# Security Development for Tartan system

2021/6/17 Team 4

# Introduction to Tartan system

System overview

- Server(Camera) & Client(Monitoring system) for live streaming
- Facial recognition with Database
- Wireless Network

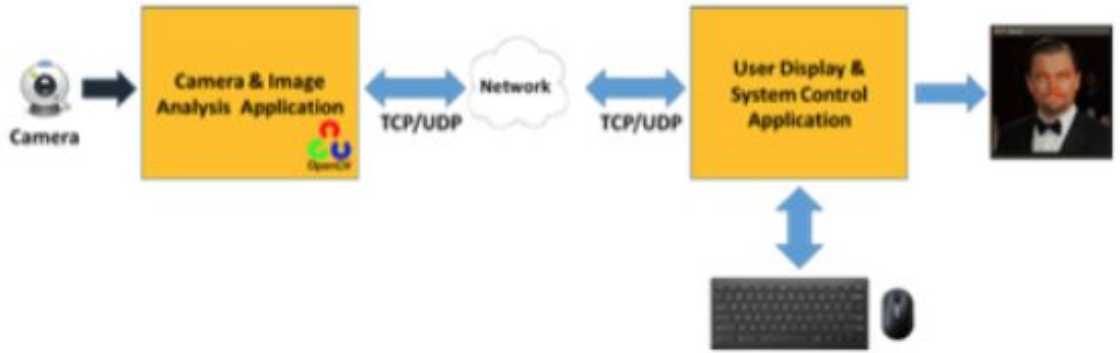



Figure 1: High-level design

# Project Goals

- Security Development for Tartan system

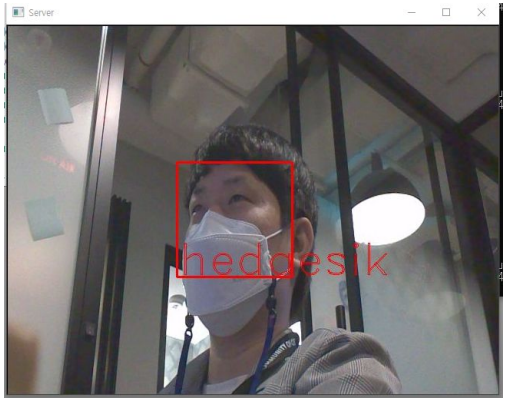
- Business perspective
  - Think of this system as MVP(minimum viable product)
  - Try to avoid implementing fancy features such as GUI.

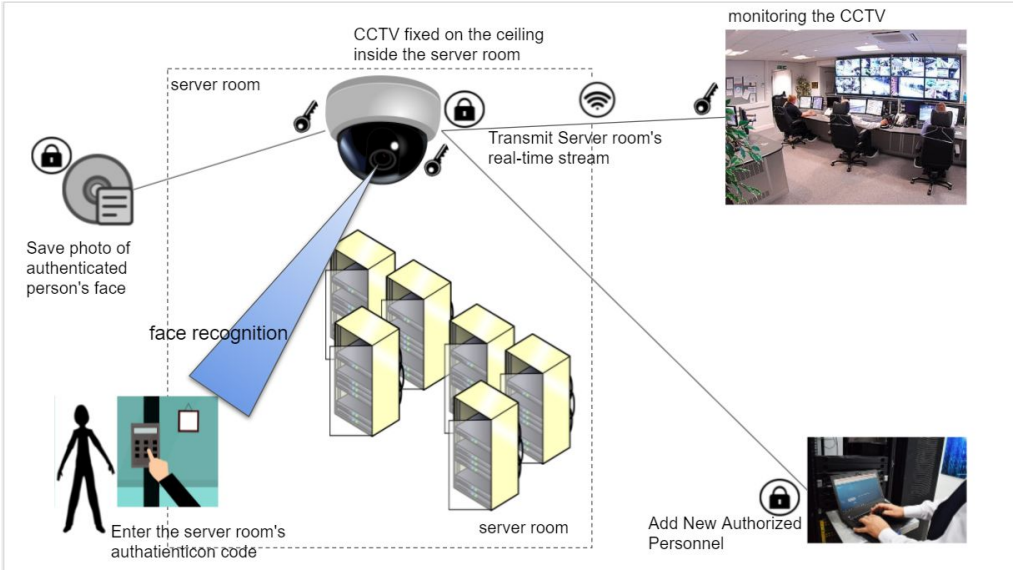
- Security Goals
  - Focused on enhancing security of the product.
  - Should be designed to achieve three principles.

# Application

## Overview

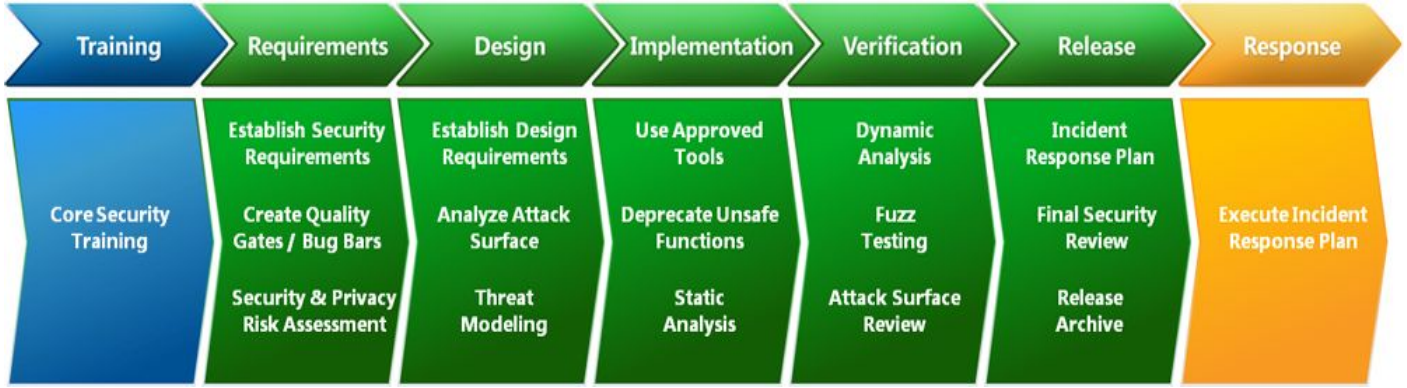
- Monitoring and tracking people with CCTV in the server room
- Detecting unauthorized people with facial recognition
- CCTV in the server room
- Monitoring system in the security room



# Security Development Process

- Based on MS-SDL
  - Some items on each stage are excluded due to the project development scope



# Requirements - Functions

- Security agents can watch and identify people through the real-time video streaming.
- A manager can access CCTV and register/unregister authorized people.
- The system should provide facial recognition.
- A manager can check the past dis/connection records.
- A manager can check the log file to see who entered the server room and when they did.

# Requirements - Security

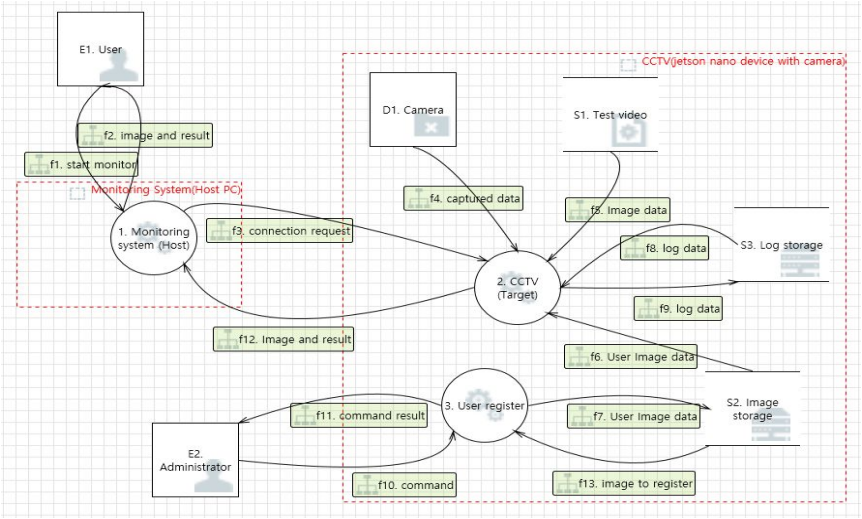
To achieve the security goals, the following security requirements are defined to identify and analyze possible threats and apply the derived mitigation.
These requirements were derived through threat analysis and mitigation measures.

- Secure network
  Network sections between CCTV and the monitoring system are encrypted.
- Personal information encryption
  Relevant data to privacy information should be encrypted securely.
- Key protection
  Keys which are used for encryption should be kept safely.
- CCTV should identify who was in or out and save the information to a log file.

# Design - Threat Modeling 1/2

- Data Flow Diagrams
  Decompose the system into parts and show that each part is not susceptible to relevant threats.

- Employ threat modeling using followings
  - STRIDE
  - PnG



# Design - Threat Modeling 2/2

- STRIDE

| Threats | Spoofing | Tampering | Repudiation | Information Disclosure | Denial Of Service | Elevation Of Privilege |
|---|---|---|---|---|---|---|
| 82 | 22 | 5 | 8 | 9 | 17 | 21 |

- PnG

| Threats | Persona 1 | Persona 2 | Persona 3 |
|---|---|---|---|
| 8 | 4 | 3 | 1 |

Jeff

An insider who is morally wrong and angry about incentives.

**Jeff**, who designed a CCTV system in his company. He has been working for this company as a network engineer. But for some reason he didn't get any incentive from the company, and he thought it was unfair.

**Motivation :**
Having complaints about incentives.
He got an offer from someone who wants to break into the building to get some information and accepted to help him.

**Goal :**
Unauthorized person who needs information can break into the server room where it is stored.

**Skills :**
knowledge of intra network system, knowledge about CCTV recognition algorithm, network skills, network hacking

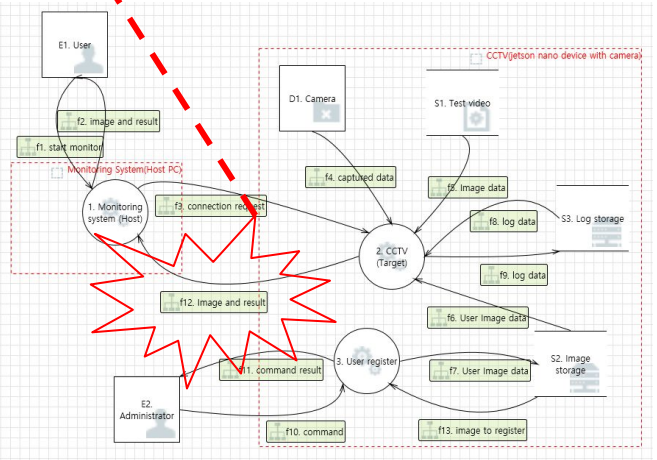
Misuse case 1

1. Jeff tries to log in CCTV with the default ID/password via ssh.
2. After login to the CCTV, Jeff adds a photo of the person who wants to infiltrate and register him as an authorized person.
3. Jeff wants him to be shown as an authorized person and let him pass through the CCTV and safely enter into the server room.

# Design - Mitigation

| Id | Title | Category | Priority | Description |
|---|---|---|---|---|
| 10 | Spoofing the 1. Monitoring system (Host) Process | Spoofing | High | 1. Monitoring system (Host) may be spoofed by an attacker and this may lead to information disclosure by 2. CCTV (Target). Consider using a **standard authentication mechanism** to identify the destination process. |

| Methods | pros | cons |
|---|---|---|
| IP/MAC | Implementation is the simplest | Since an attack that modifies IP and MAC is possible, spoofing cannot be reliably prevented. |
| ID/Password | Implementation is simple. | To prevent the password from being exposed, the communication section must be encrypted, and a module for user credentials is required.<br>If exposed to sniffing attacks, it can be neutralized. |
| **Certificate** | **the most effective authentication** | **There is a burden of creating, distributing, and managing certificates.** |



# Design - Risk Assessments

## OWASP

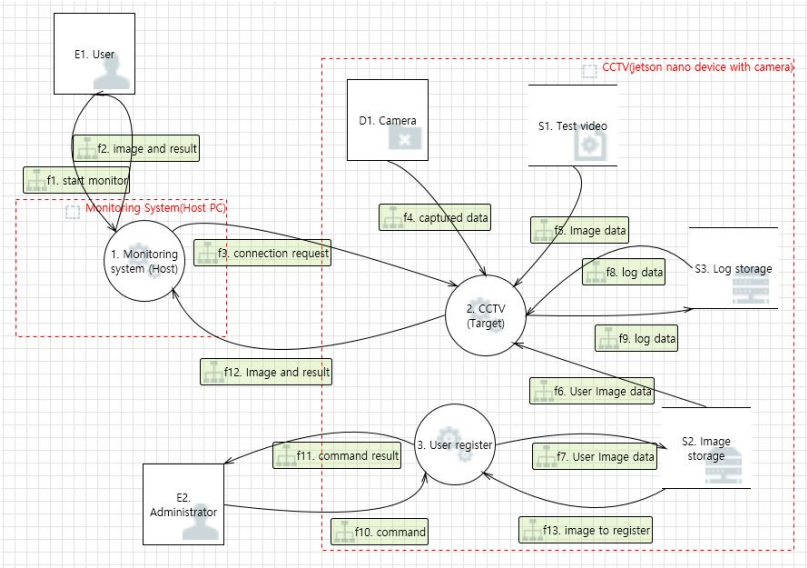
| Threats | Mitigated | High | Medium | Low |
|---|---|---|---|---|
| 90 | 25 | 11 | 45 | 9 |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| f12. Image and result | #11- Potential Lack of Input Validation for 1. Monitoring system (Host) [Tampering]<br>#11. 1. Data flowing across f12. Image and result may be tampered with by an attacker. This may lead to a denial of service attack against 1. Monitoring system (Host) or an elevation of privilege attack against 1. Monitoring system (Host) or an information disclosure by 1. Monitoring system (Host). Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach. | Threat Agent | Skill level | 3 - Network and programming skills | 6.25 | HIGH | Technical Impact | Loss of confidentiality | 4 - Minimal critical data disclosed, extensive non-sensitive data disclosed | 4.875 | MEDIUM | High |
| | | | Motive | 9 - High reward | | | | Loss of integrity | 7 - Extensive seriously corrupt data | | | |
| | | | Opportunity | 4 - Special access or resources required | | | | Loss of availability | 7 - Extensive primary services interrupted | | | |
| | | | Group Size | 4 - Intranet users | | | | Loss of accountability | 7 - Possibly traceable | | | |
| | | Vulnerability | Ease of discovery | 7 - Easy | | | Business Impact | Financial damage | 3 - Minor effect on annual profit | | | |
| | | | Ease of exploit | 5 - Easy | | | | Reputation damage | 7 - | | | |
| | | | Awareness | 9 - Public knowledge | | | | Non-compliance | 3 - | | | |
| | | | Intrusion detection | 9 - Not logged | | | | Privacy violation | 1 - | | | |

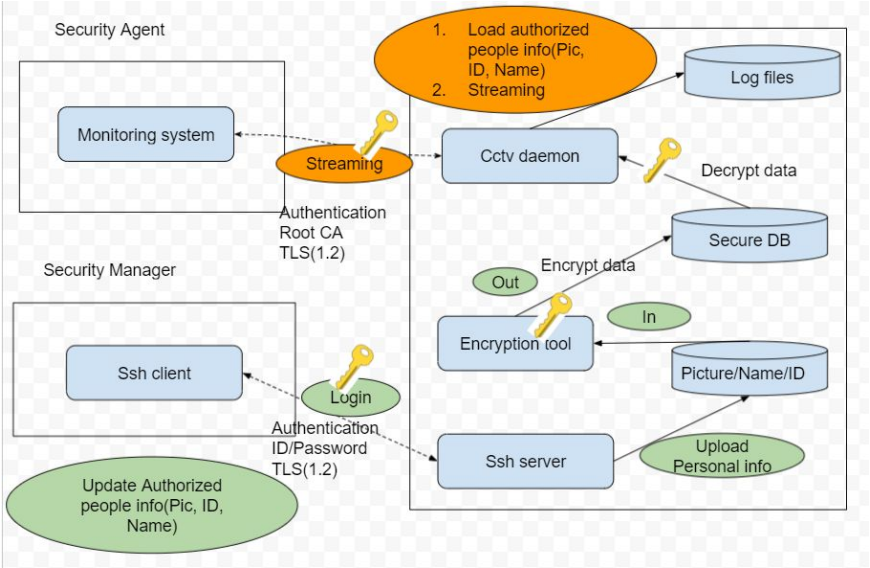
# Design - Mitigations - Overall

## Risk mitigations

DFD

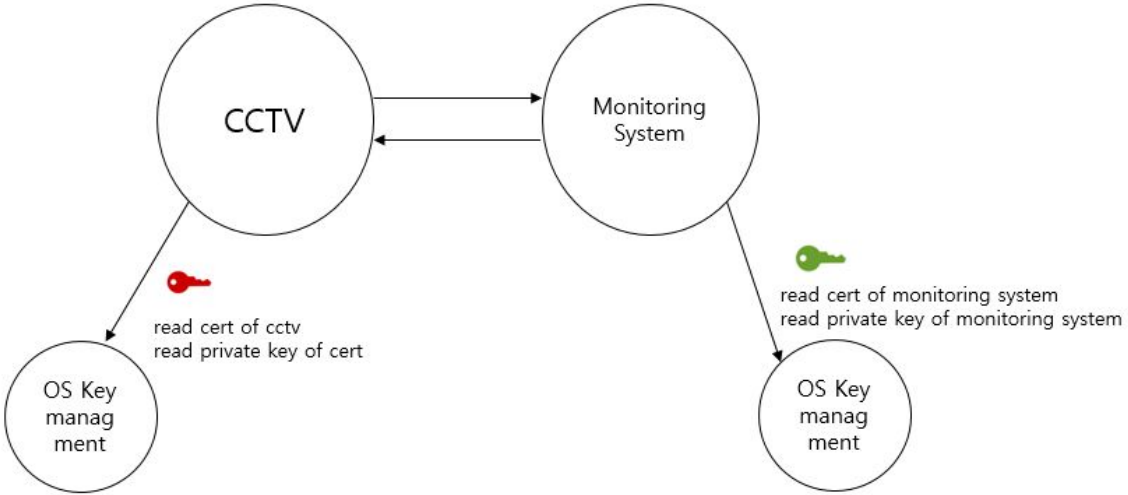


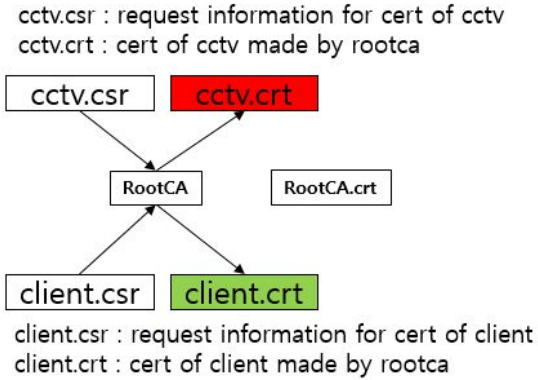

Scenario view



# Design - Mitigations - Mutual Authentication

## TLS connection using PKI

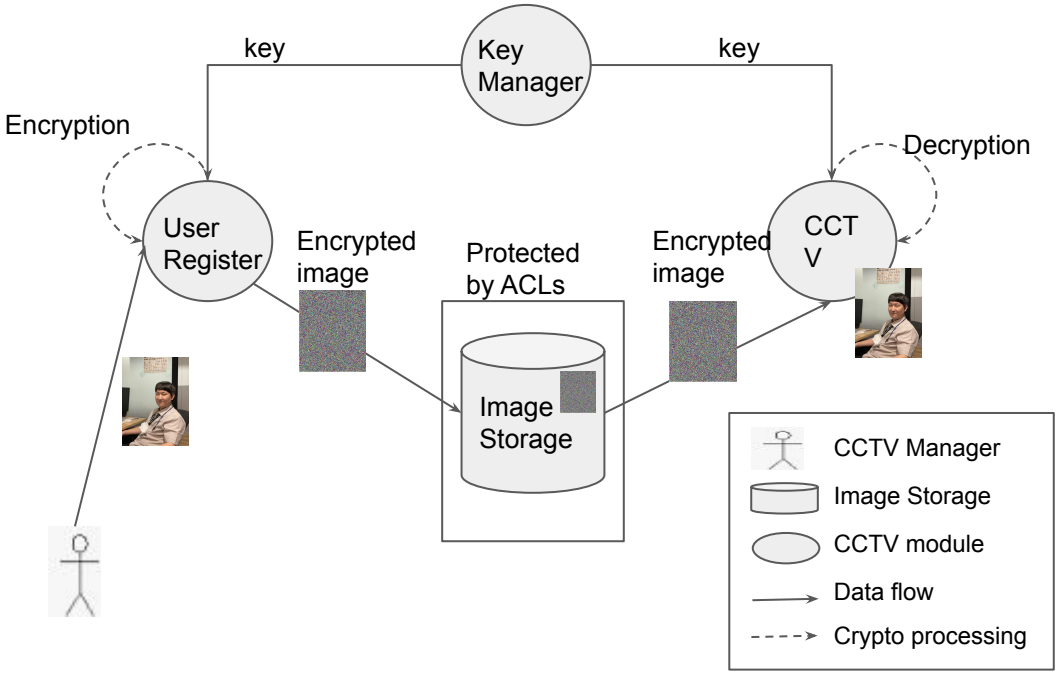



1.Verify monitoring system by cert of monitoring system with root CA cert.
2.Verify CCTV by cert of CCTV with root CA cert
3.If authentication is success, then network transport channel is encrypted by TLS1.2

# Design - Mitigations - User Info protection

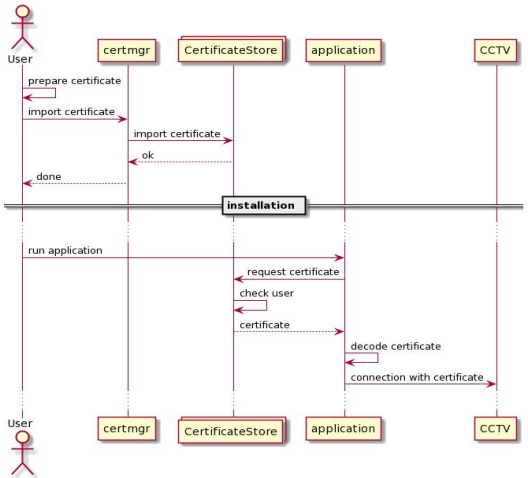
## User image protection overview



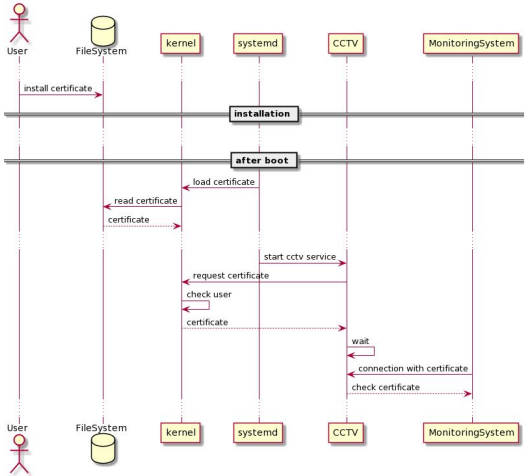

- User image file
  - User image file is encrypted using AES-128 CBC in User Register
  - Encrypted user image file is decrypted using AES-128 CBC in CCTV
  - AES-128 Key is provided by KeyManager
- User image filename
  - User image file name is encrypted using AES-128 CBC in User Register
  - Encrypted file name is Base64 encoded in User Register
  - Apply substitution to slash(/) characters in Base64 encoded file name
  - Encoded file name is Base64 decoded in CCTV
  - Encrypted filename is decrypted using AES-128 CBC in CCTV
  - AES-128 Key is provided by KeyManager

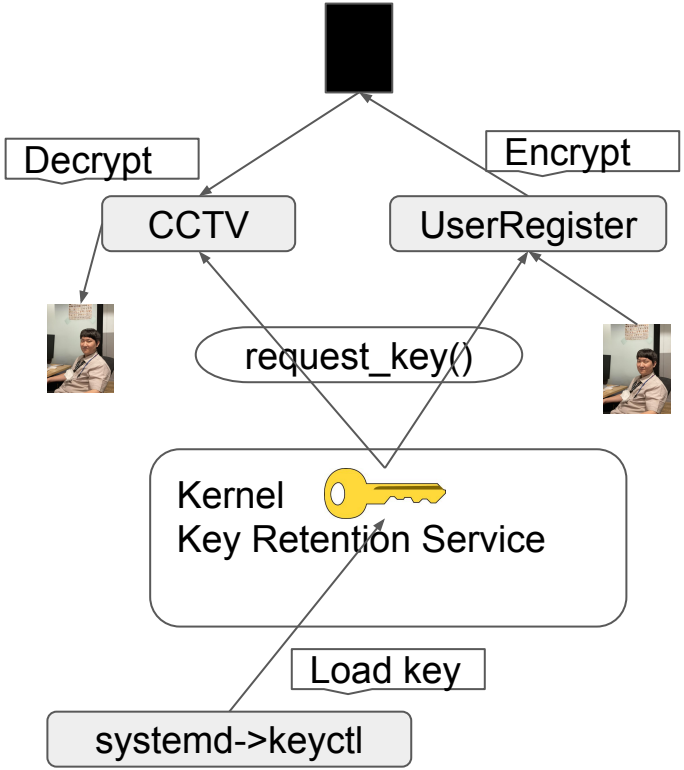
# Design - Mitigations - Key Management

Monitoring System(Windows)




CCTV(Linux)



# Implementation

Secure Coding

- Static Analysis
  Analyzing the source code prior to the compilation provides a highly scalable method of security code review and helps ensure that secure coding policies are being followed.

**Result of static analysis tools**

| Tools | Target | Total Detected | false positive | To mitigate | Remark |
|---|---|---|---|---|---|
| **Sonarcloud** | Monitoring system components | 247 | 247 | 0 | - 218 issues are detected as a code smell type, which is false positive and the rest are minor issues. |
| **Code x-ray** | all components | 24 | 24 | 0 | - The issues detected by Blocker(1 issue) and Major(4 issues) are about the files(out of scope) or have no effect on the code. |
| **Flawfinder** | CCTV and user register components | 48 | 43 | 5 | - 5 issues are fixed.(2 issues related to integer overflow, 3 related to statically-sized buffer) |

*) We decided to fix the issues found in the static analysis if necessary for items greater than Major (FlawFinder Level 3).

# Verification

## Test Report

| Test Cases | Pass | Fail |
|---|---|---|
| 19 | 19 | 0 |

**Test Case #1(Functional Requirement)**
**Purpose**
- This TC verifies the real-time CCTV person detection function of the CCTV system.

**Precondition**
- The Monitoring System is installed.(also cert. key is installed)
- A Security agent is logged in.
- CCTV is running and streaming camera video.

**Test Constraints**
- Only one monitoring system can be connected to CCTV.

# Deliverables

https://github.com/hijang/lsc_cctv

# Lessons learned

- The more we know about the system, the better we can design and implement threat mitigations
- If I had realized earlier that my mentor was also a stakeholder who should share information, I would have been able to get more help.