Water Blaster Security Project

Team2. S.H.I.E.L.D

Security Analysis Document History

| No | Write Date | Doc Version | Writer | Description |
| --- | --- | --- | --- | --- |
| 1 | 2024.06.11 | Draft 0.1 | Janghyun Ko | Initial document |
| 2 | 2024.06.24 | Version 1.0 | Hoonmin Lee | First Official Version |

Table of Contents

[1.1 Project Name](#_qx6ubyxsxk5e)

[1.2 Project Purpose](#_i93pnpfovwao)

[1.3 Project Member & Role](#_ptiycfn7de45)

[1.4 Project Schedule](#_wvfh7qs14lp4)

[2.1 Overview](#_h7blosct1l2c)

[2.2 System Architecture](#_nf6aw96e2d9w)

[2.2.1 Modes of Operation](#_hcmlcm7bw8ab)

[2.3 Functional Requirement](#_qq89rl6dhow4)

[2.4 Security Requirement](#_it8y1koetfr)

[3.1 Asset Identification](#_3vlhcyau6qsq)

[3.1.1 Identify system components and asset using DFD](#_55fnx260jz7)

[3.1.2 Key Assets](#_2bf8nnlmijvn)

[3.1.3 Evaluating connection types and security properties](#_287i4ng61cw5)

[3.2 Threat & Vulnerability Analysis](#_s8ylsg36v1ac)

[Identify threats to the system and its vulnerabilities.](#_fh5q9s15oe74)

[3.2.1 Case 1:Denial of Service(DoS)](#_bc1fnvz81o7c)

[3.2.2 Case 2: Information Disclosure](#_4r58nt9ifju6)

[3.3. Security Risk Assessment](#_sp3om5gfxa49)

[3.3.1 Assess Risks](#_vyscixex8hdk)

[3.3.2 Impact Rating](#_c1i1ej9q162b)

[3.3.3 Attack Feasibility Rating](#_9l3vtubsd5hk)

[3.4. Mitigating Threats](#_h31tot9thl0c)

[3.4.1 Specify mitigation strategies](#_54d441msenw4)

[3.4.2 Security Requirement Analysis & Design](#_fmxtov9nwsno)

[3.4.2.1 Operation and Roles of Individual Security Components](#_w3teecy5e5ws)

[==수정필요(auth)==](#_sjidfofq0q7)

[3.4.2.2 Design for System Protection](#_yhwgd9pratn0)

[3.4.2.2.1 Component Interaction](#_cl5koln5r0yp)

[3.4.2.2.2 Secure Communication & Authentication](#_ygr9e5xaguys)

[4.1 Application](#_19wt6us0oun6)

[4.1.1 OpenSSL Library](#_q0te7efh5sa7)

[4.1.2 Credential Manager Library](#_ipsxawyupqs0)

[4.1.3 Compile Option](#_86igtdab5av2)

[Appendix A : Security Analysis Result](#_prnggxtwgoai)

[= The end ======](#_iwr0x7d325f6)

[3.4.2.2.1 User Auth Flow Diagram](#_svj8g732twm4)

[Self Signed Root Certificate](#_qg6f5z1ka9tl)

[Running environment: PC (Windows)](#_qg6f5z1ka9tl)

[DemoCannon](#_qg6f5z1ka9tl)

[User Authenticator](#_qg6f5z1ka9tl)

[server storage](#_qg6f5z1ka9tl)

[Server Certificate](#_qg6f5z1ka9tl)

[ECDH Secret Key](#_qg6f5z1ka9tl)

[ENC Secret Key](#_qg6f5z1ka9tl)

[Execution environment: Server (Linux)](#_qg6f5z1ka9tl)

[Build Server](#_qg6f5z1ka9tl)

[Secure Coding](#_qg6f5z1ka9tl)

[Sanitizer](#_qg6f5z1ka9tl)

[Private Server](#_ls1h4jilssvk)

[Design user authentication](#_5qkrse37vcpb)

[User Auth Flow Diagram](#_ufwe8nydb7kg)

[List of Login\_MSG(return value)](#_39sc2lckel5h)

[Security Analysis Korean Document [Draft 0.1]](#_a00tfzvz2z2h)

[공격 표면(Attack Surface) 분석 및 보안 대책](#_cl8kxkje9jv1)

[Security Analysis English Document [Draft 0.1]](#_htfyslipvw21)

[Attack Surface Analysis and Security Measures](#_zac5pwylbdl9)

[=3.Security Analysis 뒤로](#_bpxrwl2o3k9c)

[=위치 조정 필요====](#_2f7bruldrodr)

[Important Notice](#_5wfidm7bwar)

[Root CA entity](#_szg2aqvx917n)

[Configure Root CA](#_2bjaf3w6p4jv)

[Operate Root CA](#_ruxnc9nzvm60)

[LG CA entity](#_7qkf5cg06iy4)

[Configure LG CA](#_ugfvt2q5nrse)

[Operate LG CA](#_xrp994bf16na)

[3rd party entity](#_4wzl8prrg4qn)

[Configure 3rd party](#_4jhj0y10ecfx)

[Operate 3rd party](#_onhmtro1dsxr)

1. Project Summary

# 1.1 Project Name

Water Blaster Security Project

# 1.2 Project Purpose

The project sponsor, SolveIt Inc., wants to launch a new product line of robot kits for educators and Do-It-Yourself (DIY) experimenters. Solvit, Inc initiated two development efforts;

1. To improve the design and functionality
2. To improve the security of the solution.

My team is chartered to improve the software security of the robot system, and specifically the Remote User Interface.

Key business and technology drivers for security:

1. Ensure safe and reliable operation of the robot system
2. Implement measures to mitigate architectural concerns of performance, safety, communication privacy and security, proof of identity
3. Design and implement added on Remote User Interface security requirements
4. Analyze and describe security threat and risk to safety vulnerabilities, including: arbitrary code / command execution, denial of service, information disclosure, data tampering, elevation of privileges, and spoofing / repudiation

# 1.3 Project Member & Role

| Name | Role | Major Task |
| --- | --- | --- |
| Hoon Min Lee (jennyleelg2004@gmail.com) | Project Management | Security - Analysis, Design, Review, Documentation |
| Jang Hyun Ko (rheels87@gmail.com) | Project Management | Protocol - Analysis, Design, Implementation, Review, Documentation |
| Jin Woo Kim (deenu0224@gmail.com) | Server developer | Server - Requirement Analysis, Code Analysis, Design, Implementation, Review |
| Yoon Se Han (yoon312@gmail.com) | Server developer | Server - Requirement Analysis, Code Analysis, Design, Implementation, Review |
| Ji Hyun Jang (zzhyuny@gmail.com) | Client developer | Client - Requirement Analysis, Code Analysis, Design, Implementation, Review |
| Nam Il Lee (lni148686@gmail.com) | Client developer | Client - Requirement Analysis, Code Analysis, Design, Implementation, Review |
| David Belasco (dbelasco@andrew.cmu.edu) | Mentor | Provides listening, growth support, consultation and advice, Review |

# 

# 1.4 Project Schedule

| Role\Date | 1 Week  (6.4~ 6.10) | 2 Week (6.11~6.17) | 3 Week (6.18~6.24) | Mid Presentation  (6/26) |
| --- | --- | --- | --- | --- |
| **Design Team** | | | | |
| Requirement Analysis | ✔️ |  |  |  |
| Security  Analysis |  | ✔️ |  |  |
| Security Design |  |  | ✔️ |  |
| Documentation |  |  | ✔️ |  |
| **Server Team** | | | | |
| Requirement Analysis | ✔️ |  |  |  |
| Code Analysis |  | ✔️ |  |  |
| Implementation |  |  | ✔️ |  |
| Review |  |  | ✔️ |  |
| **Client Team** | | | | |
| Requirement Analysis | ✔️ |  |  |  |
| Code Analysis |  | ✔️ |  |  |
| Implementation |  |  | ✔️ |  |
| Review |  |  | ✔️ |  |
| **Project Management Team** | | | | |
| Project Planning | ✔️ |  |  |  |
| Resource Allocation | ✔️ |  |  |  |
| Risk Assessment |  | ✔️ |  |  |
| Status Reporting |  |  | ✔️ |  |

2. System Define

# 2.1 Overview

1. Remote User Interface

* Purpose: The Remote User Interface serves as the control hub for safely and reliably operating the robot system, providing capabilities to monitor the robot's state. It enables users to control various operational modes of the robot and monitor several functions such as real-time camera video feeds of the target course, the barbette's laser, computer vision for target identification, and active shots from the water gel cannon.
* Functions:
  + Mode Control: Allows users to switch between different operational modes of the robot.
  + Real-Time Monitoring: Provides monitoring of the robot's status and environmental data in real time.
  + Safety Mode Entry: Enables users to switch to safety mode when unsafe conditions are detected or undesirable actions occur.

2. Robot (Server)

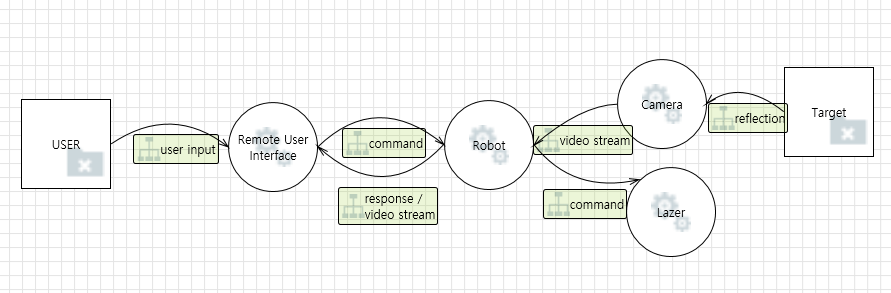
* Purpose: The Robot acts as the server within the network architecture, executing commands received from the Remote User Interface. It communicates via a network connection, autonomously or manually locating and identifying targets based on the commands.
* Functions:
  + Command Execution: Performs tasks as directed by the Remote User Interface.
  + Autonomous Operation: Capable of identifying and targeting without human intervention, using advanced algorithms.
  + Feedback Provision: Reports back to the Remote User Interface about the status of tasks and any operational issues.

3. WiFi Router (Network)

* Purpose: The WiFi Router facilitates the wireless communication between the Remote User Interface and the Robot, serving as a crucial component of the network infrastructure. It ensures secure and reliable data transmission.
* Functions:
  + Data Transmission: Ensures the secure and reliable transmission of commands, video feeds, and sensor data between the Remote User Interface and the Robot.
  + Network Security: Enhances security measures to protect the integrity and confidentiality of data during transmission.
  + Connection Management: Manages and optimizes the connection between the Remote User Interface and the Robot to ensure consistent performance.

# 2.2 System Architecture

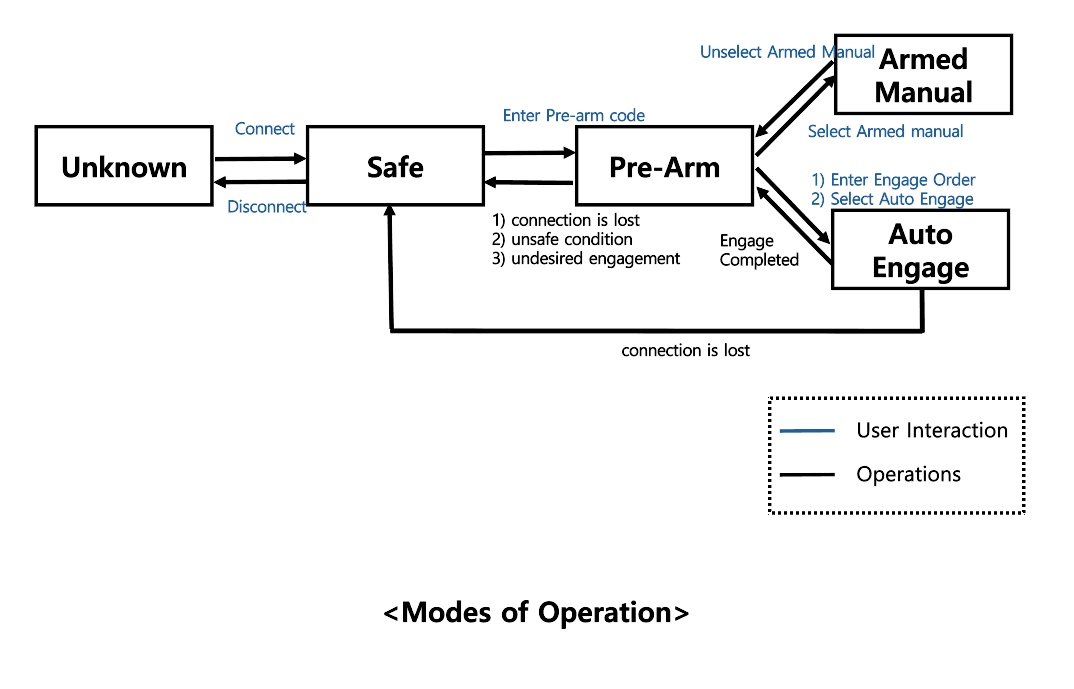
Data Flow Diagram (DFD) visualizes the data flow and key functions of a system consisting of remote user interfaces, robots, and networks. This diagram shows how these components interact with each other and the paths through which data is exchanged.



1. Remote User Interface:
   * Through this interface, users can change the operational modes of the robot and check its status.
   * The interface communicates with the robot over the network, sending user commands and receiving robot status updates.
2. Robot:
   * Executes tasks based on commands received from the Remote User Interface.
   * Utilizes various sensors and actuators to perform its functions, reporting its status back to the interface.
3. Network:
   * Handles the data communication between the Remote User Interface and the Robot.
   * Ensures reliable and secure transmission of control signals and status information using security protocols.

## 2.2.1 Modes of Operation

The robot's operating mode and its functions operate as shown in the picture below. These modes enable a variety of functions and safety measures, allowing flexible control depending on the situations the robot faces.



1. **Unknown Mode**
   * Initial state, waiting for network connection.
2. **Safe Mode**
   * The default mode after establishing a network connection.
   * Allows the user to disconnect (Disconnect) or enter a Pre-arm code.
3. **Pre-arm Mode**
   * Entered after inputting the Pre-arm code.
   * Use the J, I, L, M keys to move the cannon left, up, right, and down respectively.
   * Possible to switch to Armed Manual mode or Auto Engage mode.
4. **Armed Manual Mode**
   * Selected from Pre-arm mode.
   * Allows manual control of the robot and correction of errors.
   * Returning to Pre-arm mode is possible by deselecting this mode via the Remote User Interface.
5. **Auto Engage Mode**
   * Selected from Pre-arm mode after entering a target sequence.
   * Automatically identifies and targets objectives.
   * Once all targets have been engaged, the system returns to Pre-arm mode.

# 2.3 Functional Requirement

The functional requirements below are the tasks and functions that each component must perform in the system, and the implementation of each requirement can be checked through “Progress”.

| Functional Area | Requirement ID | Primary Requirement | Sub-Feature ID | Feature Description | Progress | Comments |
| --- | --- | --- | --- | --- | --- | --- |
| Remote User Interface | FR 1-1 | Control the robot's mode of operation. | | | O |  |
| FR 1-2 | Interface allows switching between different modes allow operator to monitor the robot’s state. | FR 1-2-1 | Interface allows switching between different modes. | O |  |
| FR 1-2-2 | Video feed of the barbette’s laser. | O |  |
| FR 1-2-3 | Computer vision to locate and identify targets. | O |  |
| FR 1-2-4 | Video of the water gel cannon active shots. | O |  |
| FR 1-2-5 | Current hit and miss counts (bonus feature) from the water gel cannon shots. | X |  |
| FR 1-3 | The interface shall allow the operator to enter a safe mode | FR 1-3-1 | when an unsafe condition is observed | 확인필요 |  |
| FR 1-3-2 | when undesired engagement of targets is observed | 확인필요 |  |
| FR 1-4 | The interface shall allow the user have the ability to take corrective action to put the robot back on course in pre-arm mode | | | O |  |
| FR 1-5 | The interface shall support the ability for an operator to stop robot actions at any time in auto engage mode. | | | O |  |
| FR 1-6 | The interface shall allow the operator to pause the robot’s autonomous operations at any time in auto engage mode. | | | X |  |
| FR 1-7 | The interface shall allow the operator to enter armed manual mode to manually engage targets using the camera and laser as the aim point. | FR 1-7-1 | In armed manual mode, the operator shall be able to rotate the barbette fuselage on its vertical axis. | O |  |
| FR 1-7-2 | In armed manual mode, the operator shall be able to raise or lower the gel water cannon through the digital servo. | O |  |
| FR 1-7-3 | In armed manual mode, the operator shall be able to turn the laser on and off. | O |  |
| FR 1-7-4 | In armed manual mode, the operator shall be able to fire the water gel cannon. | O |  |
| FR 1-7-5 | In armed manual mode, the operator shall be able to pan and tilt the camera as an extension of the barbette and cannon if necessary. | O |  |
| FR 1-8 | In Unknown mode, the interface shall be none of the features or modes shall be available until the user connects to their Gel Water Cannon. | | | O |  |
| FR 1-9 | In Safe mode, the user shall be able to disconnect the remote user interface from the robot. | | | O |  |
| FR 1-10 | In Safe mode, the user shall be able to enter a Pre-arm code to enter Pre-arm mode. | | | O |  |
| FR 1-11 | In Pre-arm mode, the user shall be able to move the cannon left, up, right, and/or down using the J, I, L, M keys respectively. | | | O |  |
| FR 1-12 | From Pre-arm mode, after entering a numbered target sequence, the user shall be able to select the Auto Engage mode. | | | O |  |
| FR 1-13 | From Pre-arm mode, the user shall be able to enter either Armed Manual or Auto Engage mode. | | | O |  |
| FR 1-14 | Pre-arm mode shall allow the user to take corrective action if any issues arise. | | | O |  |
| FR 1-15 | In Armed Manual mode, the operator shall be able to control the robot's pan/tilt servos, laser, and camera manually. | | | O | Self-implementation completed |
| FR 1-16 | The user must unselect the armed manual mode in the remote user interface to leave Armed Manual mode. | | | O |  |
| Robot | FR 2-1 | The system shall be in an Unknown mode when the user first starts the remote user interface | | | O |  |
| FR 2-2 | The system shall initiate to the Safe mode when a network connection is established between the robot and the remote user interface. | | | O |  |
| FR 2-3 | In Auto Engage mode, the robot shall operate without user intervention, locating and identifying targets. | | | O |  |
| FR 2-4 | In Auto Engage mode, the robot shall engage targets in order of number sequence. | | | O |  |
| FR 2-5 | In Auto Engage mode, the robot shall alert the user that the target was engaged through a message. | | | O |  |
| FR 2-6 | In Auto Engage mode, the robot shall record hits or misses of the target (bonus feature). | | | X |  |
| FR 2-7 | After all targets are engaged, the robot shall return to Pre-arm mode. | | | O |  |
| Fault Conditions | FR 3-1 | When firing in Auto Engage mode, if a gel projectile is not fired (dry fire), the operator shall place the robot in Pre-arm or Safe mode to inspect the cannon. | | | X |  |
| FR 3-2 | If the robot cannot locate the correct numbered target in Auto Engage mode, it shall move to the next target after a set amount of time, alerting the user that the target cannot be found. | | | O | Self-implementation completed |
| FR 3-2’ | If the robot cannot locate the correct numbered target in Auto Engage mode, it shall move into Pre-arm mode and alert the user to troubleshoot the issue. | | | O | Self-implementation completed |
| Unsafe Conditions | FR 4-1 | If the network connection between the robot and the remote interface is not established or is lost, the remote user interface shall indicate no connection. | | | X |  |
| FR 4-2 | If the network connection between the robot and the remote interface is not established or is lost, the operator shall not be permitted to perform any operations | | | O | Self-implementation completed |
| FR 4-3 | If the network connection to the remote interface is lost, the robot shall automatically stop actions until the network connection is recovered. | | | O | Self-implementation completed |
| FR 4-4 | If the network connection is recovered, the robot shall resume actions when commanded by the remote operators in manual or automatic mode. | | | O | Self-implementation completed |
| FR 4-5 | If the barbette traverses more than 15 degrees left or right of the center aim line, the robot shall automatically stop actions. | | | O | Self-implementation completed |
| FR 4-6 | If the barbette traverses more than 15 degrees up or down in elevation of the center aim line, the robot shall automatically stop actions. | | | O | Self-implementation completed |
| FR 4-7 | If the cannon continues to fire either repeatedly on a single target or while not aimed at a target, the robot shall automatically stop actions. | | | X |  |
| FR 4-8 | If the system is in an Unknown, Safe, or Pre-arm state, the robot shall not fire the cannon. | | | O | Self-implementation completed |

# 2.4 Security Requirement

Focusing on the security requirements of the remote user interface is because it is the passage through which interaction and data transmission between the user and the system take place, and strengthening this part is very important to improve the security level of the entire system. Security Requirements are mainly It focuses on the Remote User Interface. Implementation of each requirement can be checked through “Progress”.

| Functional Area | Requirement ID | Primary Requirement | Sub-Feature ID | Feature Description | Progress | Comments |
| --- | --- | --- | --- | --- | --- | --- |
| Remote User Interface | SR 1 | Set up a username and password for the Remote User Interface. | SR 1a | Ensure the password is secure, minimum of 10 characters, includes numbers and symbols. | O | Self-implementation completed |
| SR 1b | Ensure the username is unique and does not conflict with another username. | O | Self-implementation completed |
| SR 1c | Force administrators to reset all passwords periodically (at least once a month). | O | Self-implementation completed |
| SR 1d | Lock the account for one hour if the incorrect password is entered more than three times. | O | Self-implementation completed |
| SR 1e | Provide administrators the ability to recover or change passwords. | O | Self-implementation completed |
| SR 2 | Initiate a TCP server-client connection with the robot. | SR 2a | Connection allows commands to be given to the robot. | O |  |
| SR 2b | Connection allows information to be received from the robot. | O |  |
| SR 2c | Connection allows modification of network settings on the Raspberry Pi and its credentials. | X |  |
| SR 3 | Provide a graphical user interface with comprehensive functionalities. | SR 3a | Buttons/Text input for login. | O | Self-implementation completed |
| SR 3b | Buttons/Text input to connect the robot via a wireless network. | O |  |
| SR 3c | Window to display the camera feed from the barbette system. | O |  |
| SR 3d | Buttons to toggle between system modes. | O |  |
| SR 3e | Buttons to toggle laser on and off. | O |  |
| SR  3f | Buttons to toggle camera feed on and off (Bonus feature). | O | Self-implementation completed |
| SR 3g | Text display of the robot’s current actions (e.g., searching for a target). | O |  |
| SR 3h | Text display if an unsafe condition is present. | X |  |
| SR 3i | Text display if a fault condition is present. | X |  |
| SR 3j | Text display if a network connection is lost. | O | Self-implementation completed |
| SR 3k | Text display if the connection with the robot is lost. | 확인필요 |  |
| SR4 | Allow transitioning between operational modes. | SR 4a | Enable user to switch between Unknown, Safe, Pre-Arm, Armed Manual, and Auto Engage modes. | O |  |
| SR 4b | Alert users of normal operating actions, fault conditions, and unsafe conditions. | X |  |
| SR 4c | Allow the user to perform basic corrective safety or operational actions through the GUI. | O |  |
| SR5 | Provide an audit log of commands entered/issued to the robot. | | | X |  |
| SR6 | Provide an audit log of information provided by the robot. | | | X |  |

3. Security Analysis

A strategic approach to meeting the functional and security requirements of the system and improving the security architecture of the overall system is as follows: First, identify the key assets of the system. Next, the vulnerabilities of these assets are identified, and the security threats that may arise through those vulnerabilities are analyzed and evaluated. Based on this analysis, requirements for mitigating security risks are derived to enhance the safety of the system. This process plays an important role in systematically managing and improving system-wide security.

The primary objectives of security analysis are:

* Identify network components and assets
* Evaluate the security properties of each component
* Define potential threat scenarios
* Assess the likelihood and impact of attacks
* Derive risk treatment decisions

The security analysis is conducted based on the following criteria:

1. STRIDE Model

The STRIDE model consists of six threat categories:

* Spoofing
* Tampering
* Repudiation
* Information Disclosure
* Denial of Service
* Elevation of Privilege

1. Security Properties

The security properties of each component are:

* Confidentiality
* Integrity
* Availability

For this purpose, in Chapter 3, security analysis activities were conducted in the format below.

# 3.1 Asset Identification

## 3.1.1 Identify system components and asset using DFD

We identified the system components and assets using Data Flow Diagrams (DFD). The identified items were categorized into servers, networks, and applications.

* Servers: Robot Control System
* Network: Wifi Router
* Applications: Remote user interface Control System,

#### 

## 3.1.2 Key Assets

The key assets that must be maintained in the Gel Water Blaster Cannon application system are:

* Control software:
  + Servers: Robot Control System
  + Software that controls the robot and its remote user interface.
  + All codes and algorithms associated with control instructions.
* Data:
  + Robot status information, path data, target identification data, etc.
  + User credentials (username, password, etc.)
  + System logs and audit records.
* Network communication:
  + TCP/IP communication between robot and remote user interface.
  + Network settings and configuration data.
* Secure assets:
  + User authentication mechanisms (e.g. password, security token)
  + Encryption keys and certificates.
* Operating mode and status information:
  + Different operating modes of the robot (safe mode, free-arm mode, armed manual mode, etc.)
  + Real-time condition monitoring data (camera feeds, laser status, computer vision information, etc.)

## 3.1.3 Evaluating connection types and security properties

We evaluated the connection types and related security properties between each component. The security properties were classified according to the CIA (Confidentiality, Integrity, Availability) criteria.

* Security Properties:
  + Confidentiality
  + Integrity
  + Availability

Security analysis activities were conducted to protect the above assets, and requirements for taking action were derived based on the analysis.

The goal is to ensure safe operation of the system along with guarantees of Confidentiality, Integrity, Availability, Authentication, and Non-repudiation for the assets mentioned above.

# 3.2 Threat & Vulnerability Analysis

#### Identify threats to the system and its vulnerabilities.

The threat analysis content for assets that need to be protected in servers, networks, and applications derived from the DFD mentioned above is the same as the attached Excel file.

Major assets were selected for each component (wifi router, robot, user application UI focus) and threat analysis was performed using the STRIDE technique according to their security properties.

The document contains the results of a threat and vulnerability analysis of the system and an analysis of potential threats to the system, including threats that could affect specific parts of the system or assets.

Below are some excerpts:



## 3.2.1 Case 1:Denial of Service(DoS)

* Components and Assets:
  + Function: Network
  + Components: Wifi Router/Robot
  + Assets: TCP Packet
* Connection Type and Security Properties:
  + Connection Type: TCP Connection (Robot to Wifi Router)
  + Security Properties: Confidentiality: -, Integrity: -, Availability: O
* Threat Category: Denial Of Service
* Threat Scenario:
  + If network traffic becomes saturated, normal data transmission may be disrupted, preventing the robot's control signals or status updates from being sent to the remote user interface.
    - Network issues can prevent robots from taking emergency action, such as going into safe mode or detecting danger and alerting operators.
    - If the bot is approaching the target, commands that should be adjusted in real-time may not be delivered, which may cause accidents or incorrect operations.

## 3.2.2 Case 2: Information Disclosure

* Components and Assets:
  + Function: Network
  + Components: Wifi Router/Robot
  + Assets: TCP Packet
* Connection Type and Security Properties:
  + Connection Type: TCP Connection (Robot to Wifi Router)
  + Security Properties: Confidentiality: -, Integrity: O, Availability: O
* Threat Category: Information Disclosure
* Threat Scenario:
  + An attacker can intercept TCP communication between the robot and the router and modify the data. This causes the robot to perform incorrect actions if an attacker modifies the robot's command data.
  + If an attacker modifies or deletes the robot's safety-related messages, the robot's safety protocols may not function properly. This could result in the robot operating in a dangerous state or failing to enter safe mode when needed, which could lead to a serious accident.

# 3.3. Security Risk Assessment

## 3.3.1 Assess Risks

The standards for Impact Rating and Attack feasibility Rating are as shown in the table below.



Risk treatment decisions are set to “Reducing the risk” in the following cases, otherwise set to “Retaining the risk”.

1. When Impact Rating is “Medium” or “High”
2. When the Attack feasibility Rating is “Medium”, “High”, or “Very High”

## 3.3.2 Impact Rating

Impact is determined by considering the operational impact rating and privacy impact rating.





The operational impact rating evaluates the impact on the functioning and operation of a system or process. It mainly considers the following factors:

1. Functional Impact
2. Continuity of operations
3. Business Impact

The privacy impact rating assesses the impact on the protection of personal information. It mainly considers the following factors:

1. Potential for personal information disclosure
2. Potential for Sensitive Information Leakage
3. Legal and Compliance

## 3.3.3 Attack Feasibility Rating

It evaluates different attack possibilities and provides a detailed explanation of why a specific score is given for each attack scenario.





The Attack feasibility rating standards are as follows.

* [Elapsed time]: How much time does it take for the attacker to carry out the attack?
  + The longer the elapsed time, the more difficult it is for an attack to be successful. Conversely, the shorter the time, the easier it is to perform.
* [Expertise]: What skills and abilities does an attacker need to attack?
  + If an attacker can successfully carry out an attack without any special skills or knowledge, many people can easily attempt the attack and expose the system to a wide range of risks.
* [Expertise]: What skills and abilities does an attacker need to attack?
  + Description: If an attacker can successfully carry out an attack without any special skills or knowledge, many people can easily attempt the attack and expose the system to a wide range of risks.
* [Knowledge of items or components]: How much do you know about the structure, operation, and security vulnerabilities of the system needed to attack an attacker?
  + Description: If an attacker can easily attack a system without deep knowledge of its structure or operating principles, the number of attack attempts can increase significantly.
* [Window of Opportunity] Are there many or few temporal or situational opportunities for an attacker to carry out an attack?
  + Explanation: The point at which the system becomes vulnerable to an attacker occurs frequently, and the longer the period, the more opportunities for attack, and the higher the risk.
* [Equipment] When attacking, are the tools used by attackers special equipment used by experts or easily accessible equipment?
  + Description: The more expensive or specialized equipment you require, the more difficult the attack may be, or the easier it may be if you can do it with regular equipment.

Attack Feasibility Rating Rationale provides a detailed explanation of why a specific score is given for each attack scenario.

# 3.4. Mitigating Threats

## 3.4.1 Specify mitigation strategies

Appropriate security requirements were derived for cases where mitigation measures were required for the identified system threats.

## 3.4.2 Security Requirement Analysis & Design

Security requirements were derived to mitigate 16 identified security threats using the Microsoft Threat Modeling Tool.

The security requirements are the same as the attached Excel file. Here are some excerpts:



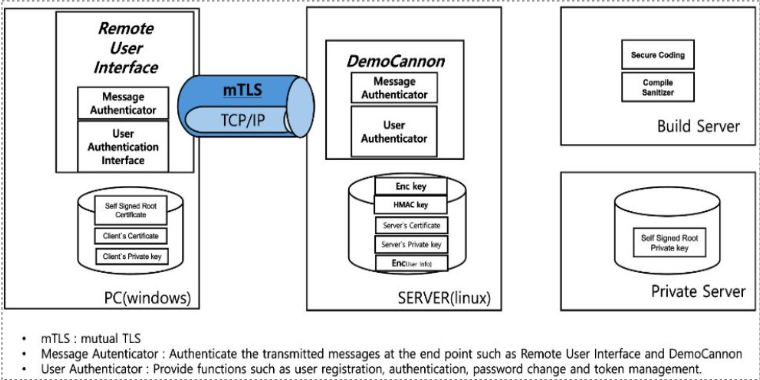
### 3.4.2.1 Operation and Roles of Individual Security Components

The role each component must perform and how it operates are as follows.

* TCP/IP Connection with TLS
  + During TLS handshaking, the server's certificate delivered to the client is issued by a Certificate Authority.
  + For the demo, a self-signed root certificate is generated and It issues a server`s certificate.
  + The self-signed root certificate is pre-placed on the client for server certificate validation.
  + The client generates a pre-master key, encrypts it with the public key extracted from the server's certificate, and sends it to the server.
  + The server decrypts the encrypted pre-master key with its private key.
  + Need additional discussion : Since the client environment is not trusted, mutual authentication is not performed.
* User authentication
  + User Authenticate Interface(Client) and User Authenticator(Server)
    - The client only provides an interface for authentication
    - The server performs user authentication.
  + Enroll a User Account Information
    - User account information such as user name and user passphrase is encrypted and stored on the server.
  + Verify a User Account
    - The hash of the user password sended by the client is compared with the hash stored on the server, and the result is responded to by the client.
  + Issue a token(separation of privilege)
    - If authentication is successful, the server generates a token and sends it to the client.
    - The server can distinguish whether an API is called by a user or an administrator through the token.

### 3.4.2.2 Design for System Protection

The Security Context View shows how each component interacts and how secure authentication and communication occurs. The remote user interface handles user authentication and maintains secure communication with the Democannon server via TLS. The servers are protected behind a firewall, and key security keys and certificates are stored securely within the servers. The build server is responsible for secure coding and code verification, while the private server holds the sensitive root private key.

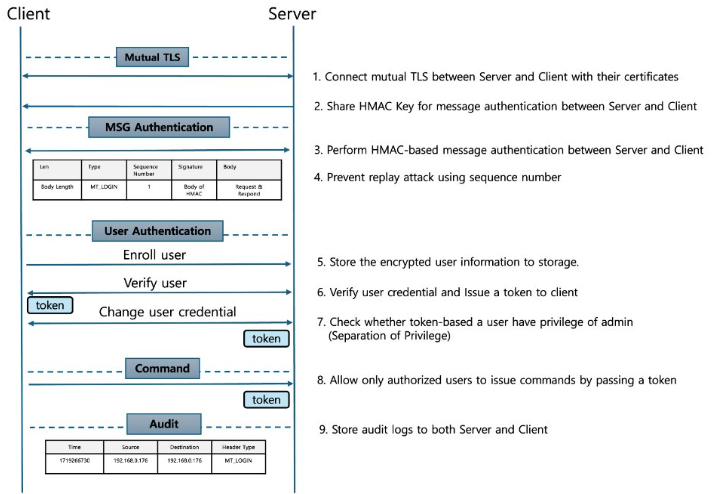


In the overall system protection design, how each component interacts and how secure authentication and communication should occur is as follows.

#### 3.4.2.2.1 Component Interaction

* Remote User Interface
  + User Authentication Interface
    - This is an interface that performs the authentication process so that users can access the system remotely.
  + Self Signed Root Certificate
    - A certificate used to ensure trustworthy communication and is used in self-signed form.
  + Running environment: PC (Windows)
    - The remote user interface runs on the Windows operating system.
* DemoCannon
  + User Authenticator
    - It is responsible for verifying the user's identity and processing authentication.
  + Server Storage
    - Server Certificate
      * This certificate is used to prove the reliability of the server.
      * Algorithm used to encrypt user information on the server: AES256
    - Execution environment: Server (Linux)
      * DemoCannon runs on the Linux operating system.
* Build Server
  + Secure Coding
    - Follow secure coding rules when writing code to prevent security vulnerabilities.
  + Sanitizer
    - It is a tool that finds and fixes potential defects or vulnerabilities in code.
* Private Server
  + Self Signed Root Private Key
    - This is the private key used to sign the root certificate and plays an important role in ensuring secure communication with the outside world.

#### 3.4.2.2.2 Secure Communication & Authentication





4. Build Configuration

# 4.1 Application

## 4.1.1 OpenSSL Library

1. **Download OpenSSL**:

- First, download the precompiled OpenSSL libraries suitable for your project's architecture (32-bit or 64-bit) from the official OpenSSL website(https://slproweb.com/products/Win32OpenSSL.html) or another trusted source.

2. **Prepare OpenSSL Directory**:

- Create a directory in your project structure where OpenSSL libraries and headers will reside. For example, `C:\OpenSSL` or `C:\Libraries\OpenSSL`.

3. **Include OpenSSL Headers**:

- Open your Visual Studio solution.

- Right-click on your project in Solution Explorer and select **Properties**.

- Navigate to **Configuration Properties > C/C++ > General**.

- Add the path to the OpenSSL `include` directory in **Additional Include Directories**. For example, if your OpenSSL headers are in `C:\OpenSSL\include`, add `C:\OpenSSL\include` to the list.

4. **Link OpenSSL Libraries**:

- Still in the project properties window, navigate to **Configuration Properties > Linker > General**.

- Add the path to the OpenSSL `lib` directory in **Additional Library Directories**. For instance, if your OpenSSL libraries are in `C:\OpenSSL\lib`, add `C:\OpenSSL\lib` to the list.

5. **Specify OpenSSL Libraries**:

- Under **Configuration Properties > Linker > Input**, add the required OpenSSL libraries to **Additional Dependencies**. Depending on your project needs, you might need to link against libraries like `libcrypto.lib` and `libssl.lib`.

## 4.1.2 Credential Manager Library

1. **Open Your Project**:

- Open your solution in Visual Studio 2022.

2. **Access Project Properties**:

- Right-click on your project in the Solution Explorer.

- Select **Properties**.

3. **Navigate to Linker Settings**:

- In the Project Properties window, go to **Configuration Properties > Linker > Input**.

4. **Add Additional Dependencies**:

- In the **Additional Dependencies** field, add `Advapi32.lib`.

## 4.1.3 Compile Option

- C++ : /permissive- /ifcOutput "x64\Release\" /GS /GL /W3 /Gy /Zc:wchar\_t /I"..\..\..\opencv\build\include" /I"..\..\Common" /I"C:\Program Files (x86)\Windows Kits\10\Include\\shared" /I"C:\Program Files (x86)\Windows Kits\10\Include\\um" /Zi /Gm- /Od /sdl /Fd"x64\Release\vc143.pdb" /Zc:inline /fp:precise /D "NDEBUG" /D "\_WINDOWS" /D "WIN32" /D "OPENSSL\_NO\_ASM" /D "\_UNICODE" /D "UNICODE" /errorReport:prompt /WX- /Zc:forScope /Gd /Oi /MD /FC /Fa"x64\Release\" /EHsc /nologo /Fo"x64\Release\" /Fp"x64\Release\LgClientDisplay.pch" /diagnostics:column

- Link: /OUT:"{PROJECT\_PATH}\x64\Release\LgClientDisplay.exe" /MANIFEST /LTCG:incremental /NXCOMPAT /PDB:"{PROJECT\_PATH}\x64\Release\LgClientDisplay.pdb" /DYNAMICBASE "libssl.lib" "libcrypto.lib" "Crypt32.lib" "kernel32.lib" "user32.lib" "gdi32.lib" "winspool.lib" "comdlg32.lib" "advapi32.lib" "shell32.lib" "ole32.lib" "oleaut32.lib" "uuid.lib" "odbc32.lib" "odbccp32.lib" /DEBUG /MACHINE:X64 /OPT:REF /PGD:"{PROJECT\_PATH}\x64\Release\LgClientDisplay.pgd" /SUBSYSTEM:WINDOWS /MANIFESTUAC:"level='asInvoker' uiAccess='false'" /ManifestFile:"x64\Release\LgClientDisplay.exe.intermediate.manifest" /LTCGOUT:"x64\Release\LgClientDisplay.iobj" /OPT:ICF /ERRORREPORT:PROMPT /ILK:"x64\Release\LgClientDisplay.ilk" /NOLOGO /LIBPATH:"{OPENSSL\_PATH}\lib" /TLBID:1

# 

Appendix

# Appendix A : Security Analysis Result

This section contains data and analysis results for activities that analyze threats and derive mitigation measures to enhance system security.

(Threat Analysis\_Team2\_제출용.xls)