Gel Water Blaster Cannon

Application

TIE Fighters (Team 3)  
  
14 June 2024

Table of Contents

[**0. Introduction** 4](#_Toc170197499)

[Team name 4](#_Toc170197500)

[Role and Responsibilities 6](#_Toc170197501)

[Terminology & Acronym 7](#_Toc170197502)

[**1. Schedule** 8](#_Toc170197503)

[**2. System Requirement** 9](#_Toc170197504)

[2.1. Functional Requirement 9](#_Toc170197505)

[2.2. Non-Functional Requirement 12](#_Toc170197506)

[**3. Security Goals** 13](#_Toc170197507)

[**4. Assets** 13](#_Toc170197508)

[**5. Threat Modeling** 13](#_Toc170197509)

[5.1. PnG 13](#_Toc170197510)

[5.2. DFD & STRIDE 14](#_Toc170197511)

[5.3. Traceability Matrix 22](#_Toc170197512)

[5.4. Security Requirements and Mitigations 23](#_Toc170197513)

[**6. System Design** 25](#_Toc170197514)

[6.1. Overall System Architecture 25](#_Toc170197515)

[6.2. Used Open Source 25](#_Toc170197516)

[6.2.1. Client 25](#_Toc170197517)

[6.2.2. Server 26](#_Toc170197518)

[6.3. Design with mitigation 26](#_Toc170197519)

[6.3.1. Secure communication 26](#_Toc170197520)

[6.3.2. PAM (Pluggable Authentication Modules) 28](#_Toc170197521)

[6.3.3. Two-factor Authentication 28](#_Toc170197522)

[6.3.4. Management of User account and password by cockpit 30](#_Toc170197523)

[6.3.5. Packet Filtering 31](#_Toc170197524)

[6.3.6. Audit logging 31](#_Toc170197525)

[6.4. Secure compile 32](#_Toc170197526)

[6.4.1. Server 32](#_Toc170197527)

[6.4.2. Client 33](#_Toc170197528)

[6.5. Secure coding 33](#_Toc170197529)

[**7. Verification** 34](#_Toc170197530)

[7.1. Overview of verification 34](#_Toc170197531)

[7.2. Test Case Design for Security Requirements 34](#_Toc170197532)

[7.3. Verification Result 34](#_Toc170197533)

[**8. Guide** 34](#_Toc170197534)

[8.1. Setup Guide 34](#_Toc170197535)

[8.1.1. Local PC 34](#_Toc170197536)

[8.1.2. Setup for two-factor authentication of Administrator 34](#_Toc170197537)

[8.1.3. Build guide for Client (User) 35](#_Toc170197538)

# **0. Introduction**

We will develop a secure implementation of the ‘Gel Water Blaster Cannon Application’ system that meets the given basic requirements.

## Team name









**Mentor: Cliff**



## Role and Responsibilities

|  |  |
| --- | --- |
| **Name** | **Responsiblity** |
| SungWook Kwon | Threat Analysist, Team leader |
| Byeongyul Kim | Server Developer |
| Tran Truong Thien Nguyen | Server Developer |
| Jae Hun Cha | Client Application Developer |
| Kyunglok Cho | Client Application Developer |
| Seok Hee Lee | Secure communication Developer |

## Terminology & Acronym

|  |  |
| --- | --- |
| **Term** | **Description** |
| Asset | An asset is a resource of value. It varies by perspective. To a business, an asset might be the availability of information, or the information itself, such as customer data. It might be intangible, such as a company's reputation. |
| Attack (Exploit) | An attack is an action taken that utilizes one or more vulnerabilities to realize a threat. |
| Attack Surface | Logical area (browser stack, infrastructure components, etc.) or physical area (hotel kiosk) that an attack may occur or originate from. |
| Authorization | The official management decision given by a senior organizational official to authorize operation of an information system and to explicitly accept the risk to organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, and the nation based on the implementation of an agreed-upon set of security controls [NIST SP 800-137, CNSSI 4009]. |
| Availability | Ensuring timely and reliable access to and use of information [NIST SP 800-137, 44 U.S.C., Sec. 3542]. |
| Capability of a product to provide a stated function if demanded, under given conditions over its defined lifetime [ISO 26262-1]. |
| Confidentiality | Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information [NIST SP 800-137, 44 U.S.C., Sec. 3542]. |
| Impact | Value of damage possibly sustained via an attack. |
| Integrity | Guarding against improper information modification or destruction and includes ensuring information non-repudiation and authenticity [NIST SP 800-137, 44 U.S.C., Sec. 3542]. |
| Privacy | The ability to provide protection against personal data discovery and misuse of that information by other users [Common Criteria Part 2]. |
| Safety | The design, implementation, operation and maintenance of the system and associated processes shall not jeopardize the health and safety of individuals, the environment or any associated assets. |
| Absence of unreasonable risk due to hazards caused by malfunctioning behavior of E/E systems [ISO 26262-1]. |
| Tampering | The ability to change data in transit or in a data store. |
| Threat | A threat is an undesired event. A potential occurrence often best described as an effect that might damage or compromise an asset or objective. It is relative to each site, industry, company and is more difficult to uniformly define. |
| Cockpit | Cockpit is a web-based interface designed for managing Linux systems, enabling system administrators to monitor and control their systems more easily. |
| PAM | Pluggable Authentication Modules |
| Vulnerability | Part of the information security infrastructure that could represent a weakness to attack in the absence of a control. |

# **1. Schedule**

Below is our schedule for phase 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | week 1 | | | | | week 2 | | | | | week 3 | | | | |
|  | M | T | W | T | F | M | T | W | T | F | M | T | W | T | F |
| Analyze the requirements |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Check install and operation of the offered source |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Requirement engineering |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Asset identify |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Threat modeling & Risk assessment |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Elicit security requirement & Mitigation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SW Architecture |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Check the open source to implement |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implement Client |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implement Server |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implement secure communication |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Integration Client and Server |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# **2. System Requirement**

We have analyzed the requirements document and refined the system requirements.

The requirements are marked as functional requirements and quality attributes and given priority.

## 2.1. Functional Requirement

|  |  |  |  |
| --- | --- | --- | --- |
| **REQ ID** | **Description** | **Status** | **Mitigation** |
| FREQ\_1\_1 | The ability to set a username and password for a user with the Remote User Interface application. | Completed | 7.3.2. PAM (Pluggable Authentication Modules)  7.3.4. Management of User account and password by cockpit |
| FREQ \_1\_2 | The system shall ensure that the user’s password is secure. Passwords must be a minimum of 10 characters long and include one number and one symbol. | Completed | 7.3.2. PAM (Pluggable Authentication Modules)  7.3.4. Management of User account and password by cockpit |
| FREQ \_1\_3 | The system shall ensure that the username is unique and does not conflict with another username. | Completed | 7.3.2. PAM (Pluggable Authentication Modules)  7.3.4. Management of User account and password by cockpit |
| FREQ \_1\_4 | The system should force an administrator to periodically reset all passwords (at least once a month). | Completed | 7.3.2. PAM (Pluggable Authentication Modules)  7.3.4. Management of User account and password by cockpit |
| FREQ \_1\_5 | If a user enters the incorrect password more than three times, then their account will be locked for one hour. | Completed | 7.3.2. PAM (Pluggable Authentication Modules)  7.3.4. Management of User account and password by cockpit |
| FREQ \_1\_6 | The system shall provide the ability for an administrator to recover or change passwords. | Completed | 7.3.2. PAM (Pluggable Authentication Modules)  7.3.4. Management of User account and password by cockpit |
| FREQ \_2\_1 | The Remote User Interface application shall provide the ability to initiate a TCP server-client connection with the robot. | Implemented in base code |  |
| FREQ \_2\_2 | The connection shall provide the remote user interface the ability to give commands to the robot. | Implemented in base code |  |
| FREQ \_2\_3 | The connection shall provide the remote user interface with the ability to receive information from the robot. | Implemented in base code |  |
| FREQ \_2\_4 | (Bonus) The connection shall provide the remote user interface with the ability to modify the network connection settings with the Raspberry Pi and the Raspberry Pi username and password settings. | Completed | 7.3.2. PAM (Pluggable Authentication Modules)  7.3.4. Management of User account and password by cockpit |
| FREQ \_3 | The Remote User Interface application shall provide a graphical user interface with the following functionality: | Implemented in base code |  |
| FREQ \_3\_1 | Buttons / Text Input to log into the Remote User Interface. | Implemented in base code |  |
| FREQ \_3\_2 | Buttons / Text Input to connect robot to Remote User Interface via a wireless network connection | Implemented in base code |  |
| FREQ \_3\_3 | A window to display the camera feed from the barbette system. | Implemented in base code |  |
| FREQ \_3\_4 | Buttons to toggle between system modes. | Implemented in base code |  |
| FREQ \_3\_5 | Buttons to toggle laser on and off. | Implemented in base code |  |
| FREQ \_3\_6 | Buttons to toggle camera feed on and off (Bonus feature). | Completed | Code modification |
| FREQ \_3\_7 | Text display of robot’s current actions (e.g., searching for target) | Implemented in base code |  |
| FREQ \_3\_8 | Text display if an unsafe condition is present. | Completed | Code modification |
| FREQ \_3\_9 | Text display if a fault condition is present. | Completed | Code modification |
| FREQ \_3\_10 | Text display if a network connection is lost. | Completed | Code modification |
| FREQ \_3\_11 | Text display if connection with the robot is lost. | Completed | Code modification |
| FREQ \_4\_1 | The Remote User Interface application shall allow the user to transition between operational modes (Unknown, Safe, Pre-Arm, Armed Manual, Auto Engage). | Implemented in base code |  |
| FREQ \_4\_2 | Alert users of normal operating actions, fault conditions, and unsafe conditions. | Implemented in base code  (There is just text.) |  |
| FREQ \_4\_3 | Allow the user to perform basic corrective safety or operational actions through the graphical user. | Implemented in base code |  |
| FREQ \_5 | The Remote User Interface application shall provide an audit log of commands entered/issued to the robot. | Completed | 7.3.5. Audit logging |
| FREQ \_6 | The Remote User Interface application shall provide an audit log of information provided by the robot. | Completed | 7.3.5. Audit logging |
| FREQ\_7\_0 | If the connection is lost between the remote user interface and the robot, the robot shall suspend auto engage mode operations, stop robot firing of the gel water cannon, turn off the laser, and enter a safe mode. | Completed | Code modification |
| FREQ \_7\_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 and the operator shall not be permitted to perform any operations. | Completed | Code modification |
| FREQ \_7\_2 | If the network connection to the remote interface is lost, the robot will automatically stop actions until a network connection recovers and is commanded by the remote operators to resume in a manual or automatic mode. | Completed | Code modification |
| FREQ \_7\_3 | If the barbette traverses more than 15 degrees left or right of the center aim line, the robot will automatically stop actions until commanded by the remote operators to resume in a manual or automatic mode. | Completed | Code modification |
| FREQ \_7\_4 | If the barbette traverses more than 15 degrees up or down in elevation of the center aim line, the robot will automatically stop actions until commanded by the remote operators to resume in a manual or automatic mode. | Completed | Code modification |
| FREQ \_7\_5 | If the cannon continues to fire, either repeatedly on a single target or while not aimed at a target, the robot will automatically stop actions until commanded by the remote operators to resume in a manual or automatic mode. | Completed | Code modification |
| FREQ \_7\_6 | If the system is in an Unknown, Safe, or Pre-Arm state, the robot shall not fire the cannon. | Implemented in base code |  |

Table 1. Functional Requirements

## 2.2. Non-Functional Requirement

|  |  |  |  |
| --- | --- | --- | --- |
| **REQ ID** | **Description** | **Status** | **Mitigation** |
| NFREQ\_01 | Performance: The system must monitor the robot in near real time. | Completed  - before: 3 fps  - after: 30 fps | Code modification |
| NFREQ \_02 | Safety: System must be able to override robot commands and enter a safe mode. | Implemented in base code |  |
| NFREQ \_03 | Communication privacy: The system must ensure that TCP communication remains private. No intermediary should be able to intercede, disrupt, or deny commands from the remote user interface to the robot. | Completed | 7.3.1. Secure communication |
| NFREQ \_04 | Proof of identity (nonrepudiation): Users and operators cannot deny the commands provided to the robot. | Completed | 7.3.1. Secure communication  7.3.3. Two factor Authentication  7.3.5. Audit logging |
| NFREQ \_05 | Reliability: The system ensures that when a network connection between the robot and the remote interface is not established or is lost, or if the network connection to the remote interface is lost, the remote user interface acts to regain connection and alerts the user. | Completed |  |

Table 2. Non Functional Requirements

# **3. Security Goals**

1. Encrypts Sensitive Information

2. Provides Authentication

3. Provides integrity of sensitive data

# **4. Assets**

We define the below items to assets that are protected.

|  |
| --- |
| **Assets** |
| User ID/password |
| Control data |
| Image data |
| Client/Server connection |
| Pre-arm code |
| Response data |
| Keys for security features |

Table 3. Assets

# **5. Threat Modeling**

We executed the threat modeling to identify the threats of the system. We used PnG and the Microsoft threat modeling tool for STRIDE.

## 5.1. PnG

\*PnG pre-assumption: A gel blaster is not a toy but a real weapon system, and our team is a weapon corporation competing with other corporations.

1st PnG

-PnG Type: Competitor who wants to make a contract with a national military

-Motivation of PnG: to win the bid competition and disqualify our team

-Goal of PnG: get control over our weapon system and cause a malfunction

-Skills of PnG:

1) Background knowledge of weapon system network protocol

2) Ability to perform DDoS attacks

2nd PnG

-PnG Type: External hacker units from hostile groups/countries

-Motivation of PnG: threaten the weapon’s owner and demand money

-Goal of PnG: override the weapon system with an intention to damage assets

-Skills of PnG:

1) Unauthorized data access

2) Create undetectable malware

3rd PnG

-PnG Type: Internal employee working for our corporation

-Motivation of PnG: unsatisfied with his/her low salary

-Goal of PnG: inject malicious codes into the system and create a backdoor to the system

-Skills of PnG:

1) Inside information

2) Ability to authorize access

## 5.2. DFD & STRIDE

The following is the current system's DFD (data flow diagram) using Microsoft threat modeling tool.

We didn’t add storage for user IDs/passwords and log data in DFD because the current system doesn’t have it.

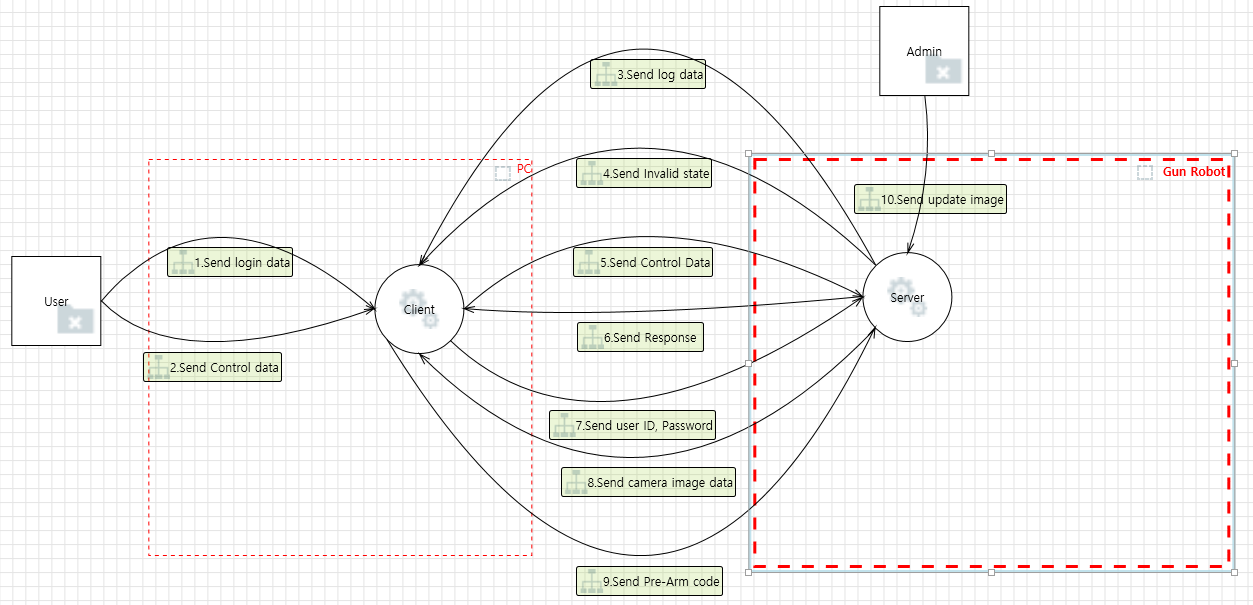


Figure 1. Data flow diagram

We found some threats in the current system by using STRIDE.

Some of the threats do not apply to our project. They are not included in the following tables.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Threat  Id | Title | Category | Interaction | Description | Security Requirement |
| 15 | Spoofing the Client Process | Spoofing | 5.Send Control Data | The attacker intercepts control data transmitted to the Server through a MITM attack and modifies the control data. | Standard authentication mechanism for identification  - Authentication is required as malicious attackers can transmit fake control data. |
| 16 | Spoofing the Server Process | Spoofing | 5.Send Control Data | Server may be spoofed by an attacker and this may lead to information disclosure by Client. | Standard authentication mechanism for identification  - Authentication is required as malicious attackers can transmit fake control data. |
| 17 | Potential Lack of Input Validation for Server | Tampering | 5.Send Control Data | This may lead to a denial of service attack against Server or an elevation of privilege attack against Server or an information disclosure by Server. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. | Verify and Authenticate data by implementing Secure communication.    Data flowing shall be encrypted by secure communication. |
| 18 | Potential Data Repudiation by Server | Repudiation | 5.Send Control Data | Server claims that it did not receive data from a source outside the trust boundary.  - In case an attacker takes control of Server, reliability of the storage of the logging data according to the authority | Standard authentication mechanism for identification is required.  Collect and store logging data sent to Server  Collect and store logging data sent from Client |
| 19 | Data Flow Sniffing | Information Disclosure | 5.Send Control Data | Data flowing may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. | Data flowing should be encrypted |
| 20 | Potential Process Crash or Stop for Server | Denial Of Service | 5.Send Control Data | Server crashes, halts, stops or runs slowly | The system should be available although unauthorized users send invalid packets. |
| 22 | Server May be Subject to Elevation of Privilege Using Remote Code Execution | Elevation Of Privilege | 5.Send Control Data | An attacker may pass data into Server in order to change the flow of program execution within Server.  Attacker sends malicious packet to make stack buffer overflow, and then attacker can get remote code execution | Secure compiler option should be applied.  Secure coding rules should be applied. |
| 23 | Elevation by Changing the Execution Flow in Server | Elevation Of Privilege | 5.Send Control Data | An attacker may pass data into Server in order to change the flow of program execution within Server to the attacker's choosing.  Attacker sends malicious packet to make stack buffer overflow, and then attacker can get remote code execution. | Authorization feature should be implemented in Server side. |
| 199 | Data Flow 5.Send Control Data Is Potentially Interrupted | Denial Of Service | 5.Send Control Data | Due to loss of availability, control data will not be accessible or available. It will cause operational damage | The system should be available although unauthorized users send invalid packets. |
| 201 | Elevation Using Impersonation | Elevation Of Privilege | 5.Send Control Data | Server may be able to impersonate the context of Client in order to gain additional privilege. | Authorization feature should be implemented in Server side. |
| 187 | Spoofing the Client Process | Spoofing | 7.Send user ID, passwords | The attacker intercepts control data transmitted to the Server through a MITM attack and modifies the control data. | Standard authentication mechanism for identification  - Authentication is required as malicious attackers can transmit fake control data. |
| 189 | Potential Lack of Input Validation for Server | Tampering | 7.Send user ID, passwords | User ID, Password may be tampered with by an attacker. Attacker can block normal user access. | - Data should be encrypted.  - Multi factor authentication is required. |
| 190 | Potential Data Repudiation by Server | Repudiation | 7.Send user ID, passwords | Server claims that it did not receive data from a source outside the trust boundary.  - In case an attacker takes control of Server, reliability of the storage of the logging data according to the authority | Standard authentication mechanism for identification is required.  Collect and store logging data sent to Server  Collect and store logging data sent from Client |
| 191 | Data Flow Sniffing | Information Disclosure | 7.Send user ID, passwords | Data flowing may be sniffed by an attacker. User ID, Password can be used brute force attack or dictionary attack. | Data flowing should be encrypted.  Passwords flowing should be hashed. |
| 192 | Potential Process Crash or Stop for Server | Denial Of Service | 7.Send user ID, passwords | Server crashes, halts, stops or runs slowly | The system should be available although unauthorized users send invalid packets. |
| 195 | Server May be Subject to Elevation of Privilege Using Remote Code Execution | Elevation Of Privilege | 7.Send user ID, passwords | An attacker may pass data into Server in order to change the flow of program execution within Server.  Attacker sends malicious packet to make stack buffer overflow, and then attacker can get remote code execution. | Secure compiler option should be applied.  Secure coding rules should be applied. |  |
| 200 | Spoofing the Server Process | Spoofing | 7.Send user ID, passwords | The attacker intercepts control data transmitted to the Server through a MITM attack and modifies the control data. | Standard authentication mechanism for identification  - Authentication is required as malicious attackers can transmit fake control data. |  |
| 206 | Data Flow Is Potentially Interrupted | Denial Of Service | 7.Send user ID, passwords | Server crashes, halts, stops or runs slowly. | The system should be available although unauthorized users send invalid packets. |  |
| 209 | Elevation by Changing the Execution Flow in Server | Elevation Of Privilege | 7.Send user ID, passwords | An attacker may pass data into Server in order to change the flow of program execution within Server.  Attacker sends malicious packet to make stack buffer overflow, and then attacker can get remote code execution. | Secure compiler option should be applied.  Secure coding rules should be applied. |  |
| 214 | Elevation Using Impersonation | Elevation Of Privilege | 7.Send user ID, passwords | Server may be able to impersonate the context of Client in order to gain additional privilege. | Access control should be applied. |  |
| 71 | Potential Lack of Input Validation for Client | Tampering | 8.Send camera image data | Camera image can be tampered. A set of fake image prepared by attacker. It can disguise normal to abnormal and abnormal to normal. May confuse users (not serious) | Servers and clients must be able to authenticate themselves. |  |
| 73 | Data Flow Sniffing | Information Disclosure | 8.Send camera image data | An attacker intercepts packets and obtains image information. | Data transmitted through communication channel should be encrypted. |  |
| 75 | Data Flow Is Potentially Interrupted | Denial Of Service | 8.Send camera image data | Continuous image data is sent by an attacker to cause malfunction. | The system must ensure availability even with excessive input. |  |
| 77 | Client May be Subject to Elevation of Privilege Using Remote Code Execution | Elevation Of Privilege | 8.Send camera image data | Remote code injection from server to client | Image data must not include vulnerabilities. |  |
| 178 | Potential Lack of Input Validation for Server | Tampering | 9.Send Pre-Arm code | Attacker can tamper the pre-arm code. Access can be blocked even if the correct code is entered. (availability) | Servers and clients must be able to authenticate themselves. |  |
| 180 | Data Flow Sniffing | Information Disclosure | 9.Send Pre-Arm code | An attacker steals unencrypted PRE-ARM code. | Data should be encrypted. |  |
| 184 | Server May be Subject to Elevation of Privilege Using Remote Code Execution | Elevation Of Privilege | 9.Send Pre-Arm code | It is possible to inject/execute shell code by exploiting vulnerabilities in the source code that receives and parses pre-arm code (if any). | Secure compiler option should be applied.  Secure coding rules should be applied. |  |
| 37 | Spoofing the User External Entity | Spoofing | 1.Send login data | User may be spoofed by an attacker and this may lead to unauthorized access to Client. Consider using a standard authentication mechanism to identify the external entity. | In normal operation mode, when unauthorized user accesses Client, Client should block the login 100% of the time. |  |
| 39 | Potential Data Repudiation by Client | Repudiation | 1.Send login data | Client claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data. | In normal operation mode, when client logins the client, Client should save the login information 100% of the time. |  |
| 40 | Data Flow Sniffing | Information Disclosure | 1.Send login data | Data flowing across 1.Send login data may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow. | In normal operation mode, when user logins the client, the password should be invisible 100% of the time. |  |
| 48 | Spoofing the User External Entity | Spoofing | 2.Send Control data | User may be spoofed by an attacker and this may lead to unauthorized access to Client. Consider using a standard authentication mechanism to identify the external entity. | In normal operation mode, when unauthorized user controls Client, Client should block the control 100% of the time. |  |
| 50 | Potential Data Repudiation by Client | Repudiation | 2.Send Control data | Client claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data. | In normal operation mode, when user controls client, client should save the control information 100% of the time. |  |
| 52 | Potential Process Crash or Stop for Client | Denial Of Service | 2.Send Control data | Client crashes, halts, stops or runs slowly; in all cases violating an availability metric. | In normal operation mode, when authorized user controls client, client should run following the user inputs 100% of the time. |  |
| 54 | Elevation Using Impersonation | Elevation Of Privilege | 2.Send Control data | Client may be able to impersonate the context of User in order to gain additional privilege. | In normal operation mode, when unauthorized user control client, client should block the input 100% of the time |  |
| 154 | Spoofing the Server Process | Spoofing | 3.Send log data | Spoofed server can send fake log data to client. | Data should be encrypted.  Message Authentication Code should be added. |  |
| 155 | Spoofing the Client Process | Spoofing | 3.Send log data | Spoofed client can intercept log data. | Data should be encrypted.  Message Authentication Code should be added. |  |
| 158 | Data Flow Sniffing | Information Disclosure | 3.Send log data | Log data can be leaked by MITM attack. | Data should be encrypted.  Message Authentication Code should be added. |  |
| 159 | Potential Process Crash or Stop for Client | Denial Of Service | 3.Send log data | Attacker can send big size packet to make client abnormal. | Client should check size of log data packet. |  |
| 162 | Client May be Subject to Elevation of Privilege Using Remote Code Execution | Elevation Of Privilege | 3.Send log data | Server can try execute malicious code by using vulnerabilities. | Critical vulnerabilities which can cause execute code shouldn't be exist. |  |
| 163 | Elevation by Changing the Execution Flow in Client | Elevation Of Privilege | 3.Send log data | Server can try change the flow of program execution. | Critical vulnerabilities which can cause execute code shouldn't be exist. |  |
| 117 | Spoofing the Server Process | Spoofing | 4.Send Invalid state | Spoofed server send invalid status message to client. it can lead client to abnormal condition. | Data should be encrypted.  Message Authentication Code should be added. |  |
| 118 | Spoofing the Client Process | Spoofing | 4.Send Invalid state | Spoofed client can intercept invalid state message from server. It can occur unsafe fire because of mismatch of state. | Data should be encrypted.  Message Authentication Code should be added. |  |
| 119 | Potential Lack of Input Validation for Client | Tampering | 4.Send Invalid state | Tampered message of invalid status can make unsafe firing because of mismatch of state. | Data should be encrypted.  Message Authentication Code should be added. |  |
| 122 | Potential Process Crash or Stop for Client | Denial Of Service | 4.Send Invalid state | Untrusted invalid state message can be cause denial of service. | Data should be encrypted.  Message Authentication Code should be added. |  |
| 125 | Client May be Subject to Elevation of Privilege Using Remote Code Execution | Elevation Of Privilege | 4.Send Invalid state | Server can try execute malicious code by using vulnerabilities. | Critical vulnerabilities which can cause execute code shouldn't be exist. |  |
| 126 | Elevation by Changing the Execution Flow in Client | Elevation Of Privilege | 4.Send Invalid state | Server can try execute malicious code by using vulnerabilities. | Critical vulnerabilities which can cause execute code shouldn't be exist. |  |
| 92 | Spoofing the Admin External Entity | Spoofing | 10.Send update image | The attacker can spoof the admin and update malicious firmware to the system | The image uploader should be verified before he/she can upload new image on the server. |  |
| 93 | Potential Lack of Input Validation for Server | Tampering | 10.Send update image | Attacker can tampering the original image with malicious image. | Server needs the method to verified if the image is valid or not (integrity and authority needs to be guaranteed). |  |
| 95 | Data Flow Sniffing | Information Disclosure | 10.Send update image | The attacker can leak image information then take and advance to investigate the filesystem | The data transmit in the communication need to be encrypted. |  |
| 97 | Data Flow image Is Potentially Interrupted. | Denial Of Service | 10.Send update image | The attacker take down the update service and admin cannot update new image. | - Have firewall to prevent well known DoS attack.  - It only can update after the image upload success fully (if not the corrupted image may overwrite old image and make system crash) |  |

Table 4. Threats from STRIDE

## 5.3. Traceability Matrix

Prioritized each threat through a team workshop.

|  |  |  |  |
| --- | --- | --- | --- |
| **Security Req. ID** | **Requirement /**  **Threat**  **description** | **Design rationale** | **Verification**  **method** |
| 1 | Spoofing: Standard authentication mechanism for identification  - Authentication is required as malicious attackers can transmit fake control data | 1. Authentication can be applied by Linux pam module. Linux pam can block unauthorized users.  2. Message Authentication Code should be added.  3. Keys for authentication should be protected. | Testing |
| 2 | Tampering: Verify and Authenticate data by implementing Secure communication – Data flowing shall be encrypted, hashed and verified by secure communication. | Data should be encrypted.  AES256 or more secure algorithm should be used. | Testing,  Review |
| 3 | Repudiation: Server claims that it did not receive data from a source outside the trust boundary.  - In case an attacker takes control of Server, reliability of the storage of the logging data according to the authority | 1. PKI-based server authentication  - CA certificate and server certificate must be issued for the Server process.  2. Server collects and stores logging data of Server.  Client collects and stores the transmitted data of client. | Testing |
| 4 | Information Disclosure: Data flowing should be encrypted | Data should be encrypted.  AES256 or more secure algorithm should be used. | Testing,  Review |
| 5 | Denial Of Service:  The system should be available although unauthorized users send invalid packets. | 1. Filtering packets should be implemented.  2. Secure private network should be implemented. | Testing |
| 6 | Elevation of Privilege: Critical vulnerabilities which can cause execute code shouldn't be exist. | 1.Secure compile option should be applied. ( Stack canary, ASLR, DEP, etc.)  2. Secure coding should be applied. | Review,  Static analysis |
| 7 | Tampering: Server needs the method to verified if the image is valid or not (integrity and authority needs to be guaranteed). | 1. Authentication for administrator should be implemented. | Testing,  Review |
| 8 | Spoofing: The firmware uploader should be verified before he/she can upload new firmware on the server. | 1. Authentication for administrator should be implemented. | Testing,  Review |

Table 5. Traceability Matrix

## 5.4. Security Requirements and Mitigations

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. ID** | **Requirement** | **Mitigation** | **Test case** |
| SREQ\_1 | Standard authentication mechanism for identification  - Authentication is required as malicious attackers can transmit fake control data | 7.3.1. Secure communication  7.3.3. Two factor Authentication | Table 10. Test Case 01  Table 11. Test Case 02 |
| SREQ\_2 | Verify and Authenticate data by implementing Secure communication – Data flowing shall be encrypted, hashed and verified by secure communication. | 7.3.1. Secure communication | Table 10. Test Case 01  Table 11. Test Case 02 |
| SREQ\_3 | Server claims that it did not receive data from a source outside the trust boundary.  - In case an attacker takes control of Server, reliability of the storage of the logging data according to the authority. | 7.3.1. Secure communication  7.3.6. Audit logging | Table 10. Test Case 01  Table 12. Test Case 03 |
| SREQ\_4 | Data flowing should be encrypted | 7.3.1. Secure communication | Table 13. Test Case 04 |
| SREQ\_5 | The system should be available although unauthorized users send invalid packets. | 7.3.5. Packet Filtering | Table 14. Test Case 05 |
| SREQ\_6 | Critical vulnerabilities which can cause execute code shouldn't be exist. | 7.4. Secure compile  7.5. Secure coding | Verification is done by Review and Static analysis tool. |
| SREQ\_7 | Server needs the method to verified if the image is valid or not (integrity and authority needs to be guaranteed). | 7.3.3. Two factor Authentication | Table 10. Test Case 01 |
| SREQ\_8 | The firmware uploader should be verified before he/she can upload new firmware on the server. | 7.3.3. Two factor Authentication | Table 10. Test Case 01 |
| SREQ\_9 | Keys for security features should be protected. (It should be stored in HSM or any secure storage) | Not implemented yet | Not implemented yet |

We assessed risks and elicited the following security requirements.

Table 6. Security Requirements and Mitigations

# **6. System Design**

## 6.1. Overall System Architecture

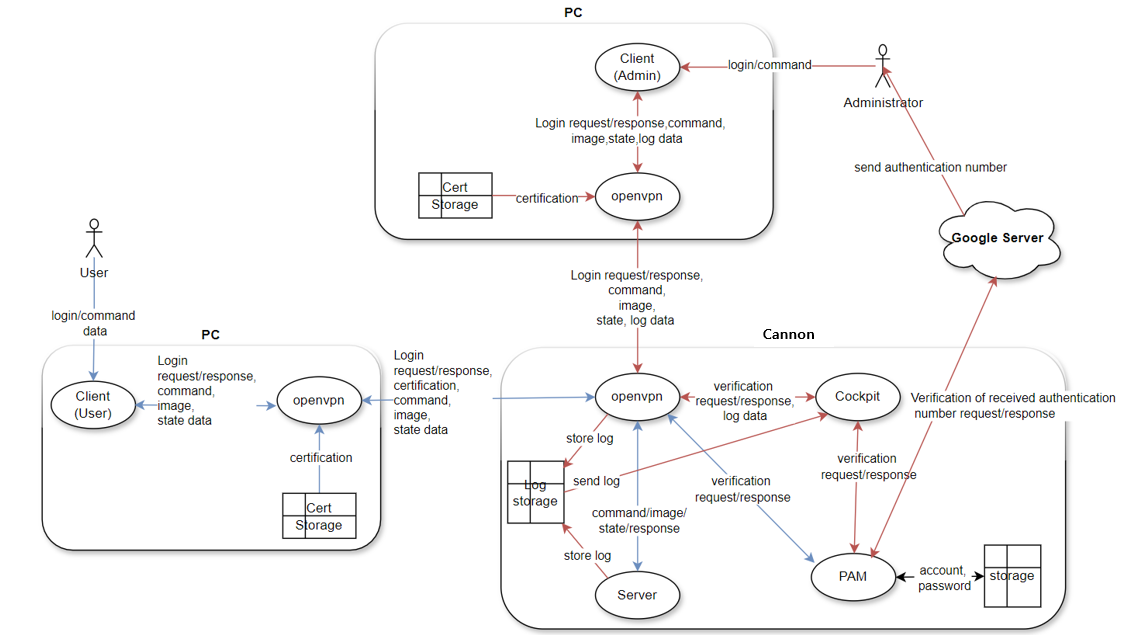


Figure 2. Overall system design

## 6.2. Used Open Source

### 6.2.1. Client

|  |  |  |  |
| --- | --- | --- | --- |
| **SW** | **version** | **The latest version** | **Description** |
| openvpn | 2.6.10 | 2.6.10 | Security fixes:  ​CVE-2024-27459  ​CVE-2024-24974  ​CVE-2024-27903  ​CVE-2024-1305 |
| opencv | 4.9.0 | 4.10.0 | There is no CVE issues on 4.9.0. |
| openssl | 3.2.1 | 3.3.1 | There are 5 issues related to 3.2.1, but they are not related to function in this project.  CVE-2019-0190  CVE-2009-3767  CVE-2009-3766  CVE-2009-3765  CVE-2009-1390 |

Table 7. Open sources of Client side

### 6.2.2. Server

|  |  |  |  |
| --- | --- | --- | --- |
| **SW** | **Version** | **The latest version** | **Description** |
| openvpn server | 2.6.10 | 2.6.10 | Security fixes:  ​CVE-2024-27459  ​CVE-2024-24974  ​CVE-2024-27903  ​CVE-2024-1305 |
| opencv | 4.9.0 | 4.10.0 | There is no CVE issues on 4.9.0. |
| libpam-modules (Pluggable Authentication Modules) | 1.5.2 | 1.6.1 | There are no CVE issues on 1.5.2. |
| openssl | 3.0.11 | 3.3.1 | There are 9 issues related to 3.0.11, but they are not related to function in this project.  CVE-2024-0727  CVE-2023-6129  CVE-2023-5678  CVE-2023-5363  CVE-2019-0190  CVE-2009-3767  CVE-2009-3766  CVE-2009-3765  CVE-2009-1390 |
| UFW | 0.36 | 0.36.2 | Frontend program for using iptables easily.  There is No CVE issues. |
| cockpit | 287 | 318 | Cockpit is a web-based graphical interface for servers.  There is no CVE issues.( Last CVE issue is for v260) |
| iptable | 1.8.9 | 1.8.10 | There is no CVE issues. |

Table 8. Open source of Server side

## 6.3. Design with mitigation

### 6.3.1. Secure communication

openvpn solution is applied for communication between client and server. openvpn supports TLS v1.3 (The latest version of TLS).

The reason for selecting openvpn is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **VPN protocols** | **Speed** | **Security** | **Ease of use** |
| OpenVPN | Moderate | High | High |
| WireGuard | High | High | Moderate |
| IKEv2/IPSec | Moderate | Moderate | High |
| L2TP/IPSec | Moderate | Moderate | Moderate |
| PPTP | High | Low | High |

Table 9. Review specification and performance

OpenVPN supports the highest encryption standard used in VPNs, which is 256-bit AES. In addition, its arsenal of security protocols relies on OpenSSL, a robust cryptographic toolkit of secure communication standards. In other words, OpenVPN is the most secure protocol.

WireGuard doesn’t support AES encryption, but it substitutes it with ChaCha20. One advantage of WireGuard is that it’s easier to audit and there’s a smaller attack surface compared to OpenVPN, since it’s implemented in a few lines of code. WireGuard is new and still developing.

IKEv2 and L2TP support different levels of AES encryption, and they use IPSec to handle the encryption itself. Their use of IPSec, however, has become a concern since the Edward Snowden leaks in 2013, since they seem to imply that the NSA is working to insert vulnerabilities to allow the agency to monitor VPN users.

PPTP is the least secure. PPTP has serious vulnerabilities, both in its use of challenge/response authentication protocol (CHAP) and the encryption standard it uses, which is MPPE. Basically, researchers have found that CHAP’s cryptography is easy to crack, and the quality of MPPE encryption is very low.

Key Features of openvpn:

-Powerful encryption, including AES-256 and Blowfish, ensuring top-level data security.

-Offers robust authentication mechanism

-Ability to navigate network address translation (NAT) gateways and firewall

-Allows for protocol flexibility, giving users the option to switch between UDP for speed and TCP for security

TLS v1.3 has removed weak algorithms such as RSA key exchange and used strong key exchange algorithms such as ECDHE by default.

Used Cipher suite: TLS\_AES\_256\_GCM\_SHA384

The following figure is the TLS v1.3 handshake sequence.

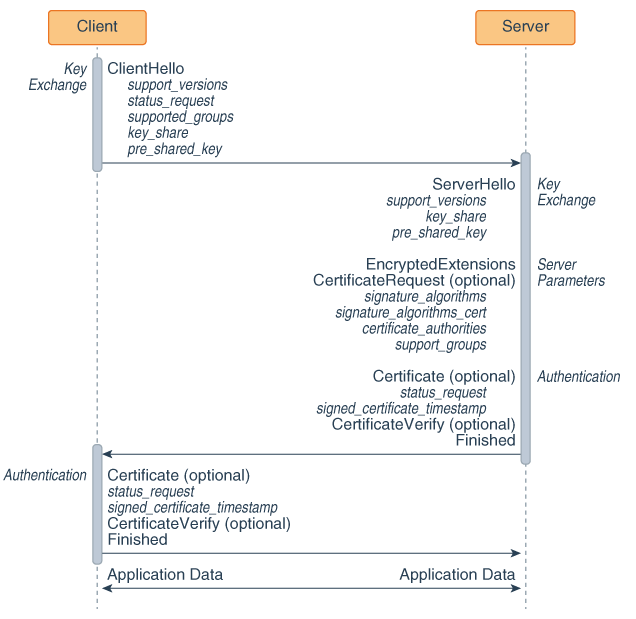


Figure 3. TLS v1.3 handshake sequence

(From <https://www.ibm.com/docs/en/sdk-java-technology/8?topic=handshake-tls-13-protocol> )

### 6.3.2. PAM (Pluggable Authentication Modules)

PAM is used for handling user authentication. It can limit the number of password attempts through PAM settings and lock the account after a certain number of incorrect attempts.

Key Features of PAM:

User Authentication: PAM can authenticate users in a variety of ways. For example, it can authenticate via a local password file or via a network service.

Account Management: PAM manages whether a user's account is valid, has not expired, and has appropriate access rights.

Password Management: PAM enforces password policies and handles password changes. For example, it can manage minimum length, complexity requirements, expiration cycles, etc.

SHA512 is used for hash algorithm for this project.

### 6.3.3. Two-factor Authentication

Two-factor authentication is implemented. We use the Google authenticator application and the administrator's user ID/password together. This feature uses cockpit and PAM.

Cockpit is a web-based interface designed for managing Linux systems, enabling system administrators to monitor and control their systems more easily.

Key Features of Cockpit:

Real-Time Monitoring: Monitor key performance metrics such as CPU, memory, and disk usage in real-time.

Log Management: View system and application logs, search and filter logs to track specific events.

Service Management: Check the status of running services and perform start, stop, and restart operations.

Network Management: Modify network interface settings and monitor network traffic.

Terminal Access: Execute commands directly from a terminal within the web interface without needing SSH access.

User Management: Add, delete, and modify system user accounts.

Sequence diagram of authentication for the administrator is like the following.

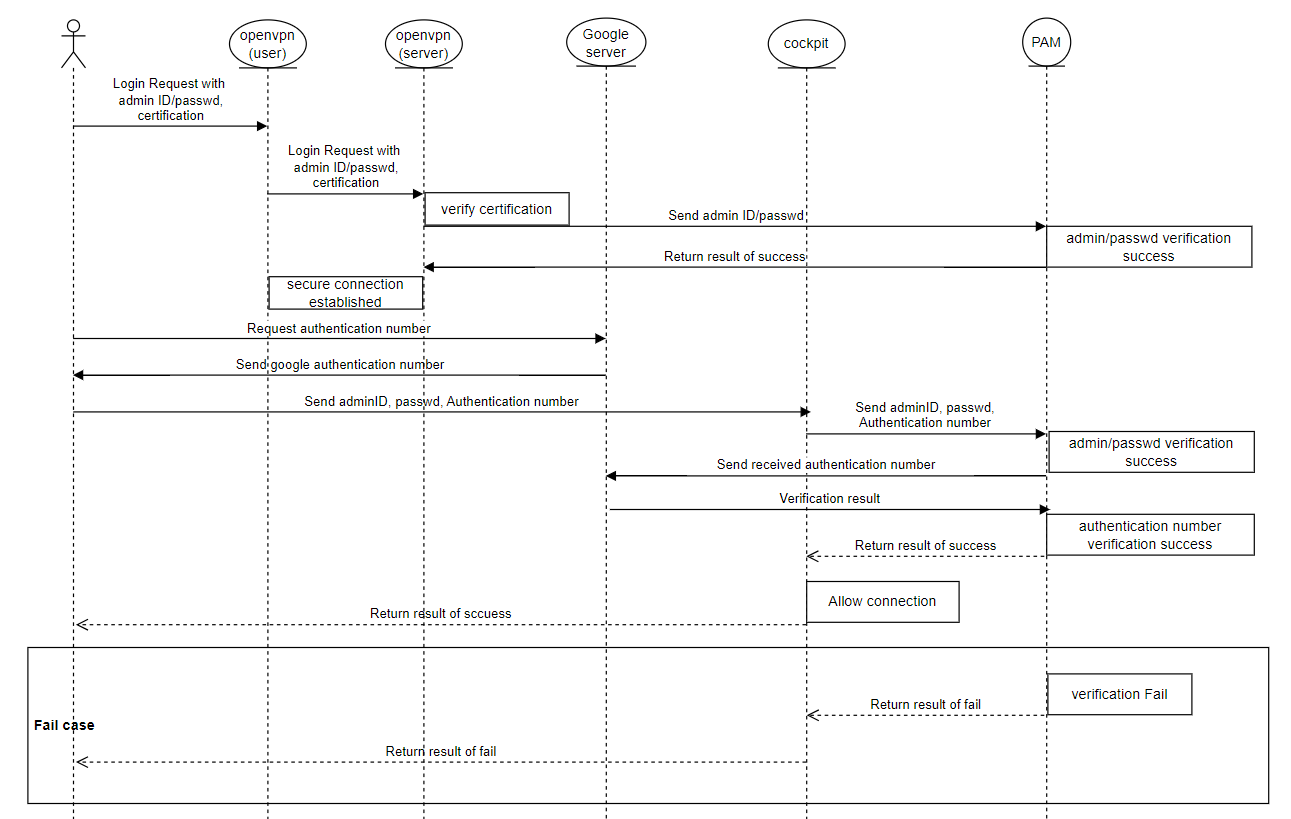


Figure 4. Sequence diagram of two factor authentication for administrator

Certification and user id/password are used for User authentication.

Sequence diagram is like the following.

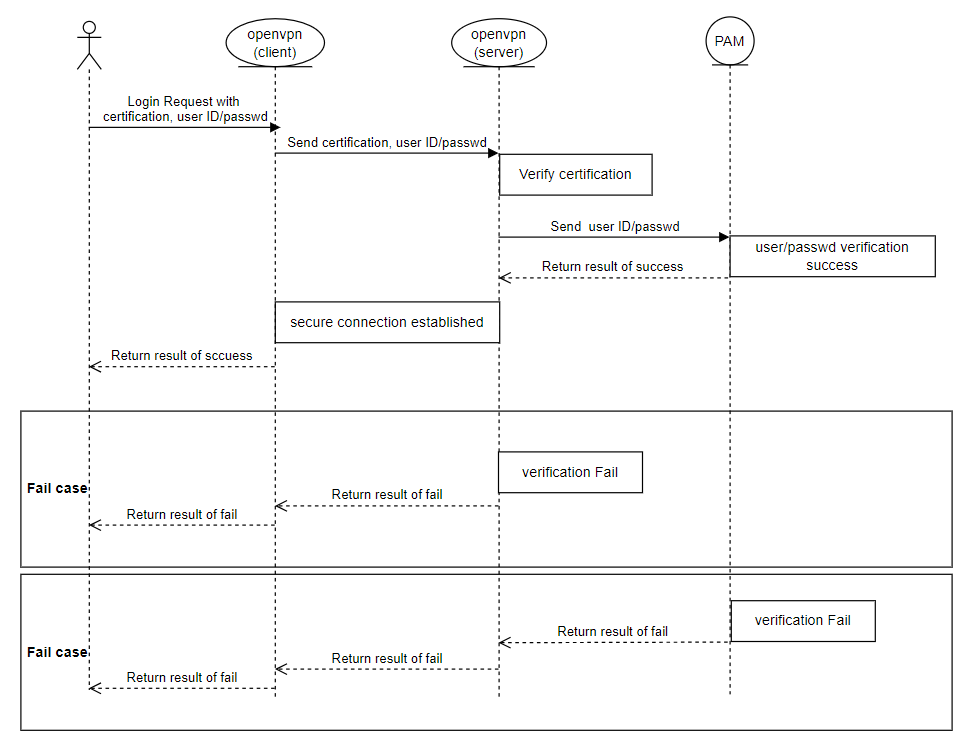


Figure 5. Sequence diagram of two factor authentication for user

### 6.3.4. Management of User account and password by cockpit

We use ‘cockpit’ to manage user accounts and passwords graphically. The cockpit displays a screen-like website on the Local PC.

The minimum length of a password is 10 characters (including one number and one symbol), and the maximum length is 127 characters.

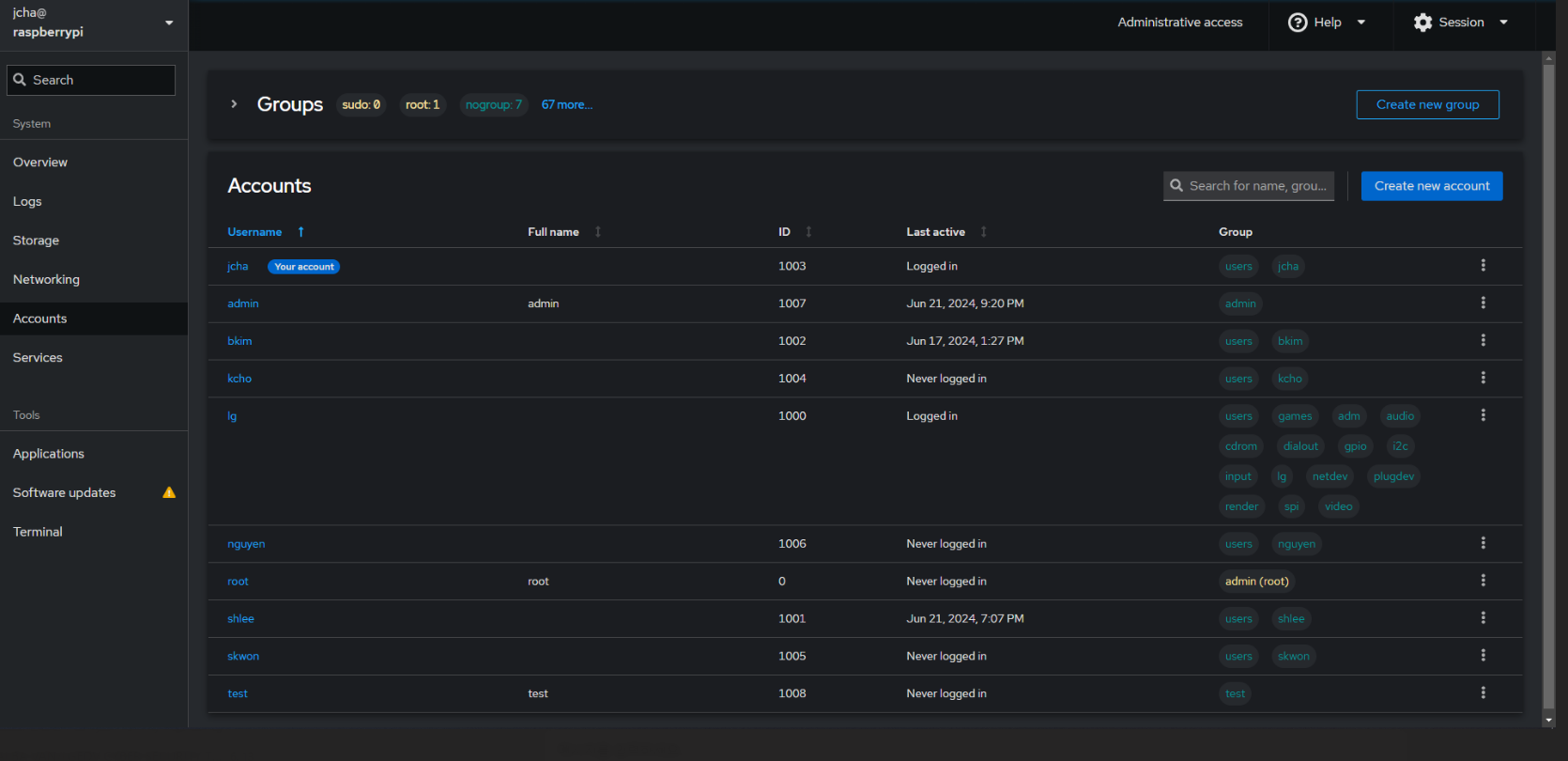


Figure 6. Screen of managing accounts and passwords

### 6.3.5. Packet Filtering

We implemented a typical Packet Filtering Firewall to provide the packet filtering and managing via the netfilter and iptables in the target. The packet-filtering is a mechanism with traffic screening rules enforcing local policies. It decides whether to route a packet through to its destination, to silently throw the packet away, or reject the packet and return an error message to the sending address based on the designed policies to sort packet header fields.

The firewall rule should have a default Packet Filtering policy. If a packet doesn’t match any rule, the default policy is applied.



Figure 7. Firewall Implementation

### 6.3.6. Audit logging

Cockpit is used for logging with the administrator account. It can show logs of command data from Client to the target.

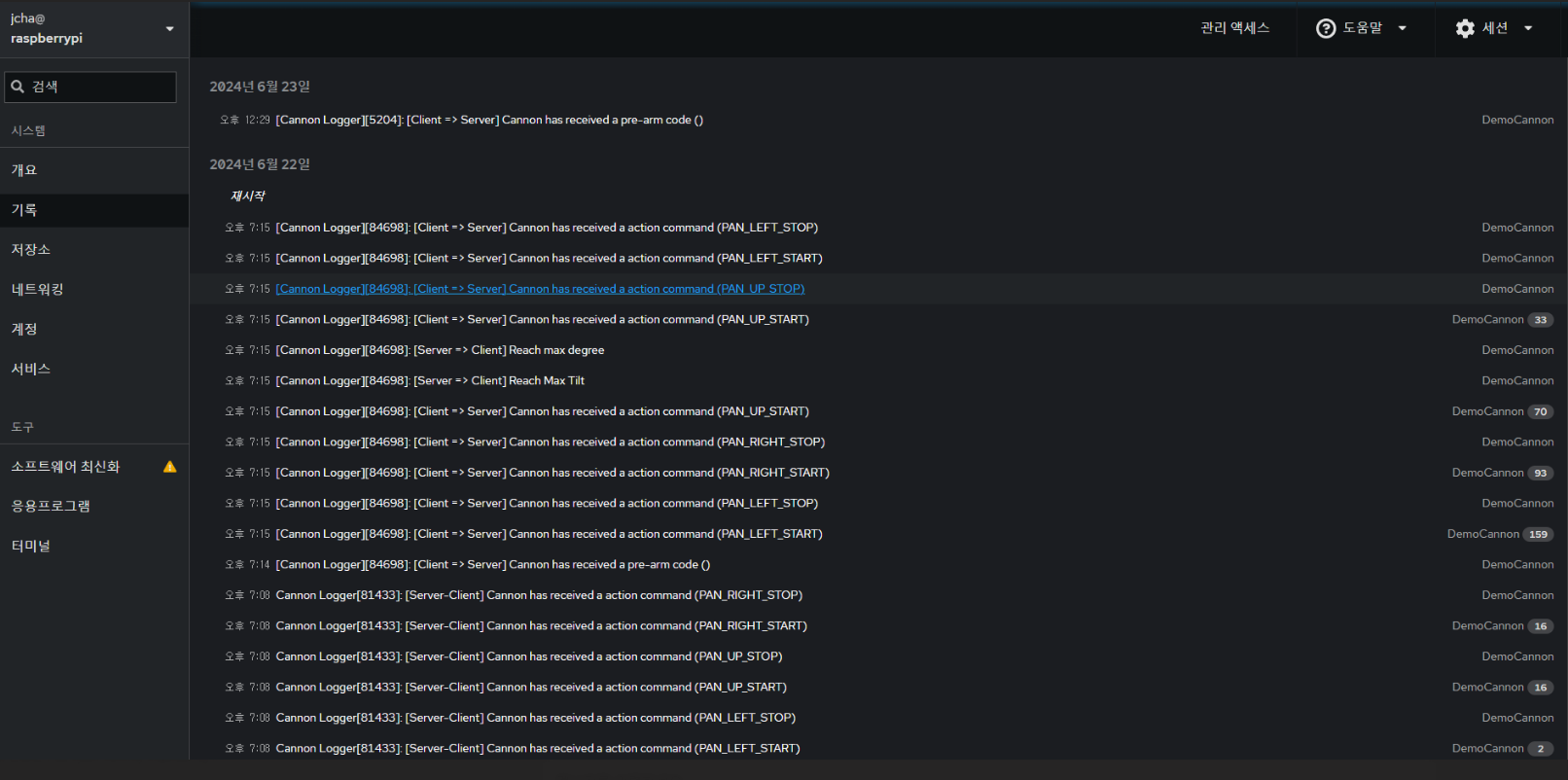


Figure 8. Audit Log with administrator account

## 6.4. Secure compile

### 6.4.1. Server

|  |  |
| --- | --- |
| **Compile/Linker option** | **Description** |
| -Wall | Enable warning in compile time |
| -s | Enable strip functions' name option |
| -fstack-protector-strong | Enable stack canary. Insert a special value (usually called a "canary") into the stack frame of the function and check that this value has not changed before the function returns. If the value has changed, the program is terminated as a security violation. |
| -Wl,-z,now | Enable FULL RELRO.  It helps prevent attacks that exploit memory-related vulnerabilities by setting certain memory areas to read-only while the program is running. RELRO stands for "Relocation Read-Only" and protects the relocation information of executables and shared libraries. |
| -fpie -fpic | Load executable at random memory addresses and prevent the kernel from relocating |
| -z noexecstack | NX disabled. It removes execution permissions from the stack memory area. This feature helps prevent attacks that exploit stack-based memory vulnerabilities by setting the stack memory area to not allow code execution when the program is running. |
| -D\_FORTIFY\_SOURCE=2 | String function mitigation. The compiler monitors the use of standard library functions and attempts to detect incorrect usage. For example, it checks to ensure that buffer sizes are not exceeded in functions such as strcpy, sprintf, etc. |

### 6.4.2. Client

|  |  |
| --- | --- |
| **Compile/Linker option** | **Description** |
| /GS | This option helps protect the application from buffer overflow attacks. It adds security checks to detect buffer overruns in your code. (canary) |
| /sdl | It enforces stricter checks and coding practices to help prevent common security vulnerabilities. |
| /guard | Ensures indirect calls are made only to valid targets. |
| /DYNAMICBASE | Randomizes the memory addresses of the executable. |
| /NXCOMPAT | Enforces Data Execution Prevention to prevent execution of code in data regions. |

## 6.5. Secure coding

SonarQube is used for secure coding analysis.

SonarQube is easy to use.

(Coverity requires a license fee and requires testing only products that the company will mass produce.)

We found the following number of issues and fixed them.

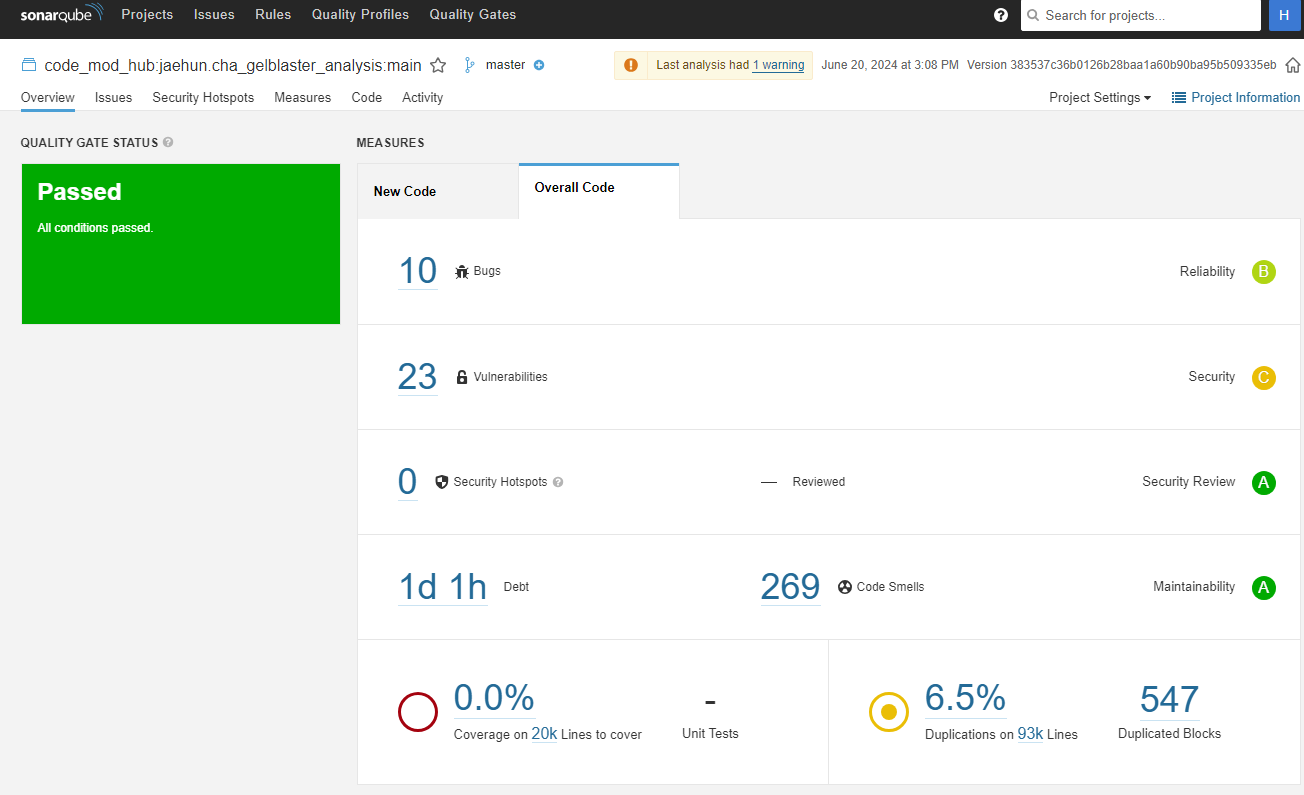


Figure 9. Result of secure coding analysis

Remained 2 Vulnerability issues:



Figure 10. Vulnerability issues

Reason for Not fixed: Port 5000 can be binding to any IP addresses, but it can be blocked by firewall policy.

# **7. Verification**

## 7.1. Overview of verification

To verify security requirements, we created Test Cases.

(We couldn’t make whole test cases for all Functional and Non-functional requirements because of short project schedule.)

## 7.2. Test Case Design for Security Requirements

## 7.3. Verification Result

# **8. Guide**

## 8.1. Setup Guide

### 8.1.1. Local PC

To install Certification for openvpn, Drivers, DLL on Local PC

- Go to 'prerequisites' directory in source code, and run setup.bat

Authentication of User

- User ID: teamx

- password: teamx123!@#

### 8.1.2. Setup for two-factor authentication of Administrator

1. Install Google Authentication application on your mobile device.

2. Execute Google Authentication application, and select '+' button (Bottom right of the screen).

3. Select 'QR code scan' menu.

4. Scanning QR code which Team3 shared with README.txt

5. New account will be added in your Google application by QR code.

6. Make sure Router connect to internet. (It is for connecting between the Google server and the target.)

7. You can see the following screen when you connect to the target.

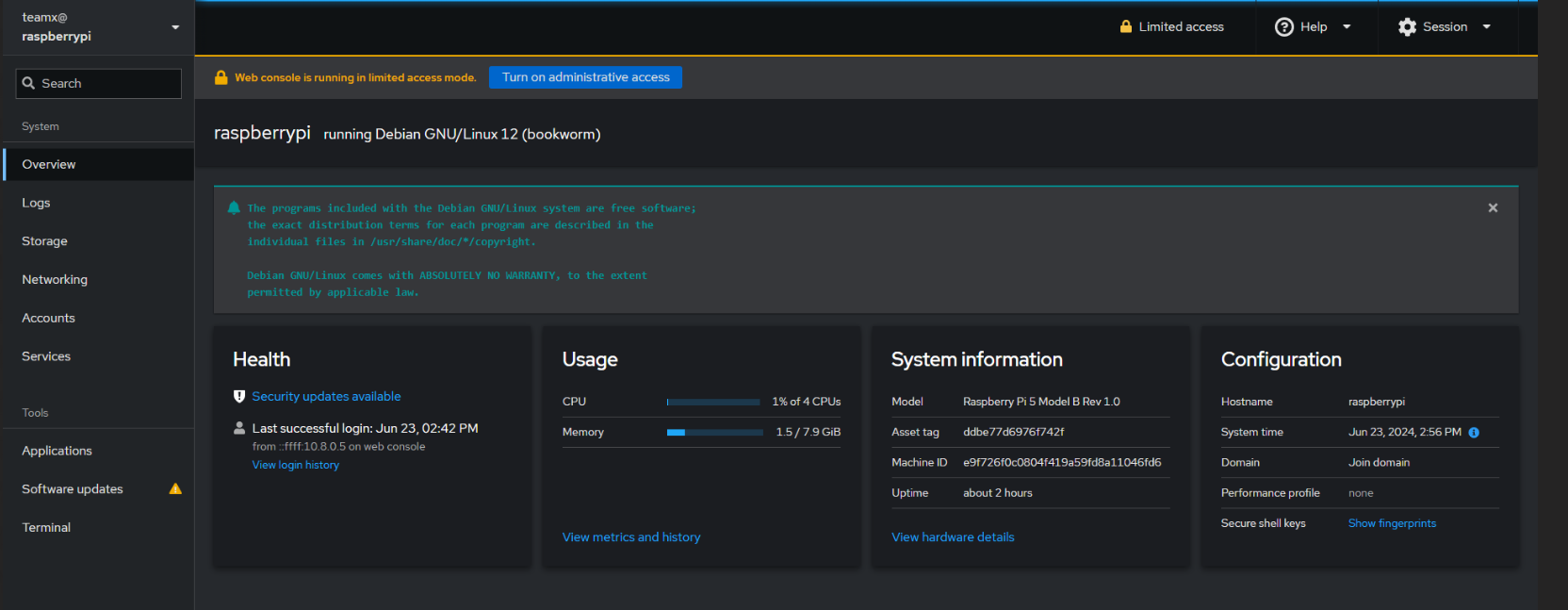


Figure 11. S

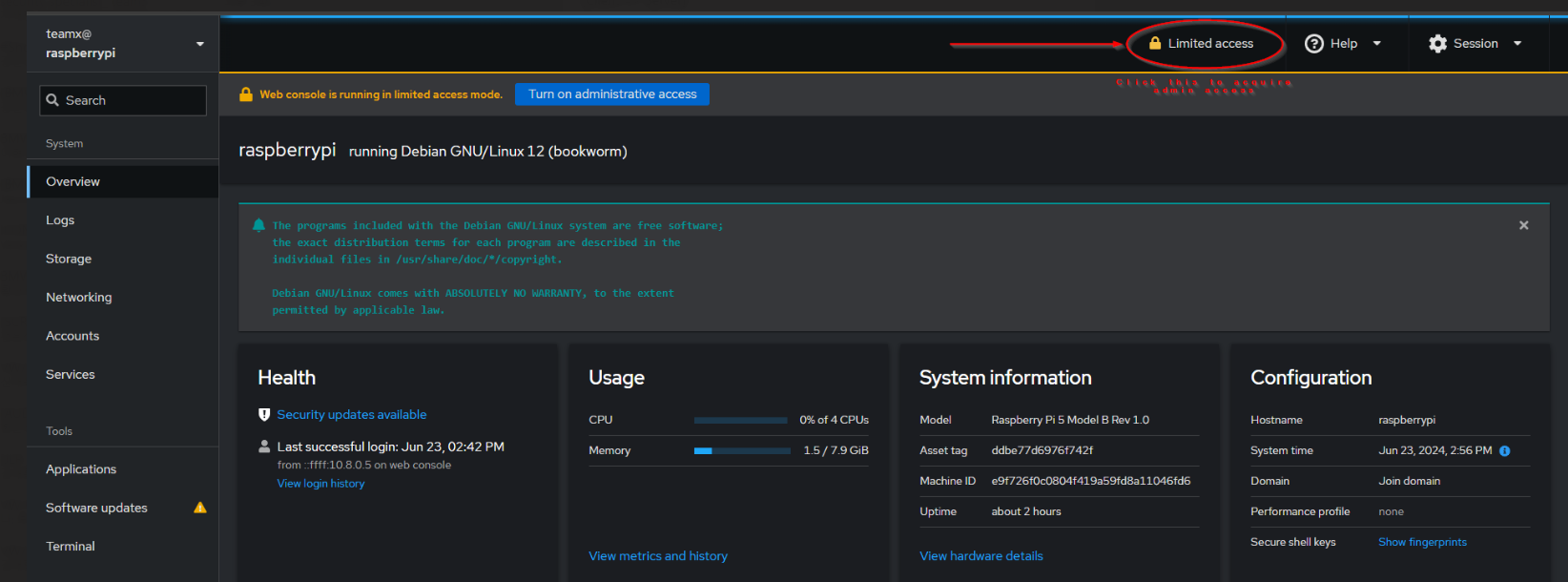


그림 12

### 8.1.3. Build guide for Client (User)

Use MS Visual Studio IDE to build the client application.