-05-19	2023-05-20
24 week	Fuzz	2023-05-21	2023-05-24	2023-05-25	2023-05-26	2023-05-27	2023-05-28	2023-05-29	2023-05-30	2023-05-31	2023-06-01	2023-06-05	2023-06-06	2023-06-07	2023-06-08
25 week	Req	2023-06-09	2023-06-12	2023-06-13	2023-06-14	2023-06-15	2023-06-16	2023-06-17	2023-06-18	2023-06-19	2023-06-20	2023-06-24	2023-06-25	2023-06-26	2023-06-27
26 week	REQ-D-01	2023-06-28	2023-07-01	2023-07-02	2023-07-03	2023-07-04	2023-07-05	2023-07-06	2023-07-07	2023-07-08	2023-07-09	2023-07-13	2023-07-14	2023-07-15	2023-07-16
27 week	DEVELOPMENT	2023-07-17	2023-07-20	2023-07-21	2023-07-22	2023-07-23	2023-07-24	2023-07-25	2023-07-26	2023-07-27	2023-07-28	2023-07-32	2023-07-33	2023-07-34	2023-07-35
28 week	Requirements from CMU documents on left	2023-07-36	2023-07-39	2023-07-40	2023-07-41	2023-07-42	2023-07-43	2023-07-44	2023-07-45	2023-07-46	2023-07-47	2023-07-51	2023-07-52	2023-07-53	2023-07-54
29 week	Tools but	2023-07-55	2023-07-58	2023-07-59	2023-07-60	2023-07-61	2023-07-62	2023-07-63	2023-07-64	2023-07-65	2023-07-66	2023-07-70	2023-07-71	2023-07-72	2023-07-73

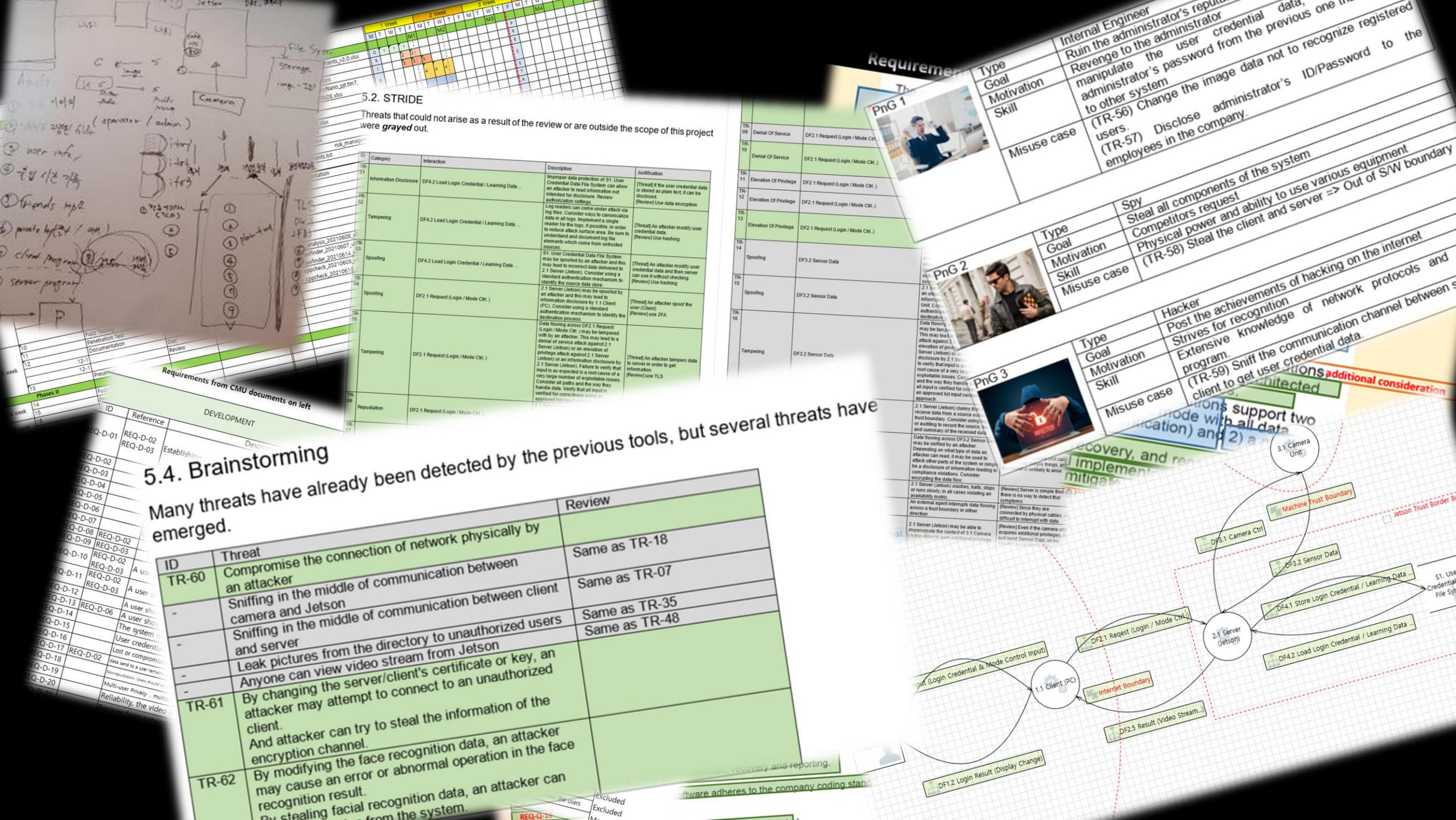
5.4. Brainstorming

5.4. Brainstorming

Many threats have already been detected by the previous tools, but several threats have emerged.

ID	Threat	Review
TR-60	Compromise the connection of network physically by an attacker	Same as TR-18
-	Sniffing in the middle of communication between camera and Jetson	Same as TR-07
-	Sniffing in the middle of communication between client and server	Same as TR-35
-	Leak pictures from the directory to unauthorized users	Same as TR-48
TR-61	By changing the server/client's certificate or key, an attacker may attempt to connect to an unauthorized client. And attacker can try to steal the information of the encryption channel.	
TR-62	By modifying the face recognition data, an attacker may cause an error or abnormal operation in the face recognition result. By stealing facial recognition data, an attacker can log in as the users from the system.	Excluded Excluded





5.2. STRIDE

Threats that could not arise as a result of the review or are outside the scope of this project were grayed out.

Requirements

ID	Category	Interaction	Description	Justification
TR-01	Information Disclosure	DF4.2 Load Login Credential / Learning Data ...	Improper data protection of S1 User Credential Data File System may allow an attacker to read information not intended for disclosure. Review authorizations.	[Thread] If the user credential data is stored as plain text, it can be disclosed [Review] Use data encryption
TR-02	Tampering	DF4.2 Load Login Credential / Learning Data ...	Log readers can come under attack via log files. Consider ways to canonicalize data from logs. Implement a single reader for all log types if possible. In order to reduce attack surface area, try to understand and document log file elements which come from untrusted sources.	[Thread] An attacker modify user credential data. [Review] Use hashing
TR-03	Spooing	DF4.2 Load Login Credential / Learning Data ...	S1 User Credential Data File System may be spoofed by an attacker and this may lead to sending data delivered to 2.1 Server (Jetson). Consider using a standard authentication mechanism to identify the source data store.	[Thread] An attacker modify user credential data and then server can use it without checking. [Review] Use hashing
TR-04	Spooing	DF2.1 Request (Login / Mode Ctrl.)	2.1 Server (Jetson) may be spoofed by an attacker and this may lead to information disclosure by 1.1 Client (PC). Consider using a standard authentication mechanism to identify the source data store.	[Thread] An attacker spoof the user (Client) [Review] use 2FA
TR-05	Tampering	DF2.1 Request (Login / Mode Ctrl.)	Data flowing across DF2.1 Request (Login / Mode Ctrl.) may be tampered with by an attacker. This may lead to a denial of service attack against 2.1 Server (Jetson) or an elevation of privilege against 2.1 Server (Jetson) or an information disclosure by 2.1 Server (Jetson). Failure to verify that input is as expected is a root cause of a very large number of security issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness testing in [Review] use TLS	[Thread] An attacker tampers data server in order to get information. [Review] use TLS
TR-06	Reputation	DF2.1 Request (Login / Mode Ctrl.)		

Requirements from CMU documents on left

ID	Reference	Development
REQ-D-01	REQ-D-02	Establishing
REQ-D-02	REQ-D-03	Documentation
REQ-D-03	REQ-D-04	Fuzz Testing
REQ-D-04	REQ-D-05	Penetration Test
REQ-D-05	REQ-D-06	Review
REQ-D-06	REQ-D-07	Present
REQ-D-07	REQ-D-08	Phases II
REQ-D-08	REQ-D-09	Documentation
REQ-D-09	REQ-D-10	Fuzz Testing
REQ-D-10	REQ-D-11	Review
REQ-D-11	REQ-D-12	Present
REQ-D-12	REQ-D-13	Phases II
REQ-D-13	REQ-D-14	Documentation
REQ-D-14	REQ-D-15	Fuzz Testing
REQ-D-15	REQ-D-16	Review
REQ-D-16	REQ-D-17	Present
REQ-D-17	REQ-D-18	Phases II
REQ-D-18	REQ-D-19	Documentation
REQ-D-19	REQ-D-20	Fuzz Testing
REQ-D-20	REQ-D-21	Review

5.4. Brainstorming

Many threats have already been detected by the previous tools, but several threats have emerged.

ID	Threat	Review
TR-60	Compromise the connection of network physically by an attacker Sniffing in the middle of communication between camera and Jetson Sniffing in the middle of communication between client and server Leak pictures from the directory to unauthorized users Anyone can view video stream from Jetson	Same as TR-18 Same as TR-07 Same as TR-35 Same as TR-48
TR-61	By changing the server/client's certificate or key, an attacker may attempt to connect to an unauthorized client. And attacker can try to steal the information of the encryption channel. By modifying the face recognition data, an attacker may cause an error or abnormal operation in the face recognition result. By stealing facial recognition data, an attacker can gain access from the system.	
TR-62		

Requirements

Type	Goal	Motivation	Skill	Internal Engineer
Denial Of Service	DF2.1 Request (Login / Mode Ctrl.)	Denial Of Service	Elevation Of Privilege	Ruin the administrator's reputation
Elevation Of Privilege	DF2.1 Request (Login / Mode Ctrl.)	Elevation Of Privilege	Elevation Of Privilege	manipulate the user credential to other system
Elevation Of Privilege	DF2.1 Request (Login / Mode Ctrl.)	Elevation Of Privilege	Elevation Of Privilege	(TR-56) Change the user password from the previous one to the new one
Spooing	DF3.2 Sensor Data	Spooing	Elevation Of Privilege	(TR-57) Disclose the administrator's ID/Password to the employees in the company
Elevation Of Privilege	DF3.2 Sensor Data	Elevation Of Privilege	Elevation Of Privilege	Steal all components of the system
Elevation Of Privilege	DF3.2 Sensor Data	Elevation Of Privilege	Elevation Of Privilege	Competitors request
Elevation Of Privilege	DF3.2 Sensor Data	Elevation Of Privilege	Elevation Of Privilege	(TR-58) Steal the client and server => Out of S/W boundary
Hacker		Hacker	Elevation Of Privilege	Post the achievements of hacking on the internet
Extensive Knowledge of network protocols and s		Extensive Knowledge of network protocols and s	Elevation Of Privilege	Program.
Sniff the communication channel between client and server		Sniff the communication channel between client and server	Elevation Of Privilege	additional consideration
Sniff the communication channel between client and server		Sniff the communication channel between client and server	Elevation Of Privilege	additional consideration
Machine Trust Boundary		Machine Trust Boundary	Elevation Of Privilege	Machine Trust Boundary
Jetson Trust Border		Jetson Trust Border	Elevation Of Privilege	Jetson Trust Border

Diagram

Notes

- Software adheres to the company coding standards
- REQ-Q-10
- REQ-Q-11
- REQ-Q-12
- REQ-Q-13
- REQ-Q-14
- REQ-Q-15
- REQ-Q-16
- REQ-Q-17
- REQ-Q-18
- REQ-Q-19
- REQ-Q-20
- REQ-Q-21
- REQ-Q-22
- REQ-Q-23
- REQ-Q-24
- REQ-Q-25
- REQ-Q-26
- REQ-Q-27
- REQ-Q-28
- REQ-Q-29
- REQ-Q-30
- REQ-Q-31
- REQ-Q-32
- REQ-Q-33
- REQ-Q-34
- REQ-Q-35
- REQ-Q-36
- REQ-Q-37
- REQ-Q-38
- REQ-Q-39
- REQ-Q-40
- REQ-Q-41
- REQ-Q-42
- REQ-Q-43
- REQ-Q-44
- REQ-Q-45
- REQ-Q-46
- REQ-Q-47
- REQ-Q-48
- REQ-Q-49
- REQ-Q-50
- REQ-Q-51
- REQ-Q-52
- REQ-Q-53
- REQ-Q-54
- REQ-Q-55
- REQ-Q-56
- REQ-Q-57
- REQ-Q-58
- REQ-Q-59
- REQ-Q-60
- REQ-Q-61
- REQ-Q-62
- REQ-Q-63
- REQ-Q-64
- REQ-Q-65
- REQ-Q-66
- REQ-Q-67
- REQ-Q-68
- REQ-Q-69
- REQ-Q-70
- REQ-Q-71
- REQ-Q-72
- REQ-Q-73
- REQ-Q-74
- REQ-Q-75
- REQ-Q-76
- REQ-Q-77
- REQ-Q-78
- REQ-Q-79
- REQ-Q-80
- REQ-Q-81
- REQ-Q-82
- REQ-Q-83
- REQ-Q-84
- REQ-Q-85
- REQ-Q-86
- REQ-Q-87
- REQ-Q-88
- REQ-Q-89
- REQ-Q-90
- REQ-Q-91
- REQ-Q-92
- REQ-Q-93
- REQ-Q-94
- REQ-Q-95
- REQ-Q-96
- REQ-Q-97
- REQ-Q-98
- REQ-Q-99
- REQ-Q-100
- REQ-Q-101
- REQ-Q-102
- REQ-Q-103
- REQ-Q-104
- REQ-Q-105
- REQ-Q-106
- REQ-Q-107
- REQ-Q-108
- REQ-Q-109
- REQ-Q-110
- REQ-Q-111
- REQ-Q-112
- REQ-Q-113
- REQ-Q-114
- REQ-Q-115
- REQ-Q-116
- REQ-Q-117
- REQ-Q-118
- REQ-Q-119
- REQ-Q-120
- REQ-Q-121
- REQ-Q-122
- REQ-Q-123
- REQ-Q-124
- REQ-Q-125
- REQ-Q-126
- REQ-Q-127
- REQ-Q-128
- REQ-Q-129
- REQ-Q-130
- REQ-Q-131
- REQ-Q-132
- REQ-Q-133
- REQ-Q-134
- REQ-Q-135
- REQ-Q-136
- REQ-Q-137
- REQ-Q-138
- REQ-Q-139
- REQ-Q-140
- REQ-Q-141
- REQ-Q-142
- REQ-Q-143
- REQ-Q-144
- REQ-Q-145
- REQ-Q-146
- REQ-Q-147
- REQ-Q-148
- REQ-Q-149
- REQ-Q-150
- REQ-Q-151
- REQ-Q-152
- REQ-Q-153
- REQ-Q-154
- REQ-Q-155
- REQ-Q-156
- REQ-Q-157
- REQ-Q-158
- REQ-Q-159
- REQ-Q-160
- REQ-Q-161
- REQ-Q-162
- REQ-Q-163
- REQ-Q-164
- REQ-Q-165
- REQ-Q-166
- REQ-Q-167
- REQ-Q-168
- REQ-Q-169
- REQ-Q-170
- REQ-Q-171
- REQ-Q-172
- REQ-Q-173
- REQ-Q-174
- REQ-Q-175
- REQ-Q-176
- REQ-Q-177
- REQ-Q-178
- REQ-Q-179
- REQ-Q-180
- REQ-Q-181
- REQ-Q-182
- REQ-Q-183
- REQ-Q-184
- REQ-Q-185
- REQ-Q-186
- REQ-Q-187
- REQ-Q-188
- REQ-Q-189
- REQ-Q-190
- REQ-Q-191
- REQ-Q-192
- REQ-Q-193
- REQ-Q-194
- REQ-Q-195
- REQ-Q-196
- REQ-Q-197
- REQ-Q-198
- REQ-Q-199
- REQ-Q-200
- REQ-Q-201
- REQ-Q-202
- REQ-Q-203
- REQ-Q-204
- REQ-Q-205
- REQ-Q-206
- REQ-Q-207
- REQ-Q-208
- REQ-Q-209
- REQ-Q-210
- REQ-Q-211
- REQ-Q-212
- REQ-Q-213
- REQ-Q-214
- REQ-Q-215
- REQ-Q-216
- REQ-Q-217
- REQ-Q-218
- REQ-Q-219
- REQ-Q-220
- REQ-Q-221
- REQ-Q-222
- REQ-Q-223
- REQ-Q-224
- REQ-Q-225
- REQ-Q-226
- REQ-Q-227
- REQ-Q-228
- REQ-Q-229
- REQ-Q-230
- REQ-Q-231
- REQ-Q-232
- REQ-Q-233
- REQ-Q-234
- REQ-Q-235
- REQ-Q-236
- REQ-Q-237
- REQ-Q-238
- REQ-Q-239
- REQ-Q-240
- REQ-Q-241
- REQ-Q-242
- REQ-Q-243
- REQ-Q-244
- REQ-Q-245
- REQ-Q-246
- REQ-Q-247
- REQ-Q-248
- REQ-Q-249
- REQ-Q-250
- REQ-Q-251
- REQ-Q-252
- REQ-Q-253
- REQ-Q-254
- REQ-Q-255
- REQ-Q-256
- REQ-Q-257
- REQ-Q-258
- REQ-Q-259
- REQ-Q-260
- REQ-Q-261
- REQ-Q-262
- REQ-Q-263
- REQ-Q-264
- REQ-Q-265
- REQ-Q-266
- REQ-Q-267
- REQ-Q-268
- REQ-Q-269
- REQ-Q-270
- REQ-Q-271
- REQ-Q-272
- REQ-Q-273
- REQ-Q-274
- REQ-Q-275
- REQ-Q-276
- REQ-Q-277
- REQ-Q-278
- REQ-Q-279
- REQ-Q-280
- REQ-Q-281
- REQ-Q-282
- REQ-Q-283
- REQ-Q-284
- REQ-Q-285
- REQ-Q-286
- REQ-Q-287
- REQ-Q-288
- REQ-Q-289
- REQ-Q-290
- REQ-Q-291
- REQ-Q-292
- REQ-Q-293
- REQ-Q-294
- REQ-Q-295
- REQ-Q-296
- REQ-Q-297
- REQ-Q-298
- REQ-Q-299
- REQ-Q-300
- REQ-Q-301
- REQ-Q-302
- REQ-Q-303
- REQ-Q-304
- REQ-Q-305
- REQ-Q-306
- REQ-Q-307
- REQ-Q-308
- REQ-Q-309
- REQ-Q-310
- REQ-Q-311
- REQ-Q-312
- REQ-Q-313
- REQ-Q-314
- REQ-Q-315
- REQ-Q-316
- REQ-Q-317
- REQ-Q-318
- REQ-Q-319
- REQ-Q-320
- REQ-Q-321
- REQ-Q-322
- REQ-Q-323
- REQ-Q-324
- REQ-Q-325
- REQ-Q-326
- REQ-Q-327
- REQ-Q-328
- REQ-Q-329
- REQ-Q-330
- REQ-Q-331
- REQ-Q-332
- REQ-Q-333
- REQ-Q-334
- REQ-Q-335
- REQ-Q-336
- REQ-Q-337
- REQ-Q-338
- REQ-Q-339
- REQ-Q-340
- REQ-Q-341
- REQ-Q-342
- REQ-Q-343
- REQ-Q-344
- REQ-Q-345
- REQ-Q-346
- REQ-Q-347
- REQ-Q-348
- REQ-Q-349
- REQ-Q-350
- REQ-Q-351
- REQ-Q-352
- REQ-Q-353
- REQ-Q-354
- REQ-Q-355
- REQ-Q-356
- REQ-Q-357
- REQ-Q-358
- REQ-Q-359
- REQ-Q-360
- REQ-Q-361
- REQ-Q-362
- REQ-Q-363
- REQ-Q-364
- REQ-Q-365
- REQ-Q-366
- REQ-Q-367
- REQ-Q-368
- REQ-Q-369
- REQ-Q-370
- REQ-Q-371
- REQ-Q-372
- REQ-Q-373
- REQ-Q-374
- REQ-Q-375
- REQ-Q-376
- REQ-Q-377
- REQ-Q-378
- REQ-Q-379
- REQ-Q-380
- REQ-Q-381
- REQ-Q-382
- REQ-Q-383
- REQ-Q-384
- REQ-Q-385
- REQ-Q-386
- REQ-Q-387
- REQ-Q-388
- REQ-Q-389
- REQ-Q-390
- REQ-Q-391
- REQ-Q-392
- REQ-Q-393
- REQ-Q-394
- REQ-Q-395
- REQ-Q-396
- REQ-Q-397
- REQ-Q-398
- REQ-Q-399
- REQ-Q-400
- REQ-Q-401
- REQ-Q-402
- REQ-Q-403
- REQ-Q-404
- REQ-Q-405
- REQ-Q-406
- REQ-Q-407
- REQ-Q-408
- REQ-Q-409
- REQ-Q-410
- REQ-Q-411
- REQ-Q-412
- REQ-Q-413
- REQ-Q-414
- REQ-Q-415
- REQ-Q-416
- REQ-Q-417
- REQ-Q-418
- REQ-Q-419
- REQ-Q-420
- REQ-Q-421
- REQ-Q-422
- REQ-Q-423
- REQ-Q-424
- REQ-Q-425
- REQ-Q-426
- REQ-Q-427
- REQ-Q-428
- REQ-Q-429
- REQ-Q-430
- REQ-Q-431
- REQ-Q-432
- REQ-Q-433
- REQ-Q-434
- REQ-Q-435
- REQ-Q-436
- REQ-Q-437
- REQ-Q-438
- REQ-Q-439
- REQ-Q-440
- REQ-Q-441
- REQ-Q-442
- REQ-Q-443
- REQ-Q-444
- REQ-Q-445
- REQ-Q-446
- REQ-Q-447
- REQ-Q-448
- REQ-Q-449
- REQ-Q-450
- REQ-Q-451
- REQ-Q-452
- REQ-Q-453
- REQ-Q-454
- REQ-Q-455
- REQ-Q-456
- REQ-Q-457
- REQ-Q-458
- REQ-Q-459
- REQ-Q-460
- REQ-Q-461
- REQ-Q-462
- REQ-Q-463
- REQ-Q-464
- REQ-Q-465
- REQ-Q-466
- REQ-Q-467
- REQ-Q-468
- REQ-Q-469
- REQ-Q-470
- REQ-Q-471
- REQ-Q-472
- REQ-Q-473
- REQ-Q-474
- REQ-Q-475
- REQ-Q-476
- REQ-Q-477
- REQ-Q-478
- REQ-Q-479
- REQ-Q-480
- REQ-Q-481
- REQ-Q-482
- REQ-Q-483
- REQ-Q-484
- REQ-Q-485
- REQ-Q-486
- REQ-Q-487
- REQ-Q-488
- REQ-Q-489
- REQ-Q-490
- REQ-Q-491
- REQ-Q-492
- REQ-Q-493
- REQ-Q-494
- REQ-Q-495
- REQ-Q-496
- REQ-Q-497
- REQ-Q-498
- REQ-Q-499
- REQ-Q-500
- REQ-Q-501
- REQ-Q-502
- REQ-Q-503
- REQ-Q-504
- REQ-Q-505
- REQ-Q-506
- REQ-Q-507
- REQ-Q-508
- REQ-Q-509
- REQ-Q-510
- REQ-Q-511
- REQ-Q-512
- REQ-Q-513
- REQ-Q-514
- REQ-Q-515
- REQ-Q-516
- REQ-Q-517
- REQ-Q-518
- REQ-Q-519
- REQ-Q-520
- REQ-Q-521
- REQ-Q-522
- REQ-Q-523
- REQ-Q-524
- REQ-Q-525
- REQ-Q-526
- REQ-Q-527
- REQ-Q-528
- REQ-Q-529
- REQ-Q-530
- REQ-Q-531
- REQ-Q-532
- REQ-Q-533
- REQ-Q-534
- REQ-Q-535
- REQ-Q-536
- REQ-Q-537
- REQ-Q-538
- REQ-Q-539
- REQ-Q-540
- REQ-Q-541
- REQ-Q-542
- REQ-Q-543
- REQ-Q-544
- REQ-Q-545
- REQ-Q-546
- REQ-Q-547
- REQ-Q-548
- REQ-Q-549
- REQ-Q-550
- REQ-Q-551
- REQ-Q-552
- REQ-Q-553
- REQ-Q-554
- REQ-Q-555
- REQ-Q-556
- REQ-Q-557
- REQ-Q-558
- REQ-Q-559
- REQ-Q-560
- REQ-Q-561
- REQ-Q-562
- REQ-Q-563
- REQ-Q-564
- REQ-Q-565
- REQ-Q-566
- REQ-Q-567
- REQ-Q-568
- REQ-Q-569
- REQ-Q-570
- REQ-Q-571
- REQ-Q-572
- REQ-Q-573
- REQ-Q-574
- REQ-Q-575
- REQ-Q-576
- REQ-Q-577
- REQ-Q-578
- REQ-Q-579
- REQ-Q-580
- REQ-Q-581
- REQ-Q-582
- REQ-Q-583
- REQ-Q-584
- REQ-Q-585
- REQ-Q-586
- REQ-Q-587
- REQ-Q-588
- REQ-Q-589
- REQ-Q-590
- REQ-Q-591
- REQ-Q-592
- REQ-Q-593
- REQ-Q-594
- REQ-Q-595
- REQ-Q-596
- REQ-Q-597
- REQ-Q-598
- REQ-Q-599
- REQ-Q-600
- REQ-Q-601
- REQ-Q-602
- REQ-Q-603
- REQ-Q-604
- REQ-Q-605
- REQ-Q-606
- REQ-Q-607
- REQ-Q-608
- REQ-Q-609
- REQ-Q-610
- REQ-Q-611
- REQ-Q-612
- REQ-Q-613
- REQ-Q-614
- REQ-Q-615
- REQ-Q-616
- REQ-Q-617
- REQ-Q-618
- REQ-Q-619
- REQ-Q-620
- REQ-Q-621
- REQ-Q-622
- REQ-Q-623
- REQ-Q-624
- REQ-Q-625
- REQ-Q-626
- REQ-Q-627
- REQ-Q-628
- REQ-Q-629
- REQ-Q-630
- REQ-Q-631
- REQ-Q-632
- REQ-Q-633
- REQ-Q-634
- REQ-Q-635
- REQ-Q-636
- REQ-Q-637
- REQ-Q-638
- REQ-Q-639
- REQ-Q-640
- REQ-Q-641
- REQ-Q-642
- REQ-Q-643
- REQ-Q-644
- REQ-Q-645
- REQ-Q-646
- REQ-Q-647
- REQ-Q-648
- REQ-Q-649
- REQ-Q-650
- REQ-Q-651
- REQ-Q-652
- REQ-Q-653
- REQ-Q-654
- REQ-Q-655
- REQ-Q-656
- REQ-Q-657
- REQ-Q-658
- REQ-Q-659
- REQ-Q-660
- REQ-Q-661
- REQ-Q-662
- REQ-Q-663
- REQ-Q-664
- REQ-Q-665
- REQ-Q-666
- REQ-Q-667
- REQ-Q-668
- REQ-Q-669
- REQ-Q-670
- REQ-Q-671
- REQ-Q-672
- REQ-Q-673
- REQ-Q-674
- REQ-Q-675
- REQ-Q-676
- REQ-Q-677
- REQ-Q-678
- REQ-Q-679
- REQ-Q-680
- REQ-Q-681
- REQ-Q-682
- REQ-Q-683
- REQ-Q-684
- REQ-Q-685
- REQ-Q-686
- REQ-Q-687
- REQ-Q-688
- REQ-Q-689
- REQ-Q-690
- REQ-Q-691
- REQ-Q-692
- REQ-Q-693
- REQ-Q-694
- REQ-Q-695
- REQ-Q-696
- REQ-Q-697
- REQ-Q-698
- REQ-Q-699
- REQ-Q-700
- REQ-Q-701
- REQ-Q-702
- REQ-Q-703
- REQ-Q-704
- REQ-Q-705
- REQ-Q-706
- REQ-Q-707
- REQ-Q-708
- REQ-Q-709
- REQ-Q-710
- REQ-Q-711
- REQ-Q-712
- REQ-Q-713
- REQ-Q-714
- REQ-Q-715
- REQ-Q-716
- REQ-Q-717
- REQ-Q-718
- REQ-Q-719
- REQ-Q-720
- REQ-Q-721
- REQ-Q-722
- REQ-Q-723
- REQ-Q-724
- REQ-Q-725
- REQ-Q-726
- REQ-Q-727
- REQ-Q-728
- REQ-Q-729
- REQ-Q-730
- REQ-Q-731
- REQ-Q-732
- REQ-Q-733
- REQ-Q-734
- REQ-Q-735
- REQ-Q-736
- REQ-Q-737
- REQ-Q-738
- REQ-Q-739
- REQ-Q-740
- REQ-Q-741
- REQ-Q-742
- REQ-Q-743
- REQ-Q-744
- REQ-Q-745
- REQ-Q-746
- REQ-Q-747
- REQ-Q-748
- REQ-Q-749
- REQ-Q-750
- REQ-Q-751
- REQ-Q-752
- REQ-Q-753
- REQ-Q-754
- REQ-Q-755
- REQ-Q-756
- REQ-Q-757
- REQ-Q-758
- REQ-Q-759
- REQ-Q-760
- REQ-Q-761
- REQ-Q-762
- REQ-Q-763
- REQ-Q-764
- REQ-Q-765
- REQ-Q-766
- REQ-Q-767
- REQ-Q-768
- REQ-Q-769
- REQ-Q-770
- REQ-Q-771
- REQ-Q-772
- REQ-Q-773
- REQ-Q-774
- REQ-Q-775
- REQ-Q-776
- REQ-Q-777
- REQ-Q-778
- REQ-Q-779
- REQ-Q-780
- REQ-Q-781
- REQ-Q-782
- REQ-Q-783
-

Handwritten notes and sketches	A grid calendar showing tasks over 3 weeks, with some items crossed out.	A slide titled "5.2. STRIDE" with a note: "Threats that could not arise as a result of the review or are outside the scope of this project were grayed out."	A table of threats categorized by Type, Goal, Motivation, Skill, and Misuse case. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).	A table of threats categorized by Type, Goal, Motivation, Skill, and Misuse case. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).
Handwritten notes and sketches	A diagram of a system architecture with various components like Camera, Storage, and Network. It shows data flow and potential attack paths.	A table of threats categorized by ID, Category, Interaction, Description, and Justification. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).	A table of threats categorized by ID, Category, Interaction, Threat, Review, and Additional Consideration. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).	A table of threats categorized by ID, Category, Interaction, Threat, Review, and Additional Consideration. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).
Handwritten notes and sketches	A table of requirements from CMU documents on left.	A table of threats categorized by ID, Category, Interaction, Threat, Review, and Additional Consideration. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).	A table of threats categorized by ID, Category, Interaction, Threat, Review, and Additional Consideration. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).	A table of threats categorized by ID, Category, Interaction, Threat, Review, and Additional Consideration. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).
Handwritten notes and sketches	A table of requirements from CMU documents on left.	A table of threats categorized by ID, Category, Interaction, Threat, Review, and Additional Consideration. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).	A table of threats categorized by ID, Category, Interaction, Threat, Review, and Additional Consideration. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).	A table of threats categorized by ID, Category, Interaction, Threat, Review, and Additional Consideration. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).
Handwritten notes and sketches	A table of requirements from CMU documents on left.	A table of threats categorized by ID, Category, Interaction, Threat, Review, and Additional Consideration. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).	A table of threats categorized by ID, Category, Interaction, Threat, Review, and Additional Consideration. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).	A table of threats categorized by ID, Category, Interaction, Threat, Review, and Additional Consideration. It includes screenshots of Png 1 (Denial of Service) and Png 2 (Elevation of Privilege).

5.4. Brainstorming

Many threats have already been detected by the previous tools, but several new ones have emerged.

ID	Threat	Review
TR-60	Compromise the connection of network physically by an attacker Sniffing in the middle of communication between camera and Jetson Sniffing in the middle of communication between client and server Leak pictures from the directory to unauthorized users Anyone can view video stream from Jetson	Same as TR-18 Same as TR-07 Same as TR-35 Same as TR-48
TR-61	By changing the server/client's certificate or key, an attacker may attempt to connect to an unauthorized client. And attacker can try to steal the information of the encryption channel.	
TR-62	By modifying the face recognition data, an attacker may cause an error or abnormal operation in the face recognition result. By stealing facial recognition data, an attacker can get data from the system.	

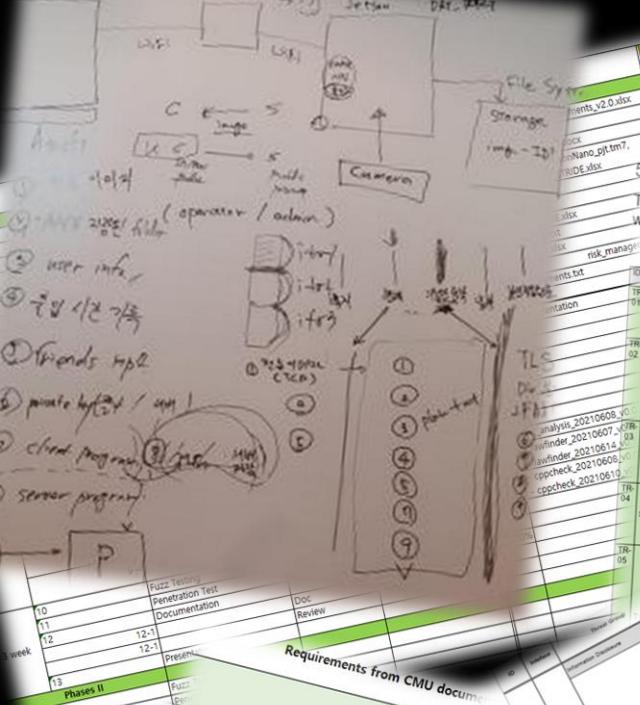
ID	Threat	Review
TR-60	Compromise the connection of network physically by an attacker Sniffing in the middle of communication between camera and Jetson Sniffing in the middle of communication between client and server Leak pictures from the directory to unauthorized users Anyone can view video stream from Jetson	Same as TR-18 Same as TR-07 Same as TR-35 Same as TR-48
TR-61	By changing the server/client's certificate or key, an attacker may attempt to connect to an unauthorized client. And attacker can try to steal the information of the encryption channel.	
TR-62	By modifying the face recognition data, an attacker may cause an error or abnormal operation in the face recognition result. By stealing facial recognition data, an attacker can get data from the system.	

ID	Threat	Review
TR-60	Compromise the connection of network physically by an attacker Sniffing in the middle of communication between camera and Jetson Sniffing in the middle of communication between client and server Leak pictures from the directory to unauthorized users Anyone can view video stream from Jetson	Same as TR-18 Same as TR-07 Same as TR-35 Same as TR-48
TR-61	By changing the server/client's certificate or key, an attacker may attempt to connect to an unauthorized client. And attacker can try to steal the information of the encryption channel.	
TR-62	By modifying the face recognition data, an attacker may cause an error or abnormal operation in the face recognition result. By stealing facial recognition data, an attacker can get data from the system.	

ID	Threat	Review
TR-60	Compromise the connection of network physically by an attacker Sniffing in the middle of communication between camera and Jetson Sniffing in the middle of communication between client and server Leak pictures from the directory to unauthorized users Anyone can view video stream from Jetson	Same as TR-18 Same as TR-07 Same as TR-35 Same as TR-48
TR-61	By changing the server/client's certificate or key, an attacker may attempt to connect to an unauthorized client. And attacker can try to steal the information of the encryption channel.	
TR-62	By modifying the face recognition data, an attacker may cause an error or abnormal operation in the face recognition result. By stealing facial recognition data, an attacker can get data from the system.	

ID	Threat	Review
TR-60	Compromise the connection of network physically by an attacker Sniffing in the middle of communication between camera and Jetson Sniffing in the middle of communication between client and server Leak pictures from the directory to unauthorized users Anyone can view video stream from Jetson	Same as TR-18 Same as TR-07 Same as TR-35 Same as TR-48
TR-61	By changing the server/client's certificate or key, an attacker may attempt to connect to an unauthorized client. And attacker can try to steal the information of the encryption channel.	
TR-62	By modifying the face recognition data, an attacker may cause an error or abnormal operation in the face recognition result. By stealing facial recognition data, an attacker can get data from the system.	

ID	Threat	Review
TR-60	Compromise the connection of network physically by an attacker Sniffing in the middle of communication between camera and Jetson Sniffing in the middle of communication between client and server Leak pictures from the directory to unauthorized users Anyone can view video stream from Jetson	Same as TR-18 Same as TR-07 Same as TR-35 Same as TR-48
TR-61	By changing the server/client's certificate or key, an attacker may attempt to connect to an unauthorized client. And attacker can try to steal the information of the encryption channel.	
TR-62	By modifying the face recognition data, an attacker may cause an error or abnormal operation in the face recognition result. By stealing facial recognition data, an attacker can get data from the system.	



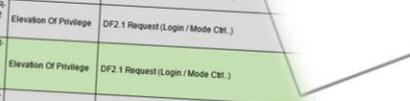
5.2. STRIDE

Threats that could not arise as a result of the review or are outside the scope of this project were **grayed** out.

ID	Category	Interaction	Description	Justification
TR-01	Information Disclosure	DF4.2 Load Login Credential / Learning Data ...	Improper data protection of S1. User Credential Data File System can allow an attacker to read information not intended for disclosure. Review authorization settings.	[Thread] If the user credential data is stored as plain text, it can be disclosed. [Review] Use data encryption
TR-02	Tampering	DF4.2 Load Login Credential / Learning Data ...	Log information can come under attack via log files. Consider using a centralized log file. Implement a single reader for the logs. If possible, in order to reduce the attack surface. Be sure to understand and document log file elements which come from untrusted sources.	[Thread] An attacker modify user credential data. [Review] Use hashing
TR-03	Spoofting	DF4.2 Load Login Credential / Learning Data ...	User Credential Data File System may be spoofed by an attacker and thus may lead to incorrect data delivered to the server (Jeton). Considering a user verification mechanism to verify the source data. The token (Jeton) may be spoofed by an attacker and this may lead to disclosure by 1. Client user and 2. standard mechanism to identify the user.	[Thread] An attacker modify user credential data and then server can use it for unauthorized checking. [Review] Use hashing
TR-04	Spoofting	DF2.1 Request (r)	DF2.1 Request may be tampered with by an attacker and thus may lead to incorrect data delivered to the server (Jeton). Considering a user verification mechanism to verify the source data. The token (Jeton) may be spoofed by an attacker and this may lead to disclosure by 1. Client user and 2. standard mechanism to identify the user.	[Thread] An attacker spoof the user (Client) [Review] use 2FA
TR-05				

TR-
62 STRIDE Tamp

Requirer



Type	Ruin the administrator's reputation
Goal	Revenge to the administrator
Motivation	manipulate the user credential data, administrator's password from the previous one to other system
Skill	(TR-56) Change the image data not to recognize registered users. (TR-57) Disclose administrator's ID/Password to the employees in the company.
Misuse case	components of the system not to use various equipment => Out of S/W boundary

Type	Steal air traffic control system
Goal	Competitors require it
Motivation	Physical power and ability
Skill	(TR-58) Steal the client and server
Misuse case	

5.5. Result of Threat Modeling

We found 28 threats below by using STRIDE, PnG, Brainstorming.

Identify threats below by using STRIDE, PnG, Brainstorming.					
Tool	Category	Interaction	Threat	Review	
TR-01	STRIDE	Information Disclosure	DF4.2 Load Login Credential / Learning Data ...	If the user credential data is stored as plain text, it can be disclosed.	User credential should be kept securely
R-02	STRIDE	Tampering	DF4.2 Load Login Credential / Learning Data ...	An attacker modify user credential data	User credential should be kept securely
	STRIDE	Spoofing	DF4.2 Load Login Credential / Learning Data ...	An attacker modify user credential data and then server accept it without checking	User credential should be kept securely
	STRIDE	Spoofing	DF2.1 Request (Login / Mode Ctrl.)	An attacker spoof the user (Client)	Need to more stronger authentication process
	STRIDE	Tampering	DF2.1 Request (Login / Mode Ctrl.)	An attacker tampers Login or Mode control data to server in order to get info from server	Need to encrypt communication channel
	STRIDE	Repudiation	DF2.1 Request (Login / Mode Ctrl.)	Clients can repudiate the actions they have performed.	Need to apply mutual authentication
	STRIDE	Information Disclosure	DF2.1 Request (Login / Mode Ctrl.)	An attack can sniff the data on the connection.	Need to consider encrypting the data flow
	STRIDE	Information Disclosure	DF2.1 Request (Login / Mode Ctrl.)	Weak authentication may lead to disclosure of information	Need to more stronger authentication process
E	Denial Of Service		DF2.1 Request (Login / Mode Ctrl.)	The information of the communication between client and server is tampered by attacker	Need to use TLS
		Elevation Of Privilege	DF2.1 Request (Login / Mode Ctrl.)	An attacker sends a malicious data to server in order to change the flow of program execution.	Need to apply input sanitization
	Denial Of Service		DF3.1 Camera Ctrl	It may be physically damaged and you may not be able to get Data from Camera	Need to protect camera unit from physical damage
	Denial Of Service			It is possible to ...	
To the company coding standards					
19 of 20 State Login Credential / Learning Data					

Sniff the network traffic (59) Sniff the user credentials (60) Sniff the session tokens (61) additional consideration					
TR-35	STRIDE	Information Disclosure	DF4.1 Store Login Credential / Learning Data	Images in the storage	
TR-41	STRIDE	Spoofting	DF4.1 Store Login Credential / Learning Data	User credential may be disclosed.	Need to encode credential data
TR-44	STRIDE	Spoofting	DF2.5 Result (Video Stream...)	User Credential Data can be exposed to attacker	Need to encode credential data
TR-45	STRIDE	Spoofting	DF2.5 Result (Video Stream...)	Server (Jession) may be spoofed by an attacker	Need to apply authentication
TR-46	STRIDE	Tampering	DF2.5 Result (Video Stream...)	Client (PC) may be spoofed by an attacker	Need to apply authentication
TR-48	STRIDE	Information Disclosure	DF2.5 Result (Video Stream...)	Video Stream may be tampered with by an attacker	Need to protect video stream over connection
TR-49	STRIDE	Denial Of Service	DF2.5 Result (Video Stream...)	Video Stream may be sniffed with an attack	Need to protect video stream over connection
TR-52	STRIDE	Elevation Of Privilege	DF2.5 Result (Video Stream...)	Client (PC) crashes, halts, stops or runs slowly	Need to remain stable in abnormal cases
TR-53	STRIDE	Elevation Of Privilege	DF2.5 Result (Video Stream...)	Server (Jession) may be able to remember encoded code	Need input sanitization
TR-56	PrG	Tampering	DF2.5 Result (Video Stream...)	An attacker may pass data into the PC Client (PC).	Need input sanitization
R-7	PrG	Information Disclosure	User credential data	Change the image data not to recognize registered users	Need to protect user credential data
	PrG	Information Disclosure	Client => Server	Disclose administrator's ID/Password to the employees of the company	Need to more stronger process for authentication
	Brainstorming	N/A	Server <=> Client	Sniff the communication between server and client to get user credential data	Need to protect the data over the connection
rainstorming		Tampering/ Information Disclosure/ Spoofting	Network	Compromise the connection of network physically by an attacker	Server need to be robust in abnormal case
			Server <=> Client	By changing the certificates certificate key, an attacker may attempt to connect unauthorized clients	
				And a message GSR (Get Session Response) is sent to the server	Need to protect or verify the certificates keys used by the server and client for TLS communication

Security Requirements

We've derived the security requirements through the STRIDE methodology. And we found out some of security requirements are linked to system requirements, section 2 above.

5.2. STRIDE

~~Threats that could m
were **grayed** out.~~

R-ID	Security Requirement	Mapping
R-01	A strong authentication method should be used.	CMU-REC
R-02	Cryptographically strong password should be used.	CMU-REC
R-03	Errors, exceptions, and abnormal conditions that may occur in the software must be handled robustly.	CMU-REC
SR-04	Input validation check is required in Client side.	
SR-05	Only the verified server and client should be connected and communicated.	
SR-06	Protect Camera from physical damage	
SR-07	Restrictions related to files are necessary to avoid system problems.	
SR-08	Save contents of the communication as a log and use as proof of non-repudiation.	
SR-09	Server and client must communicate using an encrypted channel.	CMU-R
SR-10	The system must perform an integrity check before using user credentials.	
SR-11	The system shall know the change of the user credential data.	
SR-12	Use well-known cryptographic libraries and robust algorithms.	
SR-13	User Credential Data should be encrypted in the storage.	CMU-I
SR-14	Video Stream over the connection should be protected.	
SR-15	A server and client program must perform an integrity check before using a certificate or key.	
SR-16	Face recognition data should be encrypted in the storage.	
SR-17	Every encryption time, newly generated random key is used for encryption to make reverse analysis difficult	
SR-18	ROOT encrypt key must be protected from binary analysis	

with system requirement	Mitigation ID
Q-D-09	MI-10
	MI-01
Q-D-15	MI-04
	MI-05
	MI-11
	MI-08
	MI-12
	MI-09
EQ-D-02	MI-02
	MI-07
	MI-07
	MI-03, MI-07
REQ-D-10	MI-03
	MI-02
	MI-13
	MI-06
	MI-14
	MI-15

Vulnerability		Attack Scenario		Mitigation / Learning Data	
Session Hijacking	An attacker modify user credential data	User credential should be kept securely		DF2.5 Result (Video Stream...)	Data can be exposed to attackers.
Session Hijacking	An attacker modify user credential data and then server can't detect it without checking	User credential should be kept securely		DF2.5 Result (Video Stream...)	User (Client) may be spoofed by an attacker
Mode Ctrl. (Client)	An attacker spoofs the user (Client)	Need to more stronger authentication process	STRIDE	Information Disclosure	Client (PC) may be spoofed by an attacker
Mode Ctrl. (Client)	An attacker tampers Login or Mode control data to server in order to get information	Need to encrypt communication channel	STRIDE	Denial Of Service	Video Stream may be tampered with by an attacker
Mode Ctrl. (Client)	Clients can replicate the actions they have performed	Need to apply mutual authentication	STRIDE	Elevation Of Privilege	Video Stream may be sniffed with an sniffer
Cat. 1	An attack can sniff the data on the connection.	Need to consider encrypting the data flow.	STRIDE	Elevation Of Privilege	Client (PC) crashes, halts, stops or runs unusually
Cat. 1	Weak authentication may lead to disclosure information	Need to more stronger authentication process	STRIDE	Tampering	Server (Server) may be able to remotely execute code
Cat. 1	The information of the communication between client and server is interrupted by third parties	Need to use TLS	PnG	Information Disclosure	An attacker may pass data into 1.1 Client (PC)
Cat. 1	An attacker sends a malicious data to server in order to change the flow of program execution	Need to apply input sanitization	PnG	Information Disclosure	Change the image data not to recognize registered users
Cat. 1	It may be physically damaged and you may not be able to get Data from Camera	Need to protect camera unit from physical damage	PnG	Information Disclosure	Disclose administrator's ID/Password to the employees in the company
Cat. 1	If it is possible to	Need to limit the	Brainstorming	N/A	Shift the communication channel between server and client to get user credential data
Cat. 1			Brainstorming	Tampering/ Information Disclosure/ Spoofing	Compromise the connection of network physically by using man-in-the-middle
Cat. 1			Brainstorming	Tampering/ Information Disclosure/ Spoofing	By changing the sever/client's certificate or key, an attacker may attempt to connect to an unauthenticated server
Cat. 1			Brainstorming	Tampering/ Information Disclosure/ Spoofing	Need to protect or verify the certificate

to steal the information of the channel.

modifying the face recognition data, an attacker may cause an error or abnormal operation in the face recognition result.

modifying the face recognition data, an attacker can

Modifying the face recognition data, an attacker may cause an error or abnormal operation in the face recognition result.

Security Requirements

We've derived the security requirements through the STRIDE methodology. And we found out some of security requirements are linked to system requirements, section 2 above.

5.2. STRIDE

~~Threats that could have been grayed out~~

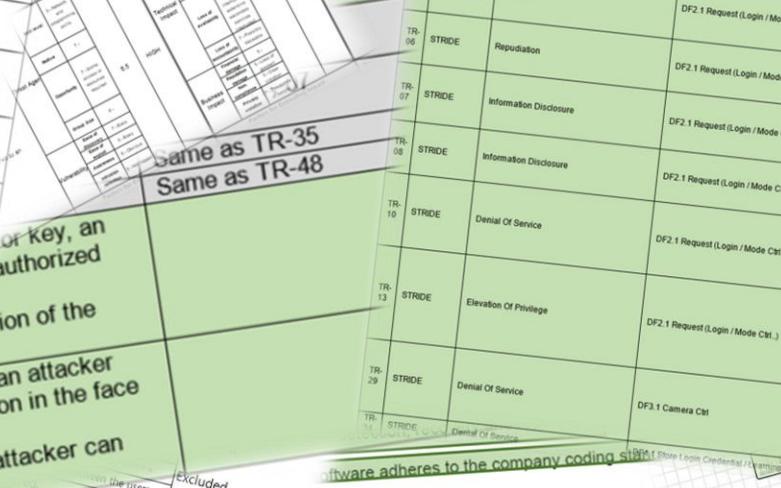
SR-ID	Security Requirement	Mapping
SR-01	A strong authentication method should be used.	CMU-REC
SR-02	Cryptographically strong password should be used.	CMU-REC
SR-03	Errors, exceptions, and abnormal conditions that may occur in the software must be handled robustly.	CMU-REC
SR-04	Input validation check is required in Client side.	
SR-05	Only the verified server and client should be connected and communicated.	
SR-06	Protect Camera from physical damage	
SR-07	Restrictions related to files are necessary to avoid system problems.	
SR-08	Save contents of the communication as a log and use as proof of non-repudiation.	
SR-09	Server and client must communicate using an encrypted channel.	CMU-REC
SR-10	The system must perform an integrity check before using user credentials.	
SR-11	The system shall know the change of the user credential data.	
SR-12	Use well-known cryptographic libraries and robust algorithms.	CMU-REC
SR-13	User Credential Data should be encrypted in the storage.	
SR-14	Video Stream over the connection should be protected.	
SR-15	A server and client program must perform an integrity check before using a certificate or key.	
R-16	Face recognition data should be encrypted in the storage.	
R-17	Every encryption time, newly generated random key is used for encryption to make reverse analysis difficult	
R-18	ROOT encrypt key must be protected from binary analysis	

3. Mitigation

We were trying to mitigate the threat
And we've derived the result below

We were trying to mitigate the threat and mentioned in the Security Requirements, And we've derived the result below.

Section R-17		Every encryption make reverse analysis difficult		ROOT encrypt key must be protected from binary analysis	
SR-18					
Excluded	Excluded	Factor	Factor	Spooling	DF4.2 Load Login Cred
Excluded	Excluded	Factor	Factor	TR-04 STRIDE Spoofing	DF2.1 Request (Login / Mode Ctrl)
Excluded	Excluded	Factor	Factor	TR-05 STRIDE Tampering	DF2.1 Request (Login / Mode Ctrl)
Excluded	Excluded	Factor	Factor	TR-06 STRIDE Repudiation	DF2.1 Request (Login / Mode Ctrl)
Excluded	Excluded	Factor	Factor	TR-07 STRIDE Information Disclosure	DF2.1 Request (Login / Mode Ctrl)
Excluded	Excluded	Factor	Factor	TR-08 STRIDE Information Disclosure	DF2.1 Request (Login / Mode Ctrl)
Excluded	Excluded	Factor	Factor	TR-10 STRIDE Denial Of Service	DF2.1 Request (Login / Mode Ctrl)
Excluded	Excluded	Factor	Factor	TR-13 STRIDE Elevation Of Privilege	DF2.1 Request (Login / Mode Ctrl)
Excluded	Excluded	Factor	Factor	TR-29 STRIDE Denial Of Service	DF3.1 Camera Ctrl
Excluded	Excluded	Factor	Factor	TR-31 STRIDE Denial Of Service	DF3.1 Camera Ctrl
Software adheres to the company coding standards					



with system requirement	Mitigation ID
Q-D-09	MI-10
	MI-01
Q-D-15	MI-04
	MI-05
	MI-11
	MI-08
	MI-12
	MI-09
EQ-D-02	MI-02
	MI-07
	MI-07
REQ-D-10	MI-03, MI-07
	MI-03
	MI-02
	MI-13
	MI-06
	MI-14
	MI-15

		Attacking / Exploiting		Mitigation / Countermeasures	
		Attacking / Exploiting		Mitigation / Countermeasures	
TRIDE	Spoofing		DF2.5 Result (Video Stream...)	Data can be exposed to others.	Need to protect credential
TRIDE	Spoofing		DF2.5 Result (Video Stream...)	Server (Jelson) may be spoofed by an attacker.	Need to apply authentication
TRIDE	Tampering		DF2.5 Result (Video Stream...)	Client (PC) may be spoofed by an attacker.	Need to apply authentication
TRIDE	Information Disclosure		DF2.5 Result (Video Stream...)	Video Stream may be tampered with by an attacker.	Need to probe video stream connection
TRIDE	Denial Of Service		DF2.5 Result (Video Stream...)	Video Stream may be sniffed with by an attacker.	Need to protect video stream connection
TRIDE	Elevation Of Privilege		DF2.5 Result (Video Stream...)	Client (PC) crashes, hangs, stops or runs slowly.	Need to remain in abnormal case
TRIDE	Elevation Of Privilege		DF2.5 Result (Video Stream...)	Server (Jelson) may be able to immediately execute code.	Need input sanitization
TRIDE	Tampering	User credential data		An attacker may pass data into 1 Client (PC).	Need input sanitization
TRIDE	Information Disclosure	Client <-> Server		Change the image data not to recognize registered users.	Need to protect user credential data
TRIDE	Information Disclosure	Server <-> Client		Disclose administrator's ID/Password to the employees in the company.	Need to more strengthen process for authentication
N/A		Network		Shift the communication channel between server and client to get user credential data.	Need to protect the data over the connection
TRIDE	Tampering/ Information Disclosure/ Spoofing	Server <-> Client		Compressing the connection of network physically as an attack.	Server need to be robust in abnormal case
TRIDE	Tampering/ Information Disclosure/ Spoofing			By changing the server/client's certificate or key, an attacker may attempt to connect to an unauthenticated server.	Need to protect or verify the certificate

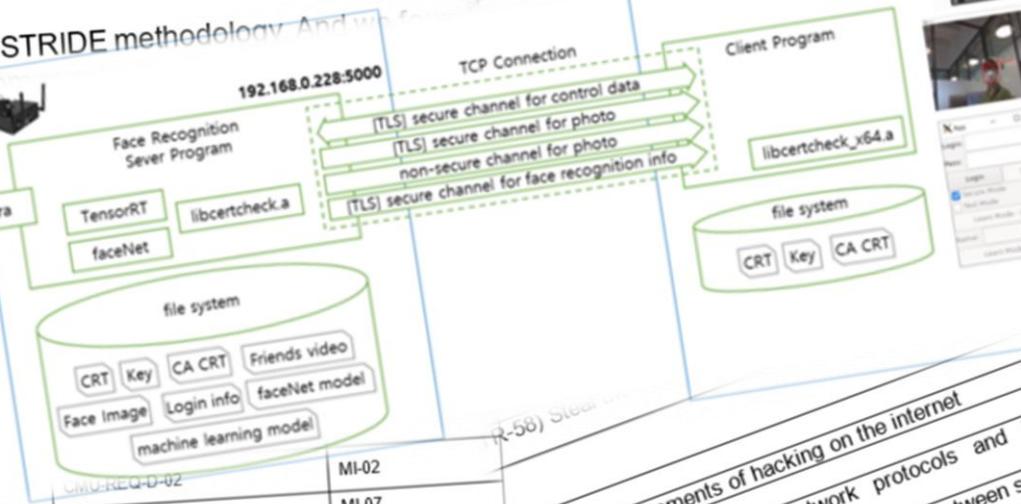
7. Security Requirements

We've derived the security requirements through the STRIDE methodology. And out some of security requirements are linked to system architecture.

5.2. STRIDE

Threats that could not arise as a threat were grayed out.

SR-ID	Security Requirement
SR-01	A strong authentication method should be used.
SR-02	Cryptographically strong password should be used.
SR-03	Errors, exceptions, and abnormal conditions that may occur must be handled robustly.
SR-04	Input validation check is required in Client side.
SR-05	Only the verified server and client should be connected and communicate.
SR-06	Protect Camera from physical damage
SR-07	Restrictions related to files are necessary to avoid system problems.
SR-08	Save contents of the communication as a log and use as proof for repudiation.
SR-09	Server and client must communicate using an encrypted channel.
SR-10	The system must perform an integrity check before using user credentials.
SR-11	The system shall know the change of the user credential data.
SR-12	Use well-known cryptographic libraries and robust algorithms.
SR-13	User Credential Data should be encrypted in the storage.
SR-14	Video Stream over the connection should be protected.
SR-15	A server and client program must perform an integrity check before using a certificate or key.
R-16	Face recognition data should be encrypted in the storage.
R-17	Every encryption time, newly generated random key is used for encryption to make reverse analysis difficult.
SR-18	ROOT encrypt key must be protected from binary analysis.



8. Mitigation

We were trying to mitigate the threat and mentioned in the Security Requirements, section 7. And we've derived the result below.

MHD	Mitigation
MI-01	Apply setting policy of cryptographically strong password - Enforce passwords longer than 7 characters - Forces the use of mixed letters of the alphabet and numbers.

MHD	Mitigation
MI-02	Communicate using Encrypted channel - using protocol TLS 1.2 or higher - Consider mutual authentication between server and client

MHD	Mitigation
MI-03	Encrypt user credential data in storage - Use OpenSSL library of latest version (1.1.1k) - Use an algorithm that are stronger than AES256 - Use CBC or GCM mode

MHD	Mitigation
MI-04	Implement robust system - Error handling - Exception handling - Finding countermeasures for predictable abnormal conditions

MHD	Mitigation
MI-05	Input validation check - Input sanitization

MHD	Mitigation
MI-06	Encrypt face recognition data in storage - Use OpenSSL library of latest version (1.1.1k) - Use an algorithm that are stronger than AES256 - Use CBC or GCM mode

MHD	Mitigation
MI-07	Integrity Check with hash function - Use OpenSSL library of latest version (1.1.1k) - Protect from physical damage - Wrap the camera module out of sight, or glue the cable to the camera.

MHD	Mitigation
MI-08	Save contents of communication as a log - Save log of the request and response between the server and the client

MHD	Mitigation
MI-09	Strong authentication method - Consider 2-Factor-Authentication method

MHD	Mitigation
MI-10	Use mutual authentication - Using protocol TLS 1.2 or higher - Use mutual authentication between server and client

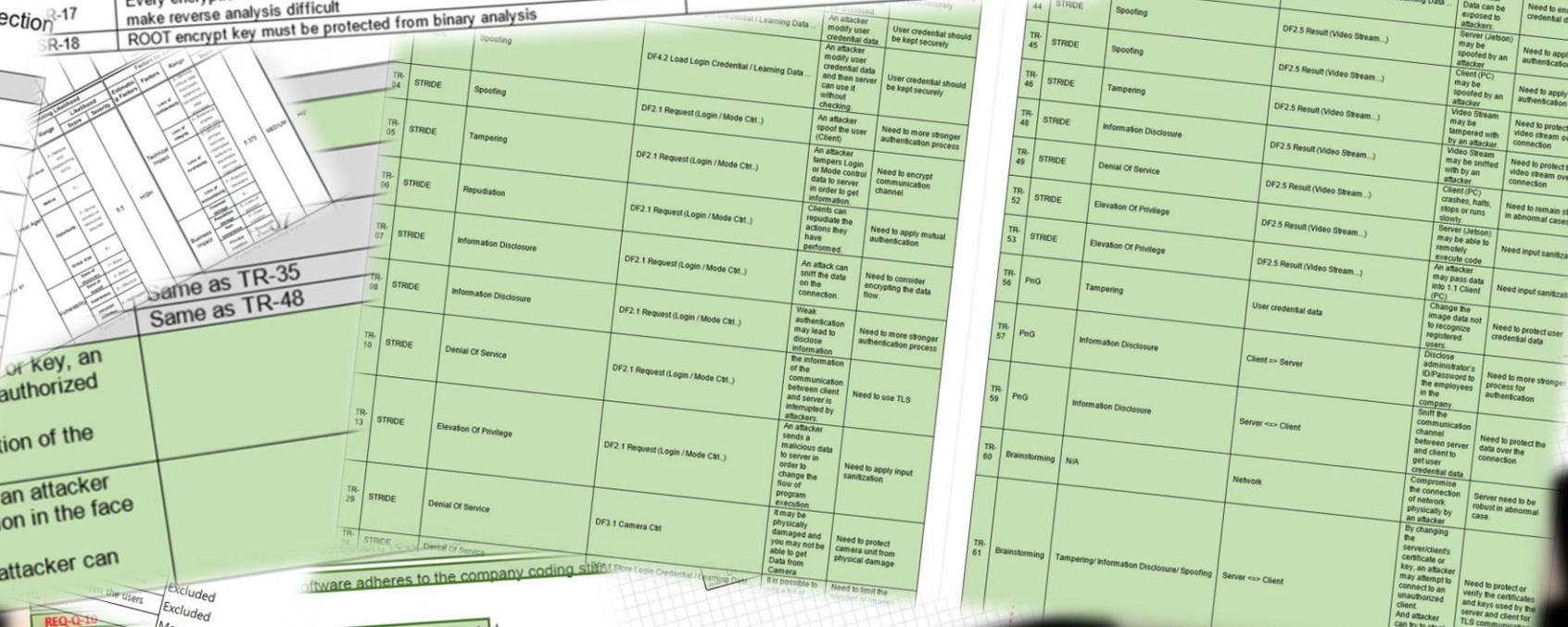
MHD	Mitigation
MI-11	Validation of image when file saving - File name verification(uniqueness) when image save : generate the name of file using random number.

MHD	Mitigation
MI-12	Certificate & Key file existence check - Use OpenSSL library of latest version (1.1.1k)

MHD	Mitigation
MI-13	File size validation when image save

MHD	Mitigation
MI-14	Integrity Check with hash function - Use OpenSSL library that are stronger than sha256

MHD	Mitigation
MI-15	Use an algorithm that are stronger than AES256



Hacker Post the achievements of hacking on the internet
Extensive knowledge of network protocols and program.

Sniff the communication channel between net user credential data.

additional consideration

Images in the storage
User credential may be disclosed.
User credential data can be exposed to an attacker.

Server (Jebon) may be spoofed by an attacker.
Client (PC) may be spoofed by an attacker.

Video Stream may be tampered with by an attacker.
Video Stream will be sniffed by an attacker.

Client (PC) crashes, halts, stops or runs slow.
Server (Jebon) may be able to resume to execute code.

An attacker may sniff the communication channel between server and client to get user credential data.

Change the image data not to recognize registered users.
Disclose administrator's ID/Password to the employee in the company.

Need to protect user credential data.

Need to protect process for authentication.

Need to protect the data over the connection.

Server need to be robust in abnormal case.

Comproose the connection of network physically by an attacker.

By changing the certificates or keys, an attacker may attempt to connect to an unauthorized client.

It can be done by changing the certificates or keys used by the server and client for TLS communication.

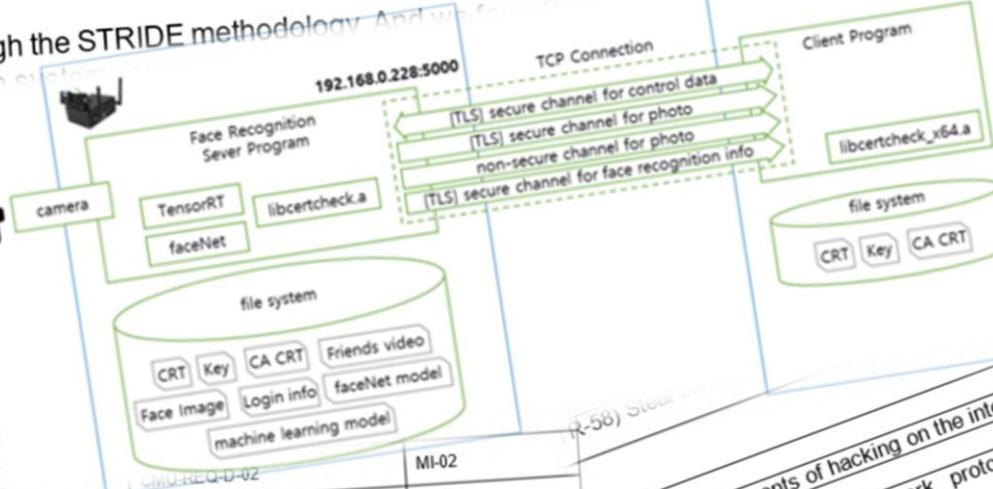
7. Security Requirements

We've derived the security requirements through the STRIDE methodology. And here are some of security requirements are linked to system.

5.2. STRIDE

Threats that could not arise as a threat were grayed out.

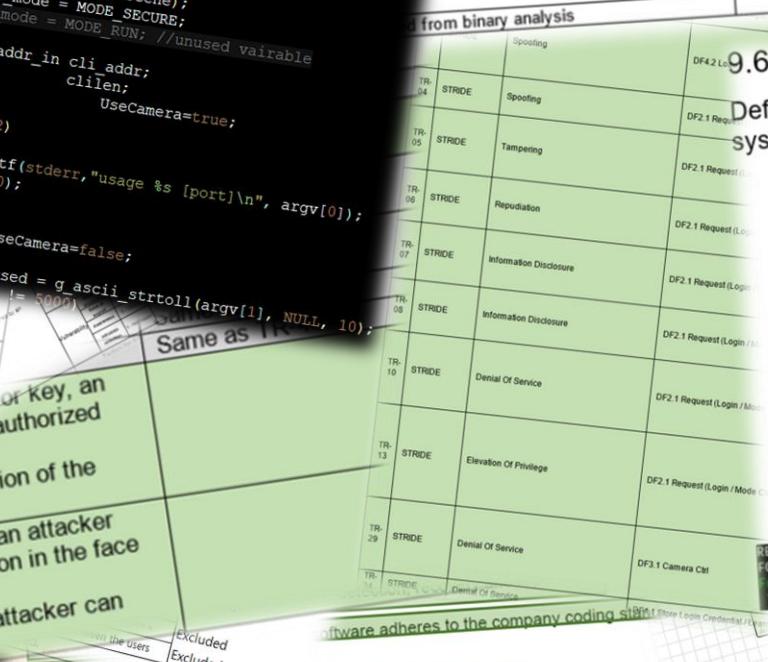
SR-ID	Security Requirement
SR-01	A strong authentication method should be used.
SR-02	Cryptographically strong password should be used.
SR-03	Errors, exceptions, and abnormal conditions that may occur must be handled robustly.
SR-04	Input validation check is required in Client side.
SR-05	Only the verified server and client should be connected and Protect Camera from physical damage



8. Mitigation

We were trying to mitigate 7. And we've derived

MHD	Mitigation
MI-01	Apply setting policy - Enforce password - Forces the use of TOTP - Consider mutual TLS - Encrypt user credentials - Use OpenSSL library - Use an algorithm - Implement robust error handling - Exception handling - Finding countermeasures for predictable abnormal conditions - Input sanitization - Encrypt face recognition data in storage - Use OpenSSL library of latest version (1.1.1k) - Use an algorithm that are stronger than AES256 - Use CBC or GCM mode - Integrity Check with hash function - Use an algorithm that are stronger than sha256 - Protect from physical damage - Wrap the camera module out of sight or glue the cable to the camera. - Save contents of communication as a log - Save log of the request and response between the server and the client - Strong authentication method - Consider 2-Factor-Authentication method - Use mutual authentication - Using protocol TLS 1.2 or higher - Use mutual authentication between server and client - Validation of image when file saving - File name verification(uniqueness) when image save : generate the name of file using random number - File size validation when image save - Certificate & Key file existence check - Integrity Check with hash function - Use OpenSSL library of latest version (1.1.1k) - Use an algorithm that are stronger than sha256
MI-02	Communicate using protocol TLS - Consider mutual TLS - Encrypt user credentials - Use OpenSSL library - Use an algorithm - Implement robust error handling - Exception handling - Finding countermeasures for predictable abnormal conditions - Input sanitization - Encrypt face recognition data in storage - Use OpenSSL library of latest version (1.1.1k) - Use an algorithm that are stronger than AES256 - Use CBC or GCM mode - Integrity Check with hash function - Use an algorithm that are stronger than sha256 - Protect from physical damage - Wrap the camera module out of sight or glue the cable to the camera. - Save contents of communication as a log - Save log of the request and response between the server and the client - Strong authentication method - Consider 2-Factor-Authentication method - Use mutual authentication - Using protocol TLS 1.2 or higher - Use mutual authentication between server and client - Validation of image when file saving - File name verification(uniqueness) when image save : generate the name of file using random number - File size validation when image save - Certificate & Key file existence check - Integrity Check with hash function - Use OpenSSL library of latest version (1.1.1k) - Use an algorithm that are stronger than sha256
MI-03	Use mutual authentication - Consider mutual TLS - Encrypt user credentials - Use OpenSSL library - Use an algorithm - Implement robust error handling - Exception handling - Finding countermeasures for predictable abnormal conditions - Input sanitization - Encrypt face recognition data in storage - Use OpenSSL library of latest version (1.1.1k) - Use an algorithm that are stronger than AES256 - Use CBC or GCM mode - Integrity Check with hash function - Use an algorithm that are stronger than sha256 - Protect from physical damage - Wrap the camera module out of sight or glue the cable to the camera. - Save contents of communication as a log - Save log of the request and response between the server and the client - Strong authentication method - Consider 2-Factor-Authentication method - Use mutual authentication - Using protocol TLS 1.2 or higher - Use mutual authentication between server and client - Validation of image when file saving - File name verification(uniqueness) when image save : generate the name of file using random number - File size validation when image save - Certificate & Key file existence check - Integrity Check with hash function - Use OpenSSL library of latest version (1.1.1k) - Use an algorithm that are stronger than sha256
MI-04	Use mutual authentication - Consider mutual TLS - Encrypt user credentials - Use OpenSSL library - Use an algorithm - Implement robust error handling - Exception handling - Finding countermeasures for predictable abnormal conditions - Input sanitization - Encrypt face recognition data in storage - Use OpenSSL library of latest version (1.1.1k) - Use an algorithm that are stronger than AES256 - Use CBC or GCM mode - Integrity Check with hash function - Use an algorithm that are stronger than sha256 - Protect from physical damage - Wrap the camera module out of sight or glue the cable to the camera. - Save contents of communication as a log - Save log of the request and response between the server and the client - Strong authentication method - Consider 2-Factor-Authentication method - Use mutual authentication - Using protocol TLS 1.2 or higher - Use mutual authentication between server and client - Validation of image when file saving - File name verification(uniqueness) when image save : generate the name of file using random number - File size validation when image save - Certificate & Key file existence check - Integrity Check with hash function - Use OpenSSL library of latest version (1.1.1k) - Use an algorithm that are stronger than sha256
MI-05	Use mutual authentication - Consider mutual TLS - Encrypt user credentials - Use OpenSSL library - Use an algorithm - Implement robust error handling - Exception handling - Finding countermeasures for predictable abnormal conditions - Input sanitization - Encrypt face recognition data in storage - Use OpenSSL library of latest version (1.1.1k) - Use an algorithm that are stronger than AES256 - Use CBC or GCM mode - Integrity Check with hash function - Use an algorithm that are stronger than sha256 - Protect from physical damage - Wrap the camera module out of sight or glue the cable to the camera. - Save contents of communication as a log - Save log of the request and response between the server and the client - Strong authentication method - Consider 2-Factor-Authentication method - Use mutual authentication - Using protocol TLS 1.2 or higher - Use mutual authentication between server and client - Validation of image when file saving - File name verification(uniqueness) when image save : generate the name of file using random number - File size validation when image save - Certificate & Key file existence check - Integrity Check with hash function - Use OpenSSL library of latest version (1.1.1k) - Use an algorithm that are stronger than sha256



9.6. Compile Options

Defenses at the compiler, check the mitigation technologies in use by processes on a Linux system.

1. checksec.sh (<https://www.trapkit.de/tools/checksec/>)

- Modern Linux distributions offer some mitigation techniques to make it harder to exploit software vulnerabilities reliably. Mitigations such as RELRO, NoExecute (NX), Stack Canaries, Address Space Layout Randomization (ASLR) and Position Independent Executables (PIE) have made reliably exploiting any vulnerabilities that do exist far more challenging. The checksec.sh script is designed to test what standard Linux OS and PaX security features are being used.

B. Result of running checksec.sh (before)

- Symbols is not striped
- RW-RUNPATH

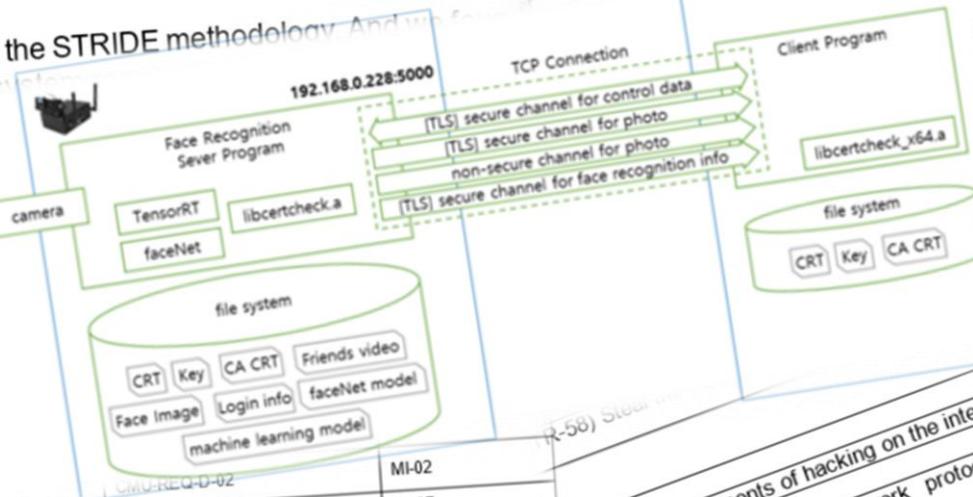
RELRO	STACK CANARY	NX	PIE	RPATH	RUNPATH	Symbols
Fortify Fortified Full RELRO Yes	Fortifiable Canary found 34	NX enabled PIE enabled LgFaceRecDemoTCP_Jetson_NanoV2	No RPATH RW-RUNPATH	5215 Symbols		

C. Result of running checksec.sh (after apply options for defenses)

- Add Symbol stripped option

7. Security Requirements

The diagram illustrates the STRIDE methodology for deriving security requirements. It shows a central box labeled "We've derived the security requirements through the STRIDE methodology" with arrows pointing to five categories: "Face Recognition", "TCP Connection", "[TLS] secure channel for control data", "[TLS] secure channel for photo", and "Client Program". Each category is associated with a specific IP address and port: "Face Recognition" is at 192.168.0.228:5000, "TCP Connection" is at 192.168.0.228:5001, "[TLS] secure channel for control data" is at 192.168.0.228:5002, "[TLS] secure channel for photo" is at 192.168.0.228:5003, and "Client Program" is at 192.168.0.228:5004.



3. Compile Options

For more information about the security features available in the compiler, refer to the [Intel® C/C++ Compiler Security Features](#).

- A. Modern Linux distributions offer some mitigation techniques to help exploit software vulnerabilities reliably. Mitigations such as RELRO, NoExec (NX), Stack Canaries, Address Space Layout Randomization (ASLR) and Position Independent Executables (PIE) have made reliably exploiting any vulnerabilities that do exist far more challenging. The checksec.sh script is designed to test what standard Linux OS and PaX security features are being used.

B. Result of running checksec.sh (before)

- i. Symbols is not striped
 - ii. RW-RUNPATH

ii.	RW-RUNPATH	ELRO	STACK CANARY	NX	PIE	RPATH	RUNPATH	Symbols
ORTIFY Fortified	Fortifiable FILE							
All RELRO	Canary found		NX enabled	PIE enabled	No RPATH	RW-RUNPATH	3215 Symbols	
	34		LgFaceRecDemoTCP_Jetson_NanoV2					

Q. Result of running checksec.sh (after apply options for defenses)

- ### C. Result of running C++

7. Security Requirements

The diagram shows a 'Client Program' on the left connected via a dashed green line to a 'Face Recognition' service on the right. The connection is labeled 'TCP Connection'. Two parallel arrows indicate data exchange: one arrow points from the Client Program to the Face Recognition service, labeled '[TLS] secure channel for control data'; the other arrow points from the Face Recognition service to the Client Program, labeled '[TLS] secure channel for photo'. Above the connection, the IP address '192.168.0.228:5000' is displayed.



With respect to quality attributes in order to apply objective standards of measurement of system and software product quality can be mentioned the measures of SW attributes

Topics	Characteristics	Description
Confidentiality	Confidential to a particular person	

Result of running checksec.sh (after apply options for defenses)

dd Symbol stripped option

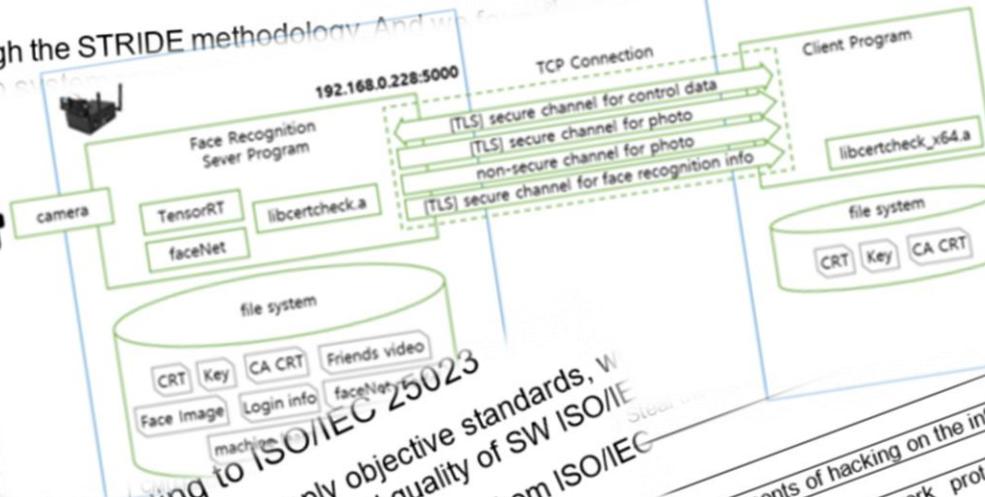
7. Security Requirements

We've derived the security requirements through the STRIDE methodology, but some of security requirements are linked to system architecture.

5.2. STRIDE

~~Threats that could not be identified were grayed out.~~

SR-ID	Security Requirement
SR-01	A strong authentication method should be used.
SR-02	Cryptographically strong password should be used.
SR-03	Errors, exceptions, and abnormal conditions that may must be handled robustly.
SR-04	Input validation check is required in Client side.
SR-05	Only the verified server and client should be connected.
	Protect Camera from physical damage



3. Mitigation

	Validation	1
pass	validation	
login		1
b9af e256f a35c b9af 0142	(Byungchu1 (Byungchu (Kyungnam3 (Kyungnam (Byungchu1	

ID	Mitigation	TC Name	Step	Expected	Execution Result
3.	Mitigation	3. Mitigation	id validation	type id more than 10 len	cannot type character more than 10
	We were trying to mitigate this issue		pass validation	type pass more than 20 len	cannot type character more than 20
	And we've derived		login	<ul style="list-style-type: none"> type id something make id to empty string type pass something type id,pass something disconnect client and server in the local network push login button connect client and server in the local network Do not meet the condition below <ul style="list-style-type: none"> - type alphabet and number in id - Minimum eight characters, at least one letter, one number and one special character on password 	<ul style="list-style-type: none"> check login button is not activated check login button is activated check alert 'Connection Fail' OK check login button is activated
	Input validation check		9	push login button and show admin user face on camera	check alert 'Show your face on camera' after 5 sec, check alert 'Connection Fail'
	Input sanitization		10	type valid id, pass	OK
	Encrypt face recognition data in storage		11	push login button and show admin user face on camera within 5sec	check id, pass, login button component are deactivated secure mode check button activated and checked check running secure run mode (camera is on and I can see the camera)
	- Use OpenSSL library of latest version (1.1.1)		logout	pre	login is needed
	- Use an algorithm that are stronger than AES				an alert
	Integrity Check with hash function				OK
	- Use OpenSSL library of latest version (1.1.1)				Excluded
	- Use an algorithm that are stronger than sha256				Excluded
	Protect from physical damage				Software adhe
	Save contents of communication as a log				
	- Save log of the request and response between client and server				
	Strong authentication method				
	Consider 2-Factor-Authentication method				
	Use mutual authentication				
	- Using protocol TLS1.2 or higher				
	Use mutual authentication between server and client				
	Validation of image when file saving				
	- File name verification(uniqueness) when image save				
	File size validation when image save				
	File & Key file existence check				
	Security Check with hash function				
	- Use OpenSSL library of latest version (1.1.1)				
	An algorithm that are stronger than sha256				

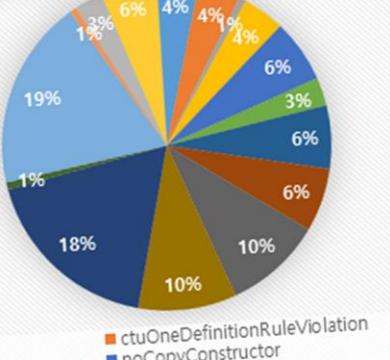
10. Quality Attributes according to ISO/IEC 2502

Here is the tabular solution for the structural static analysis problem.

In this static analysis

We're actually thinking of how to check vulnerabilities of the code and we wanted to detect them using any kind of static tools. Firstly, we used two tools in syllabus— Flawfinder. The reason why is this tool is introduced in the syllabus and it's appropriate for the time pressure so that we can adapt it.

Tools	Support C/C++	Free software	Latest release	Comment
Flawfinder	O	O	O (2021-06-03)	Detecting BOF and reporting HTML and csv format for reviewer
RATS	O	O	X (2014-01-01)	Detecting BOF, TOCTOU, Race condition
spotBugs	X (Java)	O	O (2021-04-16)	Like as findbug, Java code
sonarQube	O	X	O (2021-05-04)	
MD	X (Java, JS, ...)	O	O (2021-05-29)	Java code
ocwork	O	X	O (2021-01-01)	
ocheck	O	O	O (2021-03-23)	Detecting BOF, exception handling, memory leak, unused variable, functions undefined
erity	O	X	O	



7. Security Requirements

We've derived the security requirements through the out some of security requirements are linked to s

SR-ID	Security Requirement
SR-01	A strong authentication method should be used.
SR-02	Cryptographically strong password should be used.
SR-03	Errors, exceptions, and abnormal conditions that may occur must be handled robustly.
SR-04	Input validation check is required in Client side.
SR-05	Only the verified server and client should be connected and Protect Camera from physical damage



8. Mitigation

	We were trying to mitigate 7. And we've derived
MI-ID	Mitigation
MI-01	<ul style="list-style-type: none"> Apply setting policy - Enforce password complexity - Forces the use of two-factor authentication
MI-02	<ul style="list-style-type: none"> Communicate using TLS 1.3 or higher - using protocol TLS 1.3 or higher - Consider mutual authentication
MI-03	<ul style="list-style-type: none"> Encrypt user credentials - Use OpenSSL library - Use an algorithm that is stronger than CBC or GCM mode
MI-04	<ul style="list-style-type: none"> Implement robust exception handling - Error handling - Exception handling - Finding countermeasures for potential security issues - Input validation check - Input sanitization
MI-05	<ul style="list-style-type: none"> Encrypt face recognition data in storage - Use OpenSSL library of latest version - Use an algorithm that are stronger than CBC or GCM mode
MI-06	<ul style="list-style-type: none"> Integrity Check with hash function - Use OpenSSL library of latest version - Use an algorithm that are stronger than SHA-256
MI-07	<ul style="list-style-type: none"> Protect from physical damage - Wrap the camera module out of sight
MI-08	<ul style="list-style-type: none"> Save contents of communication as a log file - Save log of the request and responses
MI-09	<ul style="list-style-type: none"> Strong authentication method - Consider 2-Factor-Authentication method
MI-10	<ul style="list-style-type: none"> Use mutual authentication - Using protocol TLS 1.2 or higher - Use mutual authentication between servers
MI-11	<ul style="list-style-type: none"> Validation of image when file saving - File name verification(uniqueness) when saving - File size validation when image save
MI-12	<ul style="list-style-type: none"> Certificate & Key file existence check - Use OpenSSL library of latest version - Use an algorithm that are stronger than SHA-256
MI-13	<ul style="list-style-type: none"> Use OpenSSl library of latest version (1.1.1) - Use an algorithm that are stronger than SHA-256

TC Name	Step	Expected	Execution Result
1 id validation	1 type id more than 10 len	cannot type character more than 10	OK
2 pass validation	1 type pass more than 20 len	cannot type character more than 20	OK
3 login	1 type id something	check login button is not activated	OK
	2 make id to empty string	check login button is not activated	OK
	3 type pass something	check login button is not activated	OK
	4 type id,pass something	check login button is not activated	OK
	5 disconnect client and server in the local network	check login button is activated	OK
	6 push login button	check alert 'Connection Fail'	OK
	7 connect client and server in the local network	check alert 'Connection Fail'	OK
8	Do not meet the condition below - type alphabet and number in id - Minimum eight characters, at least one letter, one number and one special character on password	check login button is activated	OK
9	push login button and show admin user face on camera	check alert 'Show your face on camera' after 5 sec, check alert 'Connection Fail'	OK
10	type valid id, pass	check alert 'Show your face on camera' after 5 sec, check alert 'Connection Fail'	OK
11	push login button and show admin user face on camera within 5sec	check id, pass, login button component are deactivated secure mode check button activated and checked check running secure run mode (camera is on and I can see the camera)	OK
out	pre	an alert login is needed	OK

9.10. Quality Attributes according to ISO/IEC 2502

With respect to quality attributes in order to apply objective adapt measurement of system and software product quality

ID	Measure Name
SCo-2-G	Data encryption
Sco-3-5	Strength
Dat	

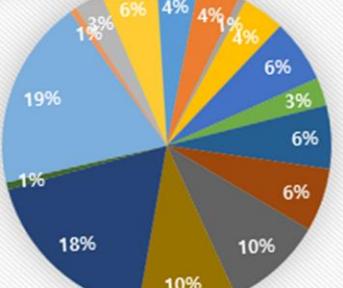
Here is the traditional star 10. Static Analysis

We're actually thinking of how to check vulnerabilities of the code and we wanted to determine them using any kind of static tools. Firstly, we used two tools in syllabus— Flawfinder. The reason why is this tool is introduced in the syllabus and it's appropriate considering the time pressure so that we can adapt it.

Tools	Support C/C++	Free software	Latest release	Comment
Flawfinder	O	O	O (2021-06-03)	Detecting BOF and reporting HTML and csv format for reviewer
RATS	O	O	X (2014-01-01)	Detecting BOF, TOCTOU, Race condition
SpotBugs	X (Java)	O	O (2021-04-16)	
SonarQube	O	X	O (2021-05-04)	Like as findbug, Java code
PMD	X (Java, JS, ...)	O	O (2021-05-29)	
Jlocwork	O	X	O (2021-01)	Java code
Cppcheck	O	O	차트 영역 (2021-03-23)	
Valgrind	O	X	O	Detecting BOF, exception handling, memory leak, unused variables and functions, uninitialized variables
LeakSanitizer	O	X	O	Need build environment

7. Security Requirements

We've derived the security requirements through the STRIDE methodology. And out some of security requirements are linked to system.



- ctuOneDefinitionRuleViolation
- noCopyConstructor
- passedByValue
- unreadVariable
- unusedLabel

- invalidPointerCast
- noExplicitConstructor
- postfixOperator
- unsignedLessThanZero
- useInitializationList

SR-ID | Security Requirement

- SR-01 A strong authentication method should be used.
- SR-02 Cryptographically strong password should be used.
- SR-03 Errors, exceptions, and abnormal conditions that may occur must be handled robustly.
- SR-04 Input validation check is required in Client side.

- SR-05 Only the verified server and client should be connected and Protect Camera from physical damage

LOG_INFO("Authentication failed\n");

return ret;

gint main(gint argc, gchar *argv[])

gint maxFacesPerScene = MAXFACES;

TCpListenPort *TcpListenPort;

TCpConnectPort *TCpConnectPort;

port control; port_sdata; port_nsdata; port_meta; asset/friend;

an encrypted channel before using user credentials

user credential data

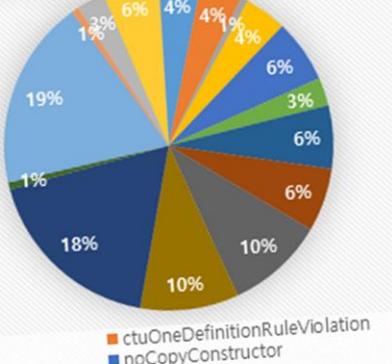
and robust algorithm in the code

in the code

constParameter missingOverride noOperatorEq uninitMemberVar

internal data corruption Examples of internal corruption prevention frequently, compare data in periodically, store data in memory usage (CERTAINLY)

attribute usage (CERTAINLY)



7. Security Requirements

We've derived the security requirements through the STRIDE methodology. All out some of security requirements are linked to each other.

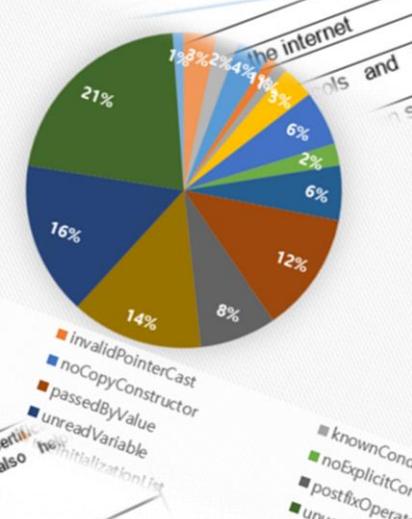
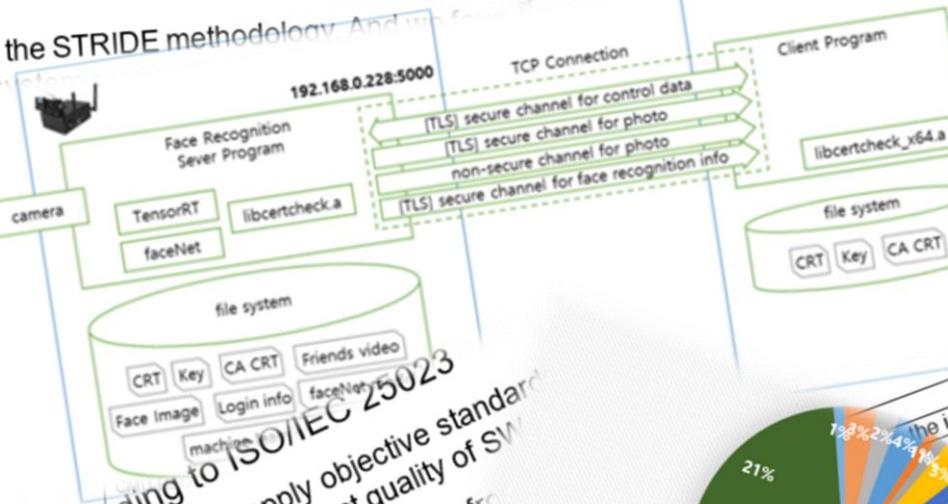
The diagram illustrates the STRIDE methodology applied to two components:

- Face Recognition**: Associated with IP address 192.168.0.228:5000.
- TCP Connection**: Associated with IP address 192.168.0.228:5000.

Security requirements are shown as arrows pointing from one component to another:

- A requirement labeled "[TLS] secure channel for control data" connects the Face Recognition component to the TCP Connection component.
- A requirement labeled "[TLS] secure channel for photo" connects the Face Recognition component to the TCP Connection component.
- A requirement labeled "Client Program" connects the TCP Connection component to the Client Program (represented by a person icon).

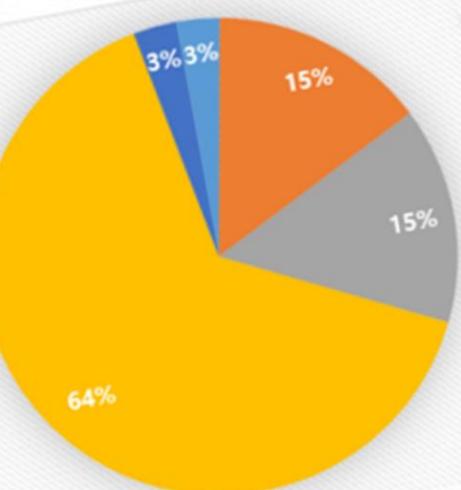
SR-ID	Security Requirement
SR-01	A strong authentication method should be used.
SR-02	Cryptographically strong password should be used.
SR-03	Errors, exceptions, and abnormal conditions that may occur must be handled robustly.
SR-04	Input validation check is required in Client side.



8. Mitigation

		validation	1
7.	We were trying to make validation pass	validation	1
	7. And we've derived	login	1
MUD	142	(Byungchul)	

TC Name		Step	Expect
M-I-01	id validation pass validation login	1 type id more than 10 len 1 type pass more than 20 len 1 type id something 2 make id to empty string 3 type pass something 4 type id,pass something 5 disconnect client and server in the local network 6 push login button 7 connect client and server in the local network 8 Do not meet the condition below - type alphabet and number in id - Minimum eight characters, at least one letter, one number and one special character on password 9 push login button and show admin user face on camera	cannot type char than 10 cannot type char than 20 check activated check login activated check login activated check login activated check alert 'Connection Fail' check login activated
	Mitigation	1 Apply setting policy - Enforce password rule - Forces the use of two-factor authentication - Communicate using https - using protocol TLS 1.3 - Consider mutual authentication - Encrypt user credentials - Use OpenSSL library - Use an algorithm that is stronger than AES - Use CBC or GCM mode	check activated
		2 Implement robust error handling - Error handling - Exception handling - Finding countermeasures for predictable attack - Input validation check - Input sanitization - Encrypt face recognition data in storage - Use OpenSSL library of latest version (1.1.1) - Use an algorithm that are stronger than SHA-256 - Use CBC or GCM mode	
	Mitigation	3 Integrity Check with hash function - Use OpenSSL library of latest version (1.1.1) - Use an algorithm that are stronger than SHA-256 4 Protect from physical damage - Wrap the camera module out of sight, or glue it to the board Save contents of communication as a log - Save log of the request and response between client and server Strong authentication method - Consider 2-Factor-Authentication method	check alert 'Show you on camera' after 5 sec, check 'Connection Fail'
		5 Use mutual authentication - Using protocol TLS 1.3 or higher - Use mutual authentication between server and client Validation of Image when file saving - File name verification(uniqueness) when image save - File size validation when image save Certificate & Key file existence check Integrity Check with hash function Use OpenSSL library of latest version (1.1.1)	check id, pass, login button component are deactivated secure mode check button activated and checked check running secure mode (camera is on and I see the camera)
	Mitigation	6 logout	generate the name of file using random number.
		pre	an all



Attributes according to ISO/IEC 25023

ID	Measure Name	Description
SCo-2-G	constParameter	const parameter
Sco-3-5	missingOverride	missing override
	noOperatorEq	no operator eq
	uninitMemberVar	uninit member var
SIn-1-G	Data integrity	internal data corruption prevention
SIn-2-G	Internal data corruption prevention	internal data corruption prevention
	Examples of internal data corruption prevention	examples of internal data corruption prevention
	frequently, compare data periodically, store data in nature usage (Certified algorithms are also helpful)	frequently, compare data periodically, store data in nature usage (Certified algorithms are also helpful)
	(Initialization list)	(Initialization list)

10d). Static Analysis

this static analysis, it is very helpful for us to check the initial vulnerabilities of our code. We're actually thinking of how to check vulnerabilities of the code and we wanted to detect design errors using any kind of static tools. Firstly, we used two tools in syllabus— Flawfinder. Then we can consider why is this tool is introduced in the syllabus and it's appropriate considering the pressure so that we can adapt it.

Support C/C++	Free software	Latest release	Comment
O	O	O (2021-06-03)	Detecting BOF and reporting HTML and csv format for reviewer
O	O	X (2014-01-01)	Detecting BOF, TOCTOU, Race condition
X (Java)	O	O (2021-04-16)	Like as findbug, Java code
O	X	O (2021-05-04)	
X (Java, JS, ...)	O	O (2021-05-29)	Java code
O	X	O (2021-01)	
O	O	차트 영역 (2021-03-23)	Detecting BOF, exception handling, memory leak, unused variables and functions, uninitialized variables
O	X	O	Need built-in tools

Define security goal

Define assets to protect

Do threat modeling

Do risk assessment to prioritize items

Define security requirements

Derive mitigations

Construct architecture

Implement the mitigations

Verify the mitigations

Define security goal

“ Protect the **user privacy**
information in our system.”

Define assets to protect + Do threat modeling



+ name

Define assets to protect + Do threat modeling



#privacy #stalker
#name



+ name

Define assets to protect + Do threat modeling



#privacy #stalker
#name

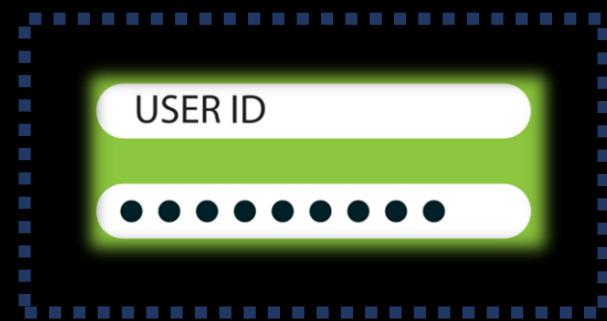


+ name



#authentication #2FA
#face #recognition

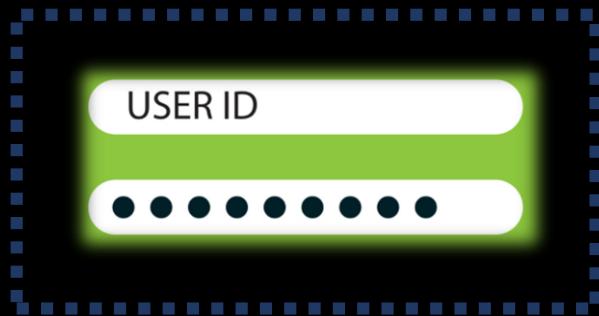
Define assets to protect + Do threat modeling



Define assets to protect + Do threat modeling



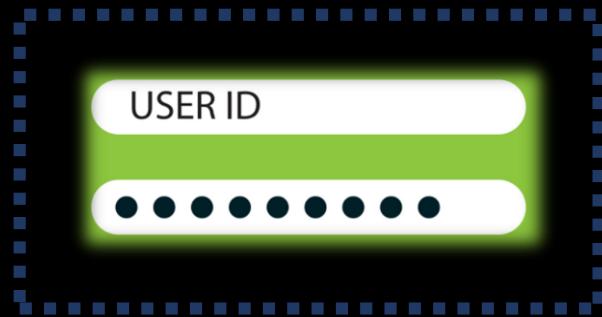
#authentication
#login #password



Define assets to protect + Do threat modeling



#authentication
#login #password



#privacy

Define assets to protect + Do threat modeling



Define assets to protect + Do threat modeling



#privacy



#authentication

Define assets to protect + Do threat modeling



Derive mitigations



+ name

- Introduce login system
- Force users to use strong password
- Apply 2FA (what you know + what you are)
- Separate meta data from image on network
- Encrypt the image + name in the storage
- Use verified crypto algorithm (AES256-CBC)
- Use a random file name in the storage
- Use TLS in network communication

Derive mitigations



+ name

Introduce login system

Force users to use strong password

Apply 2FA (what you know + what you are)

Separate meta data from image on network

Encrypt the image + name in the storage

Use verified crypto algorithm (AES256-CBC)

Use a random file name in the storage

Use TLS in network communication

Derive mitigations



+ name

Introduce login system

Force users to use strong password

Apply 2FA (what you know + what you are)

Separate meta data from image on network

Encrypt the image + name in the storage

Use verified crypto algorithm (AES256-CBC)

Use a random file name in the storage

Use TLS in network communication

Derive mitigations



+ name

- Introduce login system
- Force users to use strong password
- Apply 2FA (what you know + what you are)**
- Separate meta data from image on network
- Encrypt the image + name in the storage
- Use verified crypto algorithm (AES256-CBC)
- Use a random file name in the storage
- Use TLS in network communication

Derive mitigations



+ name

- Introduce login system
- Force users to use strong password
- Apply 2FA (what you know + what you are)
- Separate meta data from image on network**
- Encrypt the image + name in the storage
- Use verified crypto algorithm (AES256-CBC)
- Use a random file name in the storage
- Use TLS in network communication

Derive mitigations



+ name

- Introduce login system
- Force users to use strong password
- Apply 2FA (what you know + what you are)
- Separate meta data from image on network
- Encrypt the image + name in the storage**
- Use verified crypto algorithm (AES256-CBC)
- Use a random file name in the storage
- Use TLS in network communication

Derive mitigations



+ name

- Introduce login system
- Force users to use strong password
- Apply 2FA (what you know + what you are)
- Separate meta data from image on network
- Encrypt the image + name in the storage
- Use verified crypto algorithm (AES256-CBC)
- Use a random file name in the storage
- Use TLS in network communication

Derive mitigations



+ name

- Introduce login system
- Force users to use strong password
- Apply 2FA (what you know + what you are)
- Separate meta data from image on network
- Encrypt the image + name in the storage
- Use verified crypto algorithm (AES256-CBC)
- Use a random file name in the storage
- Use TLS in network communication

Derive mitigations



+ name

- Introduce login system
- Force users to use strong password
- Apply 2FA (what you know + what you are)
- Separate meta data from image on network
- Encrypt the image + name in the storage
- Use verified crypto algorithm (AES256-CBC)
- Use a random file name in the storage
- Use TLS in network communication

Derive mitigations



Encrypt the hashed credential in the storage
Use TLS in network communication

Derive mitigations



Encrypt the hashed credential in the storage

Use TLS in network communication

Derive mitigations



Encrypt the hashed credential in the storage
Use TLS in network communication

Derive mitigations



Embed the root key into code obfuscated
Use TLS in network communication

Derive mitigations



Embed the root key into code obfuscated

Use TLS in network communication

Derive mitigations



Embed the root key into code obfuscated
Use TLS in network communication

Derive mitigations

Introduce login system

Force users to use strong password

Encrypt the hashed credential in the storage

Apply 2FA (what you know + what you are)

Separate meta data from image on network

Encrypt the image + name in the storage

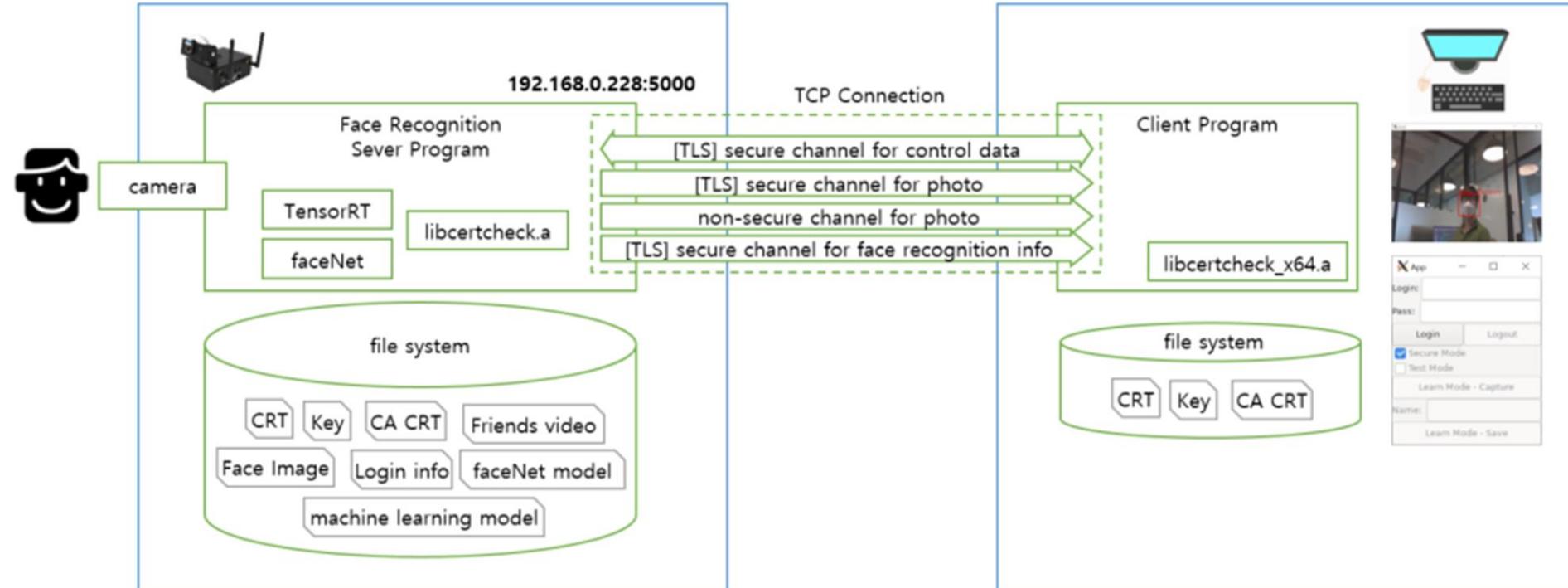
Use verified crypto algorithm (AES256-CBC)

Embed the root key into code obfuscated

Use a random file name in the storage

Use TLS in network communication

Construct architecture

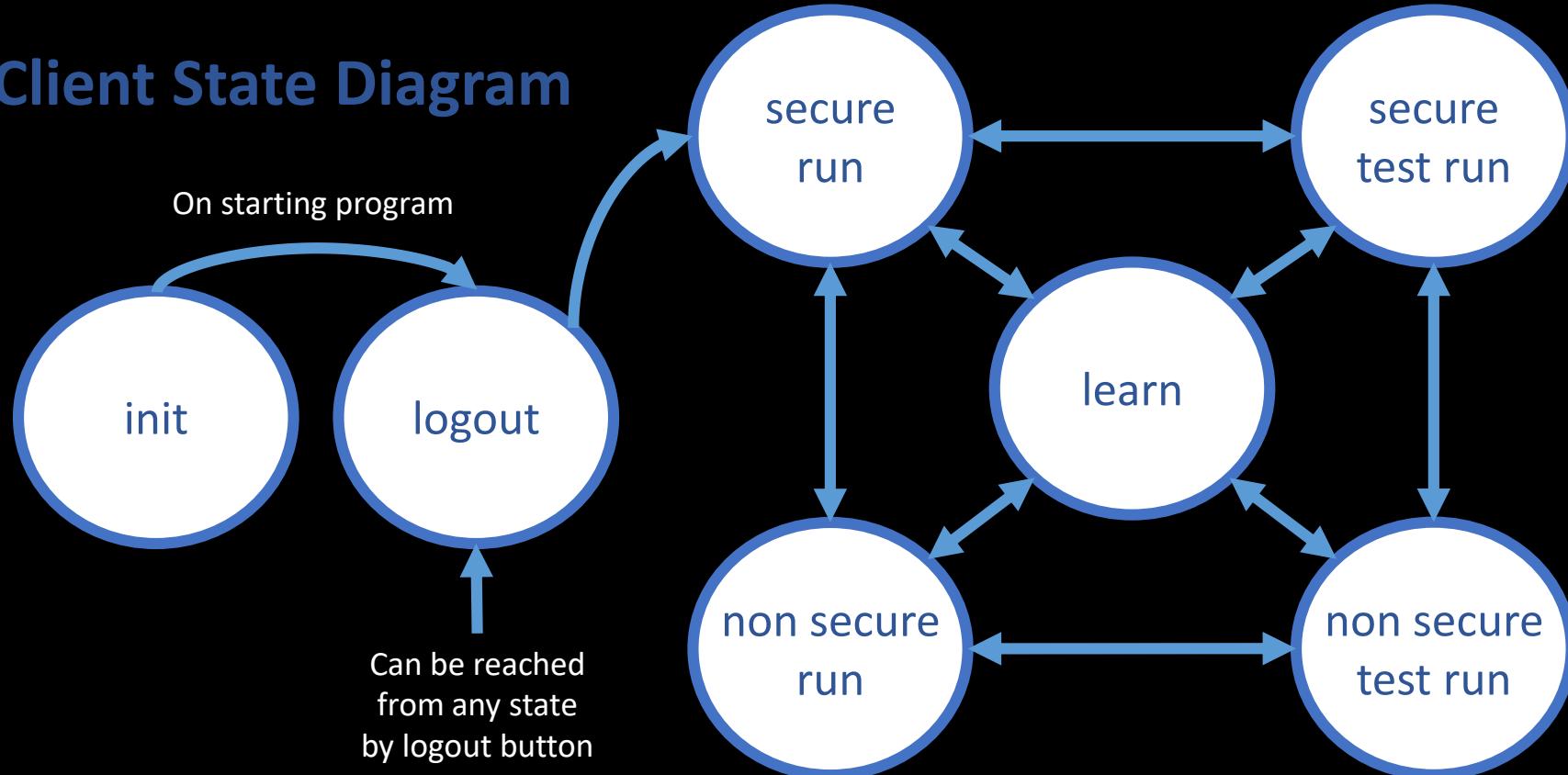


Construct architecture

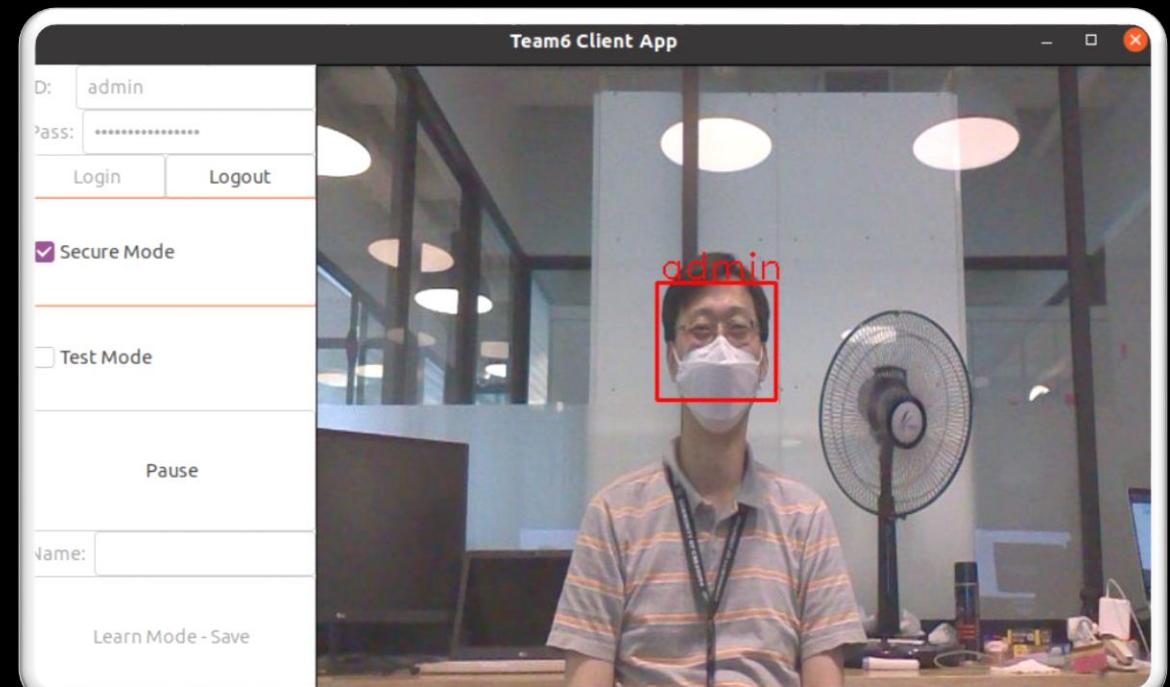
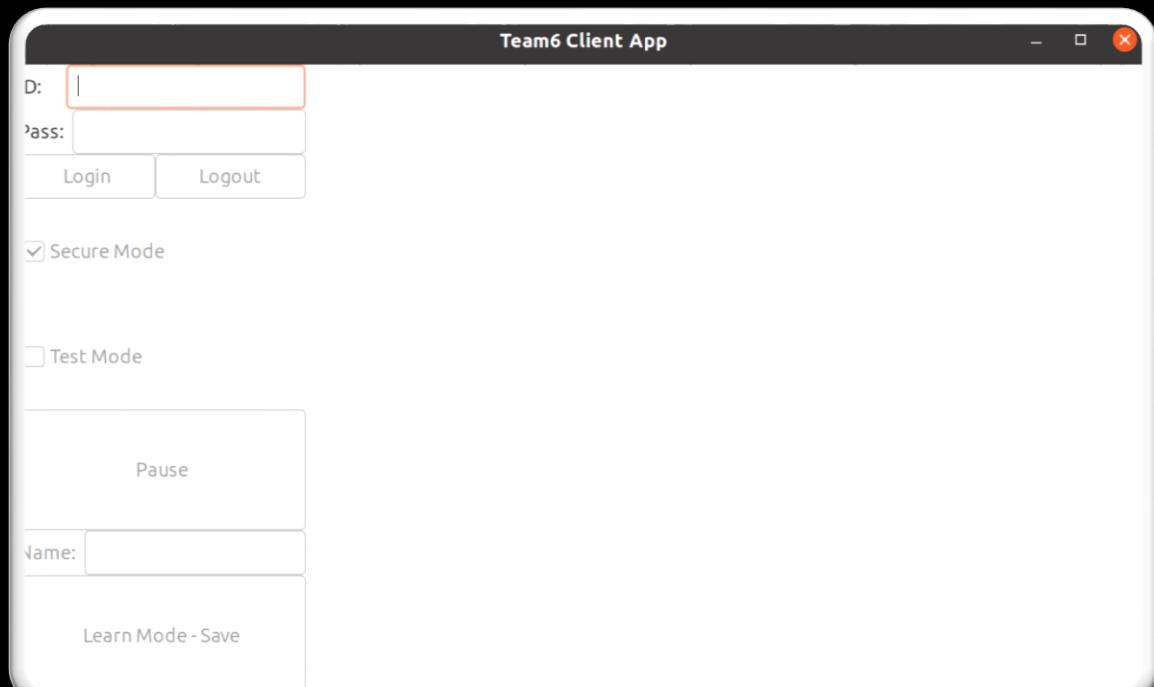
```
.  
  └── source  
      ├── client  
      │   └── src  
      │           // client source codes  
      ├── common  
      │   └── keys  
      │       ├── ca  
      │       │   // self signed root certificate  
      │       ├── client  
      │       │   // CA signed certificate & Private key for client  
      │       └── server  
      │           // CA signed certificate & Private key for server  
      └── libs  
          └── libcertcheck  
                  // crypto library to support crypto API and to verify the integrity of keys.  
                  // the root key for the crypto API are included as a string with obfuscated.  
  └── server  
      ├── facenetModels  
      ├── imgs  
      ├── mtCNNModels  
      ├── src  
      └── trt_l2norm_helper  
              // TensorRT L2-Norm Helper
```

Construct architecture

Client State Diagram



Construct architecture



Construct architecture

Let me show you demo clip..