Automatic License Plate Recognition System

(Team5)

Table of Contents

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

[Role and Responsibilities](#_q2vzus4n5sz5) 4

[Terminology & Acronym](#_5uztr4zbxrnk) 5

[**1. Schedule**](#_30j0zll) **6**

[**2. System Requirement**](#_3znysh7) **7**

[2.1. Client](#_e5xphywgn9xo) 7

[2.2. Server](#_5697ltobxmlk) 8

[**3. Security Goals**](#_a495xknerc34) **8**

[**4. Assets**](#_tyjcwt) **8**

[**5. Threat Modeling**](#_3dy6vkm) **9**

[5.1. DFD & STRIDE](#_1t3h5sf) 9

[5.2. PnG](#_2s8eyo1) 19

[5.3. Brainstorming](#_17dp8vu) 20

[5.4. Result of Threat Modeling](#_3rdcrjn) 21

[**6. Security Risk Assessment**](#_bg09i17isnr0) **22**

[**7. Security Requirements**](#_lnxbz9) **25**

[**8. Mitigation**](#_bikomnlytmyf) **25**

[**9. Architecture**](#_1ksv4uv) **27**

[9.1. Overall SW Architecture](#_44sinio) 27

[9.2. Used Open Source](#_2jxsxqh) 27

[9.2.1. Client](#_t5tvrl7bfwnr) 27

[9.2.2. Server](#_9wxijq47qt0b) 27

[9.3. Crypto algorithms](#_gn1sbdkiva1f) 28

[9.3.1. TLS](#_hra1fx8h0t8) 28

[9.3.2. Encryption Data](#_s6f917puk6pw) 28

[9.3.3. JWT(JSON Web Token)](#_dlxfis75uun8) 28

[9.4. Source Directory](#_z337ya) 29

[**10. Implementation & Test Result**](#_jdwn08athwem) **29**

[10.1. Implementation](#_y8074r8yk1ba) 29

[10.2. Test Result](#_u7n94c4lfdri) 30

[**11. Guide**](#_eubogox3jt60) **32**

[11.1. Setup Guide](#_ey0tf56zxphi) 32

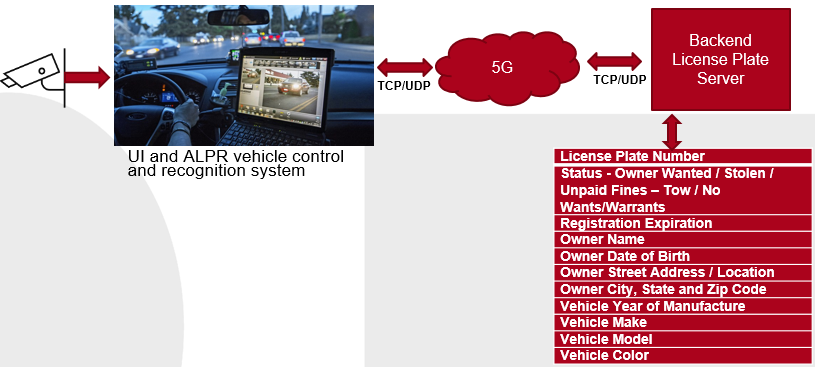
[11.1.1. Tools](#_5u698ekgffbb) 32

[11.1.2. Setup environment](#_us6ppoj1hg76) 32

[11.2. User Guide](#_2o3yjf68lnjp) 34

# **0. Introduction**

We will develop a secure implementation of the ALPR(Automatic License Plate Recognition) system that meets the given basic requirements.



## **Role and Responsibilities**

| **Name** | **Role and Responsibilities** |
| --- | --- |
| **Paul Lim** | - Team Leader  - Client App Development  - Client Security |
| **Jong Soo Oh** | - Architecture  - System/Security Requirement |
| **Jinhwan Kim** | - Threat Modeling  - Risk Assessment  - Validation |
| **Sangwook Lee** | - Server Security  - 2FA (OTP)  - DB, Mail management |
| **Dawoon Park** | - Server Development  - Server Security  - TLS, Certificate |
| **Minyong Ha** | - Server Security  - Authentication Design  - JWT, Cryptography |

## **Terminology & Acronym**

| **Terminology**  **& Acronym** | **Definitions** |
| --- | --- |
| ALPR | Automatic License Plate Recognition |
| CA | Certification Authority |
| CSR | Certificate Signing Request |
| DFD | Data-Flow Diagram |
| JPA | JAVA Persistence API |
| JWT | Json Web Token |

# **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 customer requirement |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Check install and operation of the offered source |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Define the system requirement |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Asset identify |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Threat modeling & Risk assessment |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Define secure requirement & Mitigation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SW Architecture |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Check the open source to implement |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implement Client |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implement Server |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Make test case |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Integration Client and Server |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Presentation phase 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# **2. System Requirement**

We have analyzed the requirement documents(2022 LG Security Project Description v2.docx and LG May 2022 Lecture Secure Coding Project Intro V1.0.pptx) and refined the system requirement.

The requirements are marked to functional requirements and quality attributes and given the priority. These are discussed and agreed with the Mentor.

## 2.1. Client

| **REQ ID** | **Description** | **CMU REQ ID** | **F/Q** | **Priority** |
| --- | --- | --- | --- | --- |
| REQ\_01 | The system shall allow an officer to login. | Client 1 | F | 6 |
| REQ\_02 | The system shall authenticate users locally and to the backend license plate database lookup | Client 1 | F | 6 |
| REQ\_03 | The system must use two factor authentication for sign on | Client 1 | Q  (Security) | 6 |
| REQ\_04 | user credentials must be protected | Client 1 | Q  (Security) | 6 |
| REQ\_05 | Lost or compromised credentials must be handled in a reasonable way | Client 2 | F | 1 |
| REQ\_06 | The system should allow the officer to choose between using a live camera and playback file in the UI | Client 12 | F | 6 |
| REQ\_07 | The system should provide an area in the user interface that always contains the current camera /playback view | Client 10 | F | 6 |
| REQ\_08 | The system should read images from the vehicle camera or a playback file and identify license plates for evaluation | Client 6 | F | 6 |
| REQ\_09 | The system should allow a law enforcement officer to select and save retrieved information locally | Client 3 | F | 3 |
| REQ\_010 | The system should allow a law enforcement officer to send retrieved information to a mobile device, such as a mobile phone to use in the field | Client 4 | F | 2 |
| REQ\_011 | The system should perform the ALPR function in real-time while maintaining a frame rate of at least 25fps | Client 7 | Q  (Performance) | 3 |
| REQ\_012 | The system should query the backend license plate server for details about the vehicle | Client 8 | F | 6 |
| REQ\_013 | The user must be alerted for vehicles that are stolen, the owner is wanted (criminal), or if it is a vehicle of interest (expired registration, unpaid tickets, owner is missing). Alerts must contain reason and vehicle make, model and color along with the isolated plate image and the recognized license plate number for operator comparison. | Client 8 | F | 6 |
| REQ\_014 | If a license plate does not generate an alert, then the user interface must display the last recognized plate image, the recognized license plate number and vehicle make, model and color so the operator can visually check if the plate matches the vehicle if desired | Client 9 | F | 1 |
| REQ\_015 | The system should allow officers to display computed camera / playback frames per second, average time per frame, jitter and frame number | Client 11 | F | 5 |
| REQ\_016 | The ability to detect network connectivity issues with the backend server within 5 seconds and automatically resolve the communication issue if possible | Client 13 | Q  (Availability) | 3 |
| REQ\_017 | The system should alert officers of any communication errors or failures | Client 14 | F | 2 |
| REQ\_018 | The system must fetch vehicle information in no more than 10 seconds as officers are often making queries in real time. | Client 15 | F/Q  (Performance) | 3 |

## 2.2. Server

| **REQ ID** | **Description** | **CMU REQ ID** | **F/Q** | **Priority** |
| --- | --- | --- | --- | --- |
| REQ\_01 | The system shall send the matched vehicle information when the server receive the query from client | Client 1 | F | 6 |
| REQ\_02 | Support multiple users | Client 1 | F | 6 |
| REQ\_03 | Ensure secure communication with the client applications | Client 1 | Q  (Security) | 6 |
| REQ\_04 | Authenticate remote laptop users | Client 1 | Q  (Security) | 6 |
| REQ\_05 | Return plate if there is not an exact match that includes a configurable minimum confidence threshold to support a partial match | Client 2 | F | 1 |
| REQ\_06 | Support configurable values via a configuration file | Client 12 | F | 6 |
| REQ\_07 | Track the average number of queries per second for each user and overall queries per second, for all users | Client 10 | F | 6 |
| REQ\_08 | Track the number partial matches and no matches for each user and all users | Client 6 | F | 6 |

# **3. Security Goals**

1. Client and server should be ***TRUSTED*** each other
2. Data exchanged between the server and the client should be ***PROTECTED***
3. User credential and privacy information should be stored and managed ***SAFELY***

# **4. Assets**

We define the below items to assets that are protected.

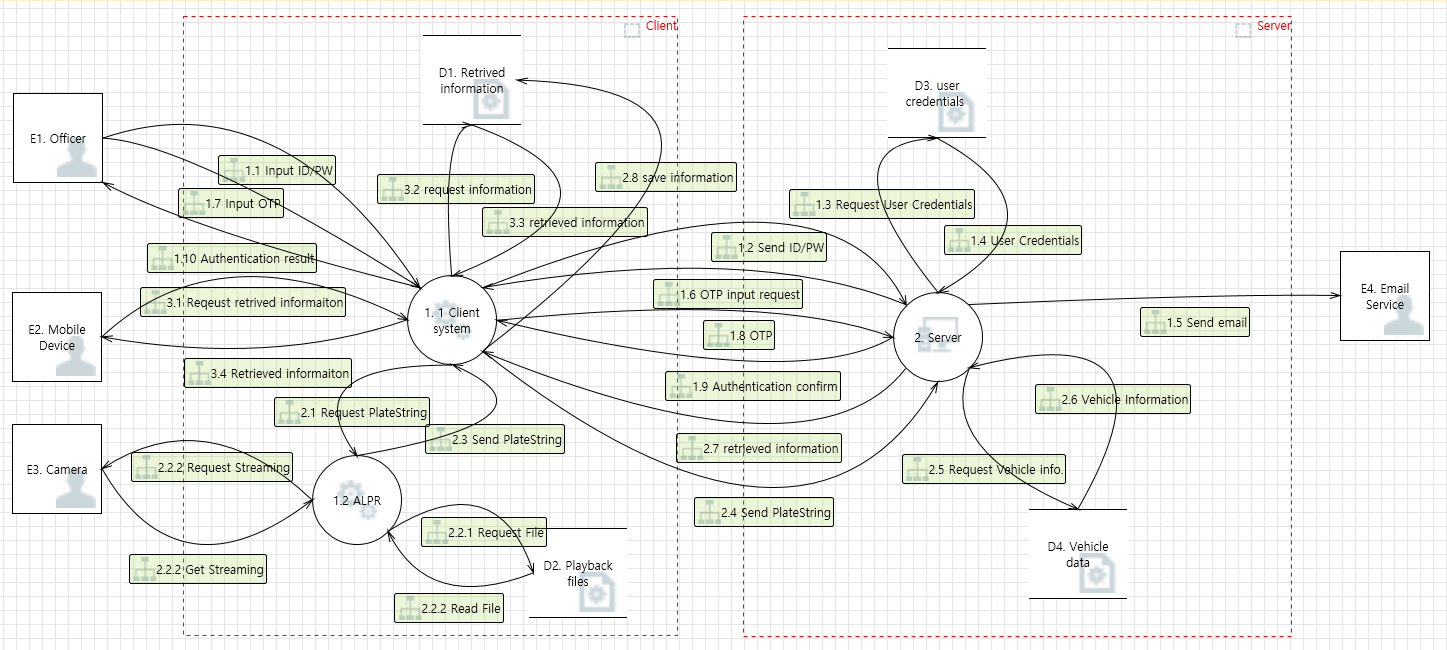
| **Assets** |
| --- |
| User credentials |
| Plate number |
| Vehicle information data |
| Client/Server connection |

# **5. Threat Modeling**

We did the threat modeling to identify the threats of the system. We used the Microsoft threat modeling tool for DFD(Data-Flow Diagram) and STRIDE. Moreover PnG and brainstorming are used to find the undetected threats by tool.

## 5.1. DFD & STRIDE

Below is a data-flow diagram of the system using Microsoft threat modeling tool.



We found some threats for our system during analyzing the threat list by tools.

But other threats are not applicable to our project. They are marked gray.

| **Id** | **Title** | **Category** | **Interaction** | **Description** | **Threat Analysis** |
| --- | --- | --- | --- | --- | --- |
| 41 | Spoofing the 1. 1 Client system Process | Spoofing | 1.2 Send ID/PW | 1. 1 Client system may be spoofed by an attacker and this may lead to unauthorized access to 2. Server. Consider using a standard authentication mechanism to identify the source process. | Client system may be spoofed. Server does not work because a fake client sends the ID/PW. |
| 42 | Spoofing the 2. Server Process | Spoofing | 1.2 Send ID/PW | 2. Server may be spoofed by an attacker and this may lead to information disclosure by 1. 1 Client system. Consider using a standard authentication mechanism to identify the destination process. | Server may be spoofed. Attackers could steal user id/pw. |
| 43 | Potential Lack of Input Validation for 2. Server | Tampering | 1.2 Send ID/PW | Data flowing across 1.2 Send ID/PW may be tampered with by an attacker. This may lead to a denial of service attack against 2. Server or an elevation of privilege attack against 2. Server or an information disclosure by 2. Server. 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. | Clients could not access the Server because ID/PW are tampered. Can't use the system.  Can't use the system because fake client send the ID/PW  Attackers insert the exploitable data in ID/PW message. it leads to abnormal behavior. |
| 44 | Cross Site Scripting | Tampering | 1.2 Send ID/PW | The web server '2. Server' could be subject to a cross-site scripting attack because it does not sanitize untrusted input. | Not applicable, Client is not a web application. |
| 45 | Potential Data Repudiation by 2. Server | Repudiation | 1.2 Send ID/PW | 2. Server 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. | Server repudiates even if the client sends the ID/PW to access the server. |
| 46 | Data Flow Sniffing | Information Disclosure | 1.2 Send ID/PW | Data flowing across 1.2 Send ID/PW 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. | Attackers could steal user id/pw. |
| 47 | Potential Process Crash or Stop for 2. Server | Denial Of Service | 1.2 Send ID/PW | 2. Server crashes, halts, stops or runs slowly; in all cases violating an availability metric. | Server does not work according to the attacker's input. |
| 48 | Data Flow 1.2 Send ID/PW Is Potentially Interrupted | Denial Of Service | 1.2 Send ID/PW | An external agent interrupts data flowing across a trust boundary in either direction. | Attacket interrupts ID/PW flow. Server does not work. |
| 49 | Elevation Using Impersonation | Elevation Of Privilege | 1.2 Send ID/PW | 2. Server may be able to impersonate the context of 1. 1 Client system in order to gain additional privilege. | Not applicable to our project  Because there is no separation of privileges in our project |
| 50 | 2. Server May be Subject to Elevation of Privilege Using Remote Code Execution | Elevation Of Privilege | 1.2 Send ID/PW | 1. 1 Client system may be able to remotely execute code for 2. Server. | Unintended behavior on the server |
| 51 | Elevation by Changing the Execution Flow in 2. Server | Elevation Of Privilege | 1.2 Send ID/PW | An attacker may pass data into 2. Server in order to change the flow of program execution within 2. Server to the attacker's choosing. | Unintended behavior on the server |
| 52 | Cross Site Request Forgery | Elevation Of Privilege | 1.2 Send ID/PW | Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker forces a user's browser to make a forged request to a vulnerable site by exploiting an existing trust relationship between the browser and the vulnerable web site. In a simple scenario, a user is logged in to web site A using a cookie as a credential. The other browsers to web site B. Website B returns a page with a hidden form that posts to web site A. Since the browser will carry the user's cookie to web site A, web site B now can take any action on web site A, for example, adding an admin to an account. The attack can be used to exploit any requests that the browser automatically authenticates, e.g. by session cookie, integrated authentication, IP whitelisting. The attack can be carried out in many ways such as by luring the victim to a site under control of the attacker, getting the user to click a link in a phishing email, or hacking a reputable website that the victim will visit. The issue can only be resolved on the server side by requiring that all authenticated state-changing requests include an additional piece of secret payload (canary or CSRF token) which is known only to the legitimate web site and the browser and which is protected in transit through SSL/TLS. See the Forgery Protection property on the flow stencil for a list of mitigations. | Not applicable, the client is not the web application. |
| 53 | Spoofing of Destination Datastore D3. user credentials | Spoofing | 1.3 Request User Credentials | D3. user credentials may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of D3. user credentials . Consider using a standard authentication mechanism to identify the destination data store. | Not applicable, DB is safely stored in the trust zone |
| 54 | Potential Excessive Resource Consumption for 2. Server or D3. user credentials | Denial Of Service | 1.3 Request User Credentials | Does 2. Server or D3. User credentials take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout. | Not applicable, DB server is not separated |
| 66 | Spoofing of Source Data Store D3. user credentials | Spoofing | 1.4 User Credentials | D3. user credentials may be spoofed by an attacker and this may lead to incorrect data delivered to 2. Server. Consider using a standard authentication mechanism to identify the source data store. | Not applicable, DB server is not separated |
| 67 | Cross Site Scripting | Tampering | 1.4 User Credentials | The web server '2. Server' could be subject to a cross-site scripting attack because it does not sanitize untrusted input. | Not applicable, It is not possible to perform cross-site attacks on DB |
| 68 | Persistent Cross Site Scripting | Tampering | 1.4 User Credentials | The web server '2. Server' could be subject to a persistent cross-site scripting attack because it does not sanitize data store 'D3. user credentials ' inputs and output. | Not applicable, It is not possible to perform cross-site attacks on DB |
| 69 | Weak Access Control for a Resource | Information Disclosure | 1.4 User Credentials | Improper data protection of D3. user credentials can allow an attacker to read information not intended for disclosure. Review authorization settings. | Attacker could read the user credentials |
| 93 | Spoofing the 2. Server Process | Spoofing | 1.7 User Confirm | 2. Server may be spoofed by an attacker and this may lead to unauthorized access to 1. 1 Client system. Consider using a standard authentication mechanism to identify the source process. | Client sends the plate number to Attacker's Server. |
| 94 | Spoofing the 1. 1 Client system Process | Spoofing | 1.7 User Confirm | 1. 1 Client system may be spoofed by an attacker and this may lead to information disclosure by 2. Server. Consider using a standard authentication mechanism to identify the destination process. | Client knows that client does not access Server. |
| 95 | Potential Lack of Input Validation for 1. 1 Client system | Tampering | 1.7 User Confirm | Data flowing across 1.7 User Confirm may be tampered with by an attacker. This may lead to a denial of service attack against 1. 1 Client system or an elevation of privilege attack against 1. 1 Client system or an information disclosure by 1. 1 Client system. 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. | Client knows that client does not access Server. |
| 96 | Potential Data Repudiation by 1. 1 Client system | Repudiation | 1.7 User Confirm | 1. 1 Client system 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. | Client repudiates even if the server sends the ID/PW to the access server. |
| 97 | Data Flow Sniffing | Information Disclosure | 1.7 User Confirm | Data flowing across 1.7 User Confirm 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. | Attacker does not profit. |
| 98 | Potential Process Crash or Stop for 1. 1 Client system | Denial Of Service | 1.7 User Confirm | 1. 1 Client system crashes, halts, stops or runs slowly; in all cases violating an availability metric. | Not applicable to our project  Does not consider this case for client |
| 99 | Data Flow 1.7 User Confirm Is Potentially Interrupted | Denial Of Service | 1.7 User Confirm | An external agent interrupts data flowing across a trust boundary in either direction. | Client know that client does not access Server. |
| 100 | Elevation Using Impersonation | Elevation Of Privilege | 1.7 User Confirm | 1. 1 Client system may be able to impersonate the context of 2. Server in order to gain additional privilege. | Not applicable to our project  Because there is no separation of privileges in our project |
| 101 | 1. 1 Client system May be Subject to Elevation of Privilege Using Remote Code Execution | Elevation Of Privilege | 1.7 User Confirm | 2. Server may be able to remotely execute code for 1. 1 Client system. | Not applicable to our project  Because there is no separation of privileges in our project |
| 102 | Elevation by Changing the Execution Flow in 1. 1 Client system | Elevation Of Privilege | 1.7 User Confirm | An attacker may pass data into 1. 1 Client system in order to change the flow of program execution within 1. 1 Client system to the attacker's choosing. | Not applicable to our project  Because there is no separation of privileges in our project |
| 11 | Spoofing the 1. 1 Client system Process | Spoofing | 2.4 Send PlateString | 1. 1 Client system may be spoofed by an attacker and this may lead to unauthorized access to 2. Server. Consider using a standard authentication mechanism to identify the source process. | Fake Client receive vehicle information from server |
| 12 | Spoofing the 2. Server Process | Spoofing | 2.4 Send PlateString | 2. Server may be spoofed by an attacker and this may lead to information disclosure by 1. 1 Client system. Consider using a standard authentication mechanism to identify the destination process. | Client does not receive the vehicle information |
| 13 | Potential Lack of Input Validation for 2. Server | Tampering | 2.4 Send PlateString | Data flowing across 2.4 Send PlateString may be tampered with by an attacker. This may lead to a denial of service attack against 2. Server or an elevation of privilege attack against 2. Server or an information disclosure by 2. Server. 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. | Client receive the unexpected vehicle information because of tampering |
| 14 | Cross Site Scripting | Tampering | 2.4 Send PlateString | The web server '2. Server' could be subject to a cross-site scripting attack because it does not sanitize untrusted input. | Not applicable, the client is not the web. |
| 15 | Potential Data Repudiation by 2. Server | Repudiation | 2.4 Send PlateString | 2. Server 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. | Server repudiates even if client send the platenumber |
| 16 | Data Flow Sniffing | Information Disclosure | 2.4 Send PlateString | Data flowing across 2.4 Send PlateString 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. | Attacker does not profit. |
| 17 | Potential Process Crash or Stop for 2. Server | Denial Of Service | 2.4 Send PlateString | 2. Server crashes, halts, stops or runs slowly; in all cases violating an availability metric. | Server does not work according to the attacker's input. |
| 18 | Data Flow 2.4 Send PlateString Is Potentially Interrupted | Denial Of Service | 2.4 Send PlateString | An external agent interrupts data flowing across a trust boundary in either direction. | Attacket interrupts PlateString flow. Server does not work. |
| 19 | Elevation Using Impersonation | Elevation Of Privilege | 2.4 Send PlateString | 2. Server may be able to impersonate the context of 1. 1 Client system in order to gain additional privilege. | Not applicable to our project  Because there is no separation of privileges in our project |
| 20 | 2. Server May be Subject to Elevation of Privilege Using Remote Code Execution | Elevation Of Privilege | 2.4 Send PlateString | 1. 1 Client system may be able to remotely execute code for 2. Server. | Unintended behavior on the server |
| 21 | Elevation by Changing the Execution Flow in 2. Server | Elevation Of Privilege | 2.4 Send PlateString | An attacker may pass data into 2. Server in order to change the flow of program execution within 2. Server to the attacker's choosing. | Unintended behavior on the server |
| 22 | Cross Site Request Forgery | Elevation Of Privilege | 2.4 Send PlateString | Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker forces a user's browser to make a forged request to a vulnerable site by exploiting an existing trust relationship between the browser and the vulnerable web site. In a simple scenario, a user is logged in to web site A using a cookie as a credential. The other browsers to web site B. Website B returns a page with a hidden form that posts to web site A. Since the browser will carry the user's cookie to web site A, web site B now can take any action on web site A, for example, adding an admin to an account. The attack can be used to exploit any requests that the browser automatically authenticates, e.g. by session cookie, integrated authentication, IP whitelisting. The attack can be carried out in many ways such as by luring the victim to a site under control of the attacker, getting the user to click a link in a phishing email, or hacking a reputable website that the victim will visit. The issue can only be resolved on the server side by requiring that all authenticated state-changing requests include an additional piece of secret payload (canary or CSRF token) which is known only to the legitimate web site and the browser and which is protected in transit through SSL/TLS. See the Forgery Protection property on the flow stencil for a list of mitigations. | Not applicable, the client is not the web. |
| 33 | Spoofing of Destination Data Store D4. Vehicle data | Spoofing | 2.5 Request Vehicle info. | D4. Vehicle data may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of D4. Vehicle data. Consider using a standard authentication mechanism to identify the destination data store. | Not applicable, DB is safely stored in the trust zone |
| 34 | Potential Excessive Resource Consumption for 2. Server or D4. Vehicle data | Denial Of Service | 2.5 Request Vehicle info. | Does 2. Server or D4. Vehicle data take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout. | Not applicable, DB server is not separated |
| 35 | Spoofing of Source Data Store D4. Vehicle data | Spoofing | 2.6 Vehicle Information | D4. Vehicle data may be spoofed by an attacker and this may lead to incorrect data delivered to 2. Server. Consider using a standard authentication mechanism to identify the source data store. | Not applicable, DB is safely stored in the trust zone |
| 36 | Cross Site Scripting | Tampering | 2.6 Vehicle Information | The web server '2. Server' could be subject to a cross-site scripting attack because it does not sanitize untrusted input. | Not applicable, It is not possible to perform cross-site attacks on DB |
| 37 | Persistent Cross Site Scripting | Tampering | 2.6 Vehicle Information | The web server '2. Server' could be subject to a persistent cross-site scripting attack because it does not sanitize data store 'D4. Vehicle data' inputs and output. | Not applicable, It is not possible to perform cross-site attacks on DB |
| 38 | Weak Access Control for a Resource | Information Disclosure | 2.6 Vehicle Information | Improper data protection of D4. Vehicle data can allow an attacker to read information not intended for disclosure. Review authorization settings. | Attacker could read the user credentials |
| 88 | Spoofing of the E4. Email Service External Destination Entity | Spoofing | 1.5 Send email | E4. Email Service may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of E4. Email Service. Consider using a standard authentication mechanism to identify the external entity. | Not applicable to our project |
| 89 | External Entity E4. Email Service Potentially Denies Receiving Data | Repudiation | 1.5 Send email | E4. Email Service claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data. | Not applicable to our project |
| 90 | Data Flow 1.5 Send email Is Potentially Interrupted | Denial Of Service | 1.5 Send email | An external agent interrupts data flowing across a trust boundary in either direction. | Not applicable to our project |
| 23 | Spoofing the 2. Server Process | Spoofing | 2.7 retrieved information | 2. Server may be spoofed by an attacker and this may lead to unauthorized access to 1. 1 Client system. Consider using a standard authentication mechanism to identify the source process. | Client receives the unexpected vehicle information |
| 24 | Spoofing the 1. 1 Client system Process | Spoofing | 2.7 retrieved information | 1. 1 Client system may be spoofed by an attacker and this may lead to information disclosure by 2. Server. Consider using a standard authentication mechanism to identify the destination process. | Attacker could steal vehicle detail information |
| 25 | Potential Lack of Input Validation for 1. 1 Client system | Tampering | 2.7 retrieved information | Data flowing across 2.7 retrieved information may be tampered with by an attacker. This may lead to a denial of service attack against 1. 1 Client system or an elevation of privilege attack against 1. 1 Client system or an information disclosure by 1. 1 Client system. 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. | Client receives the unexpected vehicle information |
| 26 | Potential Data Repudiation by 1. 1 Client system | Repudiation | 2.7 retrieved information | 1. 1 Client system 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. | Client repudiates even if server send the vehicle information |
| 27 | Data Flow Sniffing | Information Disclosure | 2.7 retrieved information | Data flowing across 2.7 retrieved information 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. | Attacker could steal vehicle detail information |
| 28 | Potential Process Crash or Stop for 1. 1 Client system | Denial Of Service | 2.7 retrieved information | 1. 1 Client system crashes, halts, stops or runs slowly; in all cases violating an availability metric. | Not applicable to our project  Does not consider this case for client |
| 29 | Data Flow 2.7 retrieved information Is Potentially Interrupted | Denial Of Service | 2.7 retrieved information | An external agent interrupts data flowing across a trust boundary in either direction. | Client receives the unexpected vehicle information |
| 30 | Elevation Using Impersonation | Elevation Of Privilege | 2.7 retrieved information | 1. 1 Client system may be able to impersonate the context of 2. Server in order to gain additional privilege. | Not applicable to our project  Because there is no separation of privileges in our project |
| 31 | 1. 1 Client system May be Subject to Elevation of Privilege Using Remote Code Execution | Elevation Of Privilege | 2.7 retrieved information | 2. Server may be able to remotely execute code for 1. 1 Client system. | Not applicable to our project  Because there is no separation of privileges in our project |
| 32 | Elevation by Changing the Execution Flow in 1. 1 Client system | Elevation Of Privilege | 2.7 retrieved information | An attacker may pass data into 1. 1 Client system in order to change the flow of program execution within 1. 1 Client system to the attacker's choosing. | Not applicable to our project  Because there is no separation of privileges in our project |

## 

## 5.2. PnG

We did the PnG to find more threats.

| **PnG 1** | **Type** | Criminal organization |
| --- | --- | --- |
| **Goal** | Stealing or destroying components of the system |
| **Motivation** | Using in crime |
| **Skill** | Physical power and ability to use various equipment  knowledge of place and system |
| **Misuse case** | By physically destroying the system, the system no longer works  Steal the system(equipments) containing the information |
| **PnG 2** | **Type** | Internal Engineer |
| **Goal** | Exporting inside information and selling it for money outside |
| **Motivation** | Monetary gain |
| **Skill** | Access to infrastructure  Exporting inside information  Negotiation skills with people who need inside information |
| **Misuse case** | Disclose administrator’s ID/Password or required information to the buyer  Export important information |
| **PnG 3** | **Type** | Hacker |
| **Goal** | Post the achievements of hacking on the internet |
| **Motivation** | For recognition and laughs |
| **Skill** | Extensive knowledge of network protocols and hacking programs. |
| **Misuse case** | Sniff the communication channel between server and client to get user credential data.  The system cannot operate any more by DDos attack |
| **PnG 4** | **Type** | Hacker hired by criminal groups |
| **Goal** | Tampering information of the system |
| **Motivation** | Monetary gain |
| **Skill** | Extensive knowledge of network protocols and hacking programs. |
| **Misuse case** | Access the system and tamper with information.  Sniff the communication channel between server and client to get user credential data. |

## 5.3. Brainstorming

We also try to brainstorm to find the threats.

| **Threat** | **Category** | **Interaction in DFD** |
| --- | --- | --- |
| Attackers could modify the plate number between client and server. Then the Server could not receive the right vehicle information. | Tampering | 2.4 Send PlateString |
| Attacker inserts the exploit to send a message of ID/PW using fake Client.  The exploit sends the user credentials to the attacker's server when the Server process reads the user credential to verificate.  Attachers could steal the user credentials. | Spoofing  Information Disclosure | 1.2 Send ID/PW  1.4 User Credentials |
| Attacker could steal the retrieved vehicle informations between server and client | Information Disclosure | 2.6 Vehicle Information |
| Attacker could make a server unavailable that fake clients send the mass request to server | Denial of Service | 1.2 Send ID/PW |
| Attacker could sniff the email. Then response to Server to complete the authentification | Information Disclosure | 1.5 Send email |

## 5.4. Result of Threat Modeling

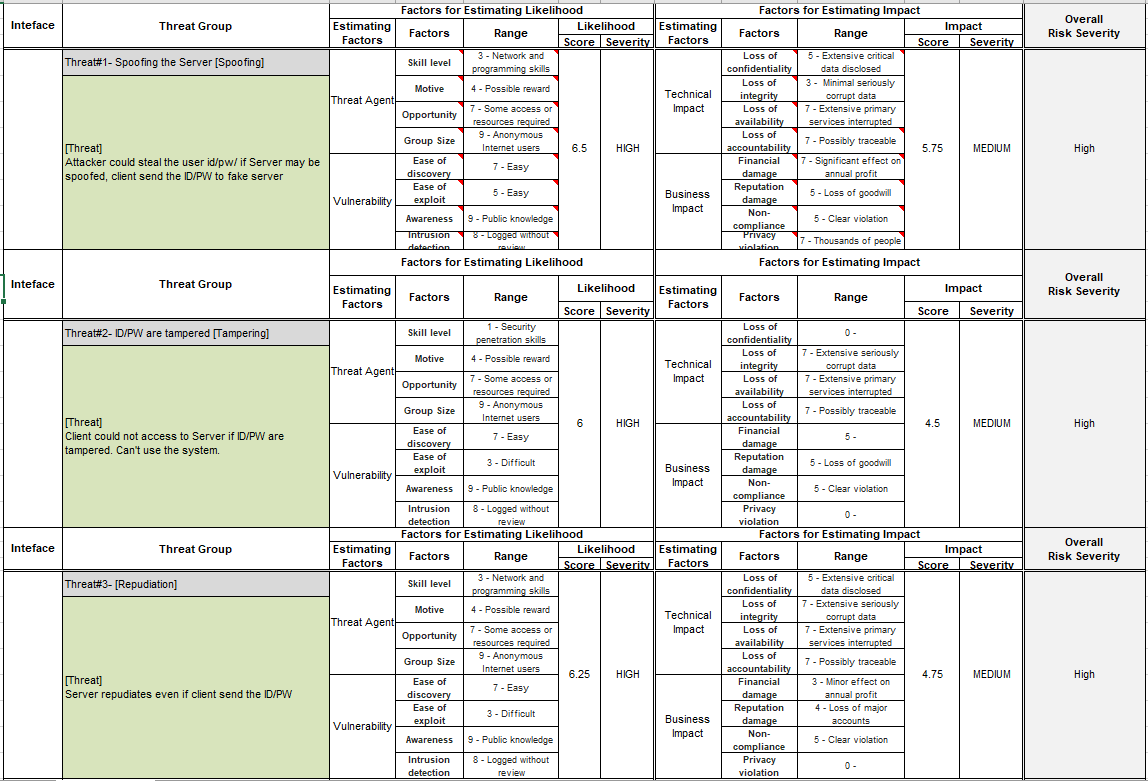
We identify the 18 threats from STRIDE, PnG and brainstorming after analyzing and merging duplicate threats.

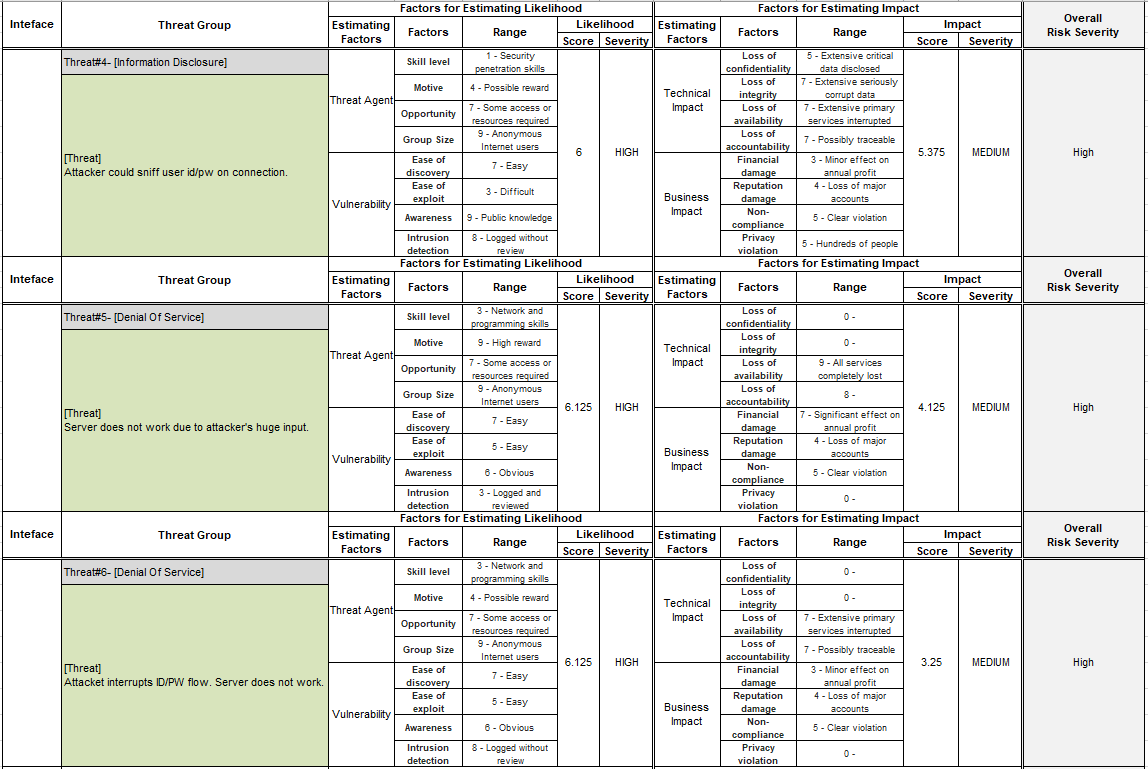
| **TH\_ID** | **Threat Description** | **Category** |
| --- | --- | --- |
| TH\_01 | If server may be spoofed, client send the ID/PW to fake server and then attacker could steal the user ID/PW | Spoofing |
| TH\_02 | Clients could not access the Server if ID/PW are tampered. Can't use the system. | Tampering |
| TH\_03 | Server repudiates even if client send the ID/PW | Repudiation |
| TH\_04 | Attacker could steal privacy data between server and client communication | Information Disclosure |
| TH\_05 | Server does not work due to attacker's a lot of attempt | Denial Of Service |
| TH\_06 | Attacker interrupts ID/PW flow. Server does not work | Denial Of Service |
| TH\_07 | Attacker steals the user credential/DB data in server through unauthorized access remotely | Information Disclosure |
| TH\_08 | Client does not send the plate number if the user confirmed from server is tampered | Tampering |
| TH\_09 | Client sends the plate number to Attacker's Server. Client does not receive the retrieved vehicle information. | Spoofing |
| TH\_10 | Server repudiates even if client send the plate number | Repudiation |
| TH\_11 | Attacker could be received vehicle info from server in case that fake client send the plate number to server | Information Disclosure |
| TH\_12 | Client receive the wrong vehicle information if the retrieved vehicle informations is tampered | Spoofing |
| TH\_13 | Client repudiates even if server send the vehicle information | Tampering |
| TH\_14 | Attacker could sniff vehicle detail information from server on connection | Tampering |
| TH\_15 | Attacker could sniff the email. Then complete the authentication. | Repudiation |
| TH\_16 | if attacker manipulate the OTP, Server can't authenticate OTP normally | Information Disclosure |
| TH\_17 | Attacker could sniff the email. Then complete the authentication. | Information Disclosure |
| TH\_18 | Server doesn't operate 2-factor authentication normally if an attacker manipulates the OTP. | Tampering |

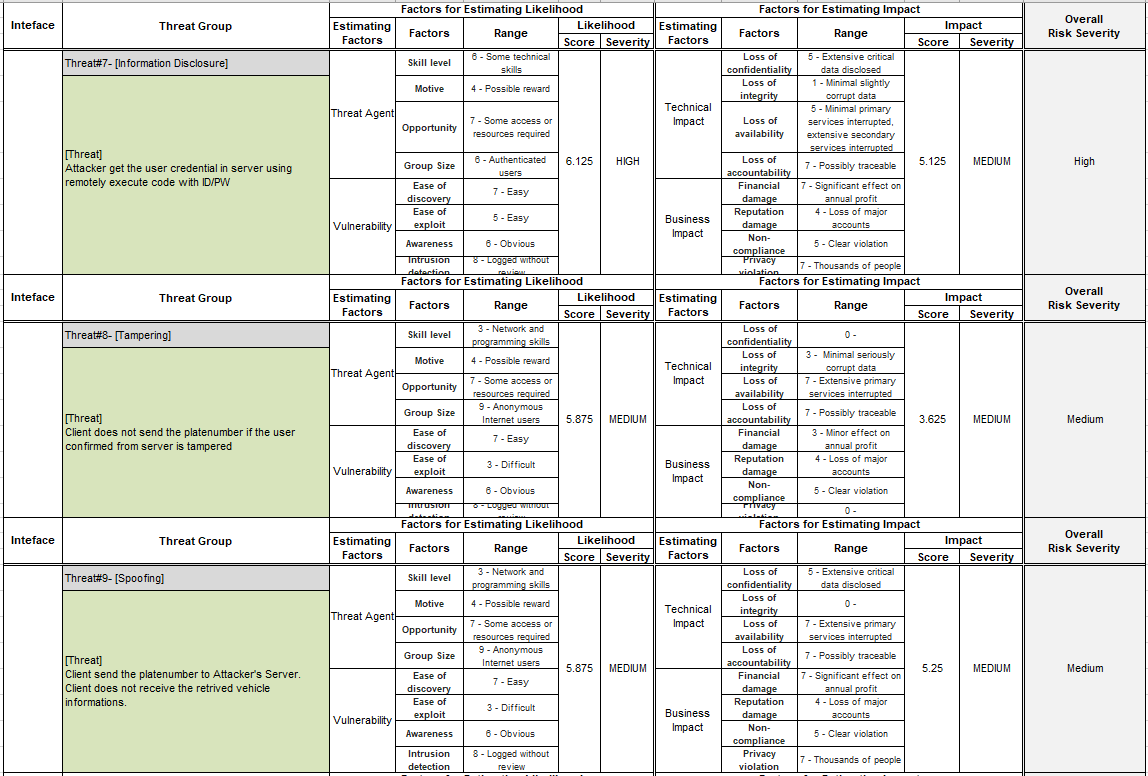
# 

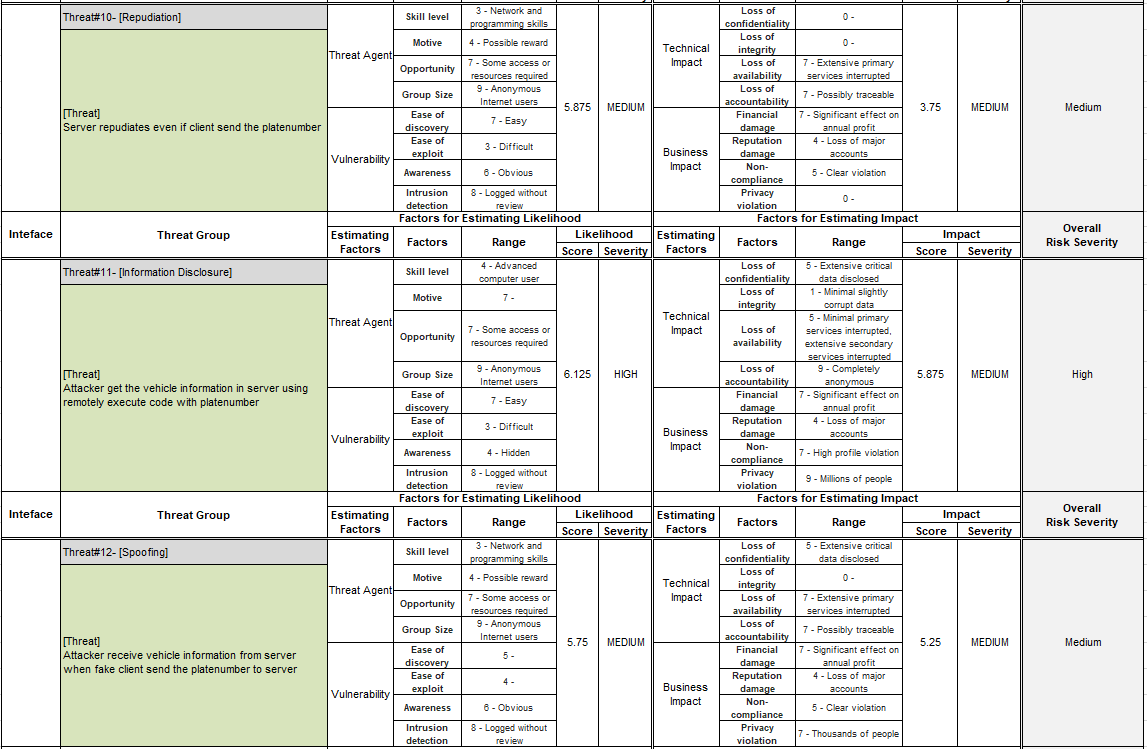
# 6. Security Risk Assessment

We measured the threat level using the OWASP tool. And 4 items with high scores were identified.



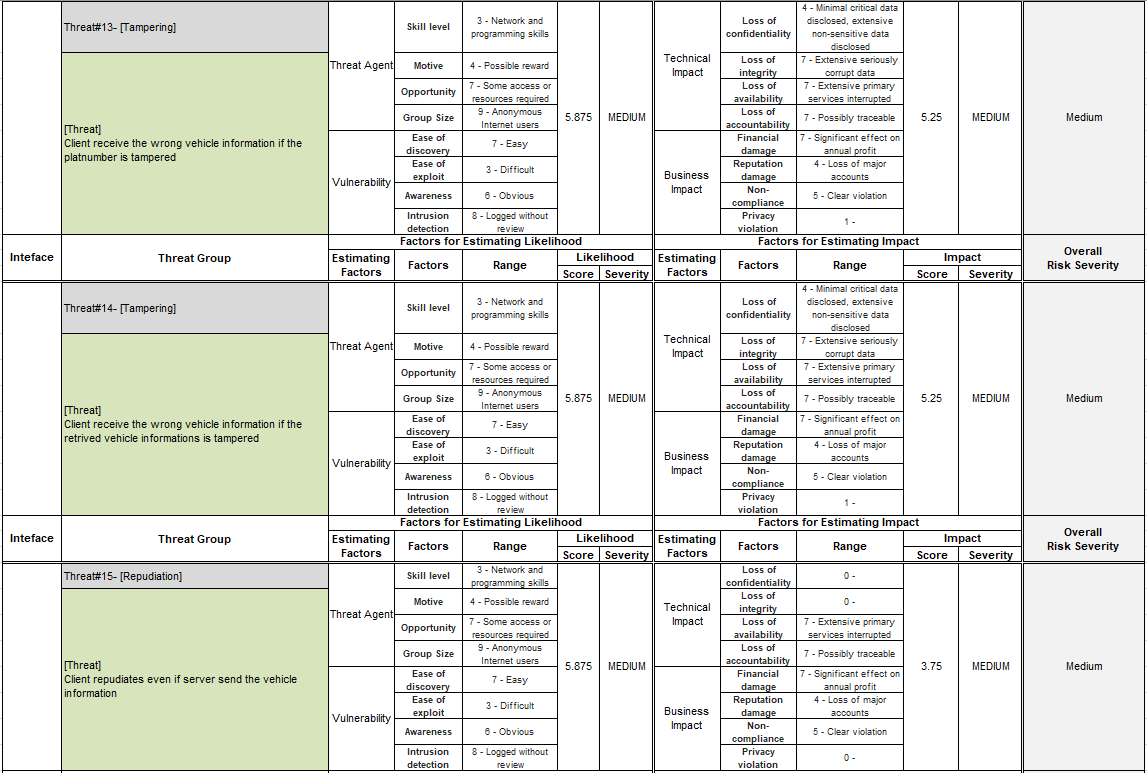
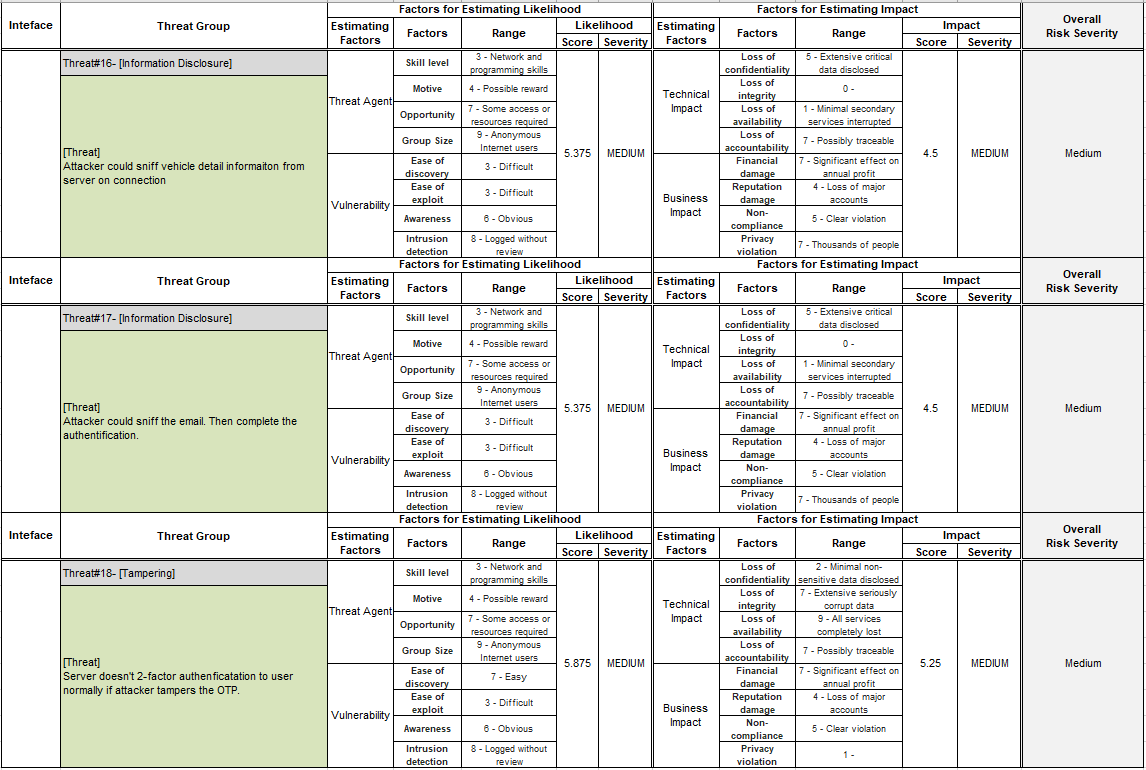












# 7. Security Requirements

We identify the security requirements to protect the threats.

| **SR\_ID** | **Security Requirement** | **TH\_ID** |
| --- | --- | --- |
| SR\_01 | Client and server must be authenticated to communication | TH\_01, TH\_09, TH\_12 |
| SR\_02 | The channel between client and server must be encrypted | TH\_04, TH\_07, TH\_11, TH\_16, TH\_17 |
| SR\_03 | The message between client and server must not modified on channel | TH\_02, TH\_08, TH\_13, TH\_14, TH\_18 |
| SR\_04 | Password must be encrypted | TH\_07 |
| SR\_05 | Different two authentication factor(Knowledge, Possession or Inherence) must be used | TH\_17, TH\_18 |
| SR\_06 | The OTP must not be exposed | TH\_17 |
| SR\_07 | Saved retrieved information in client must be encrypted | TH\_7, TH\_11 |
| SR\_08 | The communications between client and server must be stored to log | TH\_3, TH\_10, TH\_15 |
| SR\_09 | Private key must not be exposed | TH\_1, TH\_9, TH\_12 |
| SR\_10 | User credentials/vehicle information in server must be encrypted | TH\_7 |
| SR\_11 | Server must restrict the query from client to prevent the brute attack | TH\_5, TH\_6 |
| SR\_12 | Server must check the input validation (code injection) | TH\_7 |

# 

# 8. Mitigation

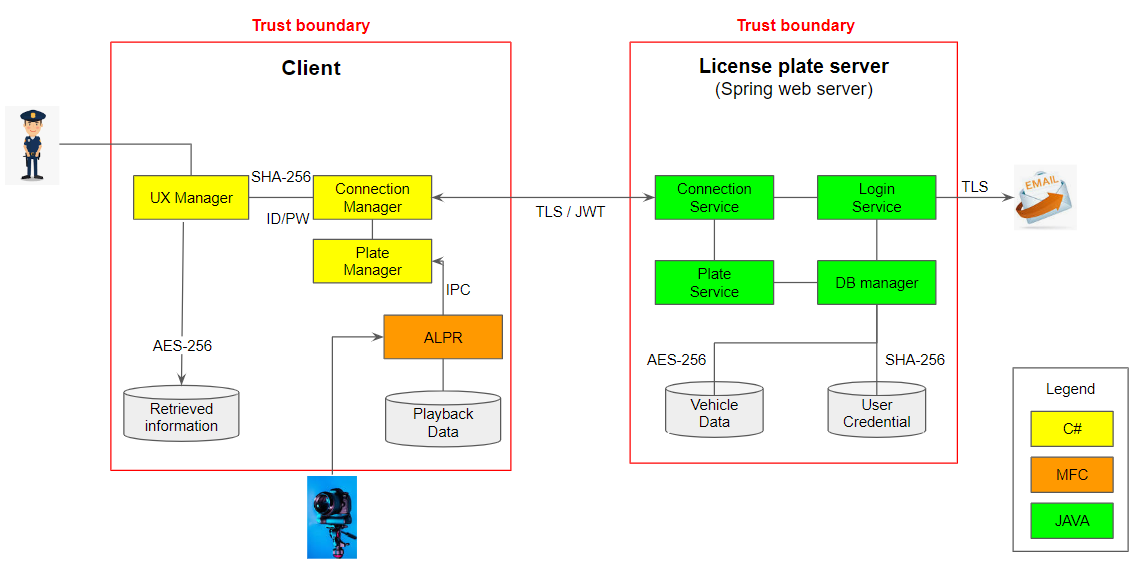
We describe the mitigations to satisfy the security requirements.

| **MI\_ID** | **Mitigation** | **Related SR\_ID** |
| --- | --- | --- |
| MI\_01 | client and server could be authenticated communicate  - mutual authentication | SR\_01 |
| MI\_02 | Server give the access authority to Client  - JWT | SR\_02 |
| MI\_03 | Channel encryption between client and server  - apply TLS 1.2 or higher | SR\_02, SR\_03, SR\_06 |
| MI\_04 | Password encryption  - Bcrypt or SHA-256 or more | SR\_04 |
| MI\_06 | Input validation  - Add perimeter filter(sanitizer) | SR\_11 |
| MI\_07 | Two authentication factor  - Use Password and OTP | SR\_05 |
| MI\_08 | Data encryption(Retrieved information, User credentials, Vehicle information)  - AES256 or more | SR\_07, SR\_10 |
| MI\_09 | The communications log  - Request and response messages must be stored on the client and server respectively. | SR\_08 |
| MI\_10 | Encryption key protect  - HSM | SR\_09 |
| MI\_11 | Input validation  - use JPA(JAVA Persistence API) | SR\_12 |

# 9. Architecture

## 9.1. Overall SW Architecture

Below is the SW architecture of System. Client has a response to interaction with the officer and sends the plate number to get vehicle information. Server manages the user authentication and give the vehicle information to the client according to request of the client.



## 9.2. Used Open Source

### 9.2.1. Client

| **SW** | **Version** | **Description** |
| --- | --- | --- |
| openALPR | 2.3.0 | Automatic License Plate Recognition engine |
| NLog | 5.0.1 | C# log lib |
| Newtonsoft.Json | 6.0.4 | handling the json format |
| Microsoft.AspNet.WebApi.Client | 5.2.9 | Serve the web communication |

### 9.2.2. Server

| **SW** | **Version** | **Description** |
| --- | --- | --- |
| SpringFramework | 2.7.0 | Java-based web platform |
| Tomcat | 9.0.63 | Web application server |
| JWT | 0.11.2 | plugin for secure communications |
| JPA | 2.7.0 | standard interface for ORM(Object-Relational Mapping) |
| H2DB | 2.1.212 | java-based database management |

## 9.3. Crypto algorithms

### 9.3.1. TLS

Used Source: OpenSSL(OpenSSL 1.1.1)/ Java keytool

- Server

1. Generate a private key
2. Generate public key and CSR for the web server
3. Extract the pfx file from the CSR and Private key
4. Register the keystore(using java keytool) to enable TLS

- Client

1. Install pfx file as certification. pfx file is generated by Server

### 

### 9.3.2. Encryption Data

AES-256 is used for Vehicle Information in Server and Retributed information in Client.

The Client/Server used each Private Key.

### 9.3.3. JWT(JSON Web Token)

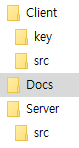
If the client has been identified by the server, the server gives the JWT token to the client.

The structure of JWT is:

* Header : Identifies which algorithm is used to generate the signature. HMAC-SHA256 is used in this project.
* Payload : Contains a set of claims. This includes user identifier and issued date of token.
* Signature : Securely validate the token by secret key. If the server checks the token is valid, the server will give authority to the client.  
   HMAC\_SHA256(  
   secret,  
   base64urlEncoding(header) + '.' +  
   base64urlEncoding(payload)  
   )

Client will use the issued token for every request to the server and the token is placed in the HTTP header. The server will check if the token is valid. If valid, the server will grant the client's request.

## 9.4. Source Directory



# 10. Implementation & Test Result

## 10.1. Implementation

| **MI\_ID** | **Mitigation** | **Implementation** |
| --- | --- | --- |
| MI\_01 | client and server could be authenticated communicate  - mutual authentication | authentication validation using CA |
| MI\_02 | Server give the access authority to Client  - JWT | Apply JWT for communication |
| MI\_03 | Channel encryption between client and server  - apply TLS 1.2 or higher | Apply TLS 1.2 |
| MI\_04 | Password encryption  - Bcrypt or SHA-256 or more | Apply SHA-256 |
| MI\_06 | Input validation  - Add perimeter filter(sanitizer) | Not implemented yet |
| MI\_07 | Two authentication factor  - Use Password and OTP | OTP is sent to the user using email.  PW and OTP for 2 factor authentication |
| MI\_08 | Data encryption(Retrieved information, User credentials, Vehicle information)  - AES256 or more | Apply AES256 |
| MI\_09 | The communications log  - Request and response messages must be stored on the client and server respectively. | Generate log file on client and server |
| MI\_10 | Encryption key protect  - HSM | Not implemented yet |
| MI\_11 | Input validation  - use JPA(JAVA Persistence API) | Apply JPA |

## 

## 

## 10.2. Test Result

| **TC No.** | **Trace To SR** | **TC Name** | **Pre-Condition** | **Step No.** | **Description** | **Expected result** | **Actual result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TC\_01 | SR\_01 | mutual authentication validation | Delete client certification | 1 | Input correct PW = yyyyyy | Unauthorized access | Index widow =Unauthorized access | Pass |
| 2 | push login button |
| Delete server certification | 3 | Input correct PW = yyyyyy | Unauthorized access | success PW  OTP input box = activate | Pass |
| 4 | push login button |
| TC\_02 | SR\_05 | PW Validation | Client and Server have correct authenticate | 1 | Input wrong PW = xxxxxxx | Wrong PW(Please confirm your PW) | Index widow = Log-in failed  Please check your ID/PW | Pass |
| 2 | push login button |
| 3 | Input correct PW = yyyyyy | success PW  activate OTP component | success PW  OTP input box = activate | Pass |
| 4 | push login button |
|  |  | 5 | Input wrong OPT =xxxxxx | Wrong OTP | OTP failed  Index widow = your OTP is wrong, Please confirm your OTP number | Pass |
| 6 | push OPT button |
| 7 | wait for 1min(TBD) without any action |  |  |  |
| 8 | Input correct OTP = yyyyyy | success LogIn | success LogIn | Pass |
| 9 | push OPT button |
| LogOut | success LogIn | 10 | push logout button | all connection with server are disconnected | all connection with server are disconnected | Pass |
| TC\_03 | SR\_02 | Access authority  (JWT Test) | success LogIn | 1 | Input plate number with invalid JWT | Invalid Token | Invalid Token | Pass |
| 2 | push plate number button |
| 3 | Input plate number with valid JWT | Normal operation | Normal operation | Pass |
| 4 | push plate number button |
| TC\_04 | SR\_03 | Channel encryption  (Communication Test) | success LogIn,  Valid JWT | 1 | Send Message(plate number : xxx)  Client to Server | Receive Same Message | Receive Same Message | Pass |
| 2 | Confirm Message(Vehicle information : xxx)  in Server |
| 3 | Send Message(plate number : xxx)  Server to Client | Receive Same Message | Receive Same Message | Pass |
| 4 | Confirm Message(Vehicle information : xxx)  in Client |
| TC\_05 | SR\_04 | Password encryption Test | success LogIn | 1 | Confirm encrypted password | Password is encrypted | Password is encrypted | Pass |
| TC\_06 | SR\_07 | Data encryption Test | success LogIn and communication between Client and Sever | 1 | Checking DB file though H2-console window of Sever | data is encrypted | data is encrypted | Pass |
| 2 | Checking Vehicle Info file of Client | data is encrypted | data is encrypted | Pass |
| TC\_07 | SR\_08 | Store log Test | success LogIn and communication between Client and Sever | 1 | Checking log file(text) of Sever | log stored | log stored | Pass |
| 2 | Checking log file(text) of Client | log stored | log stored | Pass |

| **TC No.** | **Trace To SR** | **TC Name** | **Description** | **Evidence** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| TC\_08 | SR\_12 | Code Review | Checking suitably apply JPA in the code |  | Pass |

# 11. Guide

## 11.1. Setup Guide

### 11.1.1. Tools

Client

| **Tool** | **Version** |
| --- | --- |
| .NET Framework | 4.7.2 |

### 

Server

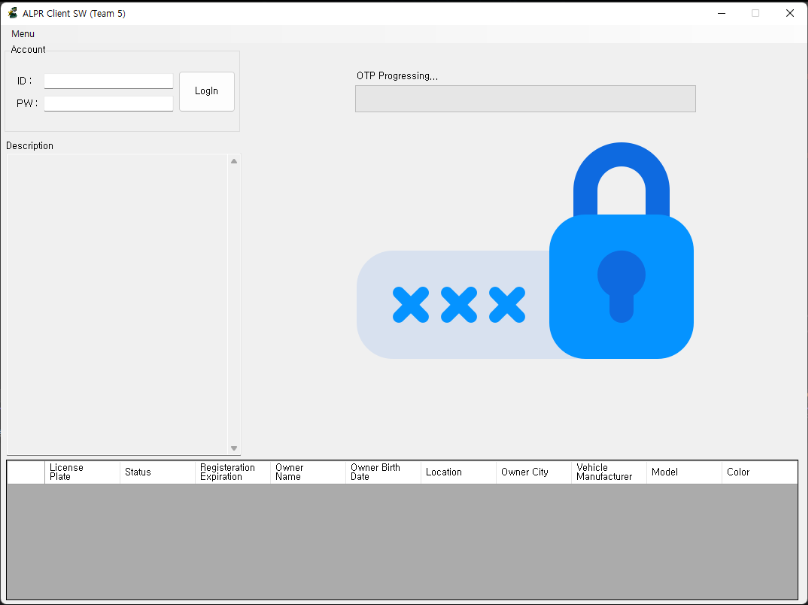
| **Tool** | **Version** |
| --- | --- |
| JAVA JRE | openjdk 11.0.12 |
| Eclipse | eclipse-jee-2022-06-R-win32-x86\_64 |
| openssl | 1.1.1 |
| Lombok | 1.18.24 |

### 11.1.2. Setup environment

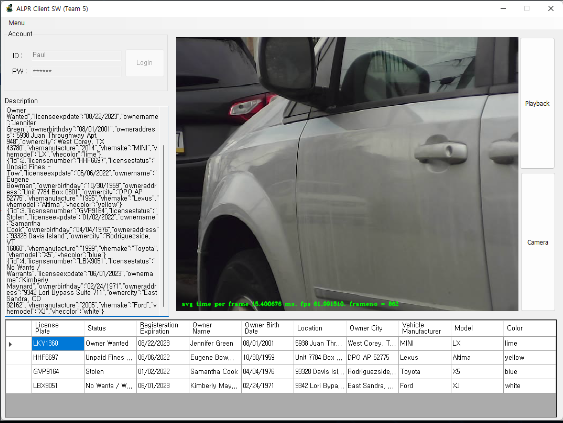
| Client | - Execution environment  1. Visual Studio 2022 Community  - .NET 4.7.2 Installation  2. Change the window display setting DPI to 100%  3. Build as below through ALPR\_Client folder  1) Double-click the client key file to install as default (File : Client.pfx, Password: qwe123.. , Location: ..\ALPR\_Client\Client\_Key)  2) Make a web server and run server (Refer to Server build Guide) and check the server IP  3) Open the OpenALPR.sln solution (Location: ..\ALPR\_Client)  4) Check the existence of lgdemo\_w (MFC project) and WindowsForms\_Clien (C# project) projects in Solution  5) Start build (Ref. Necessary libraries and reference links are made inside, build should be successful if the folder configuration is not changed)  7) Do not change or delete file in debug/release folder without execution file (location : ..\ALPR\_Client\WindowsForms\_client\bin\Debug)  6) Designate WindowsForms\_Client as the startup project  7) Change the serverURL value on line 58 of the form1.cs to your server IP obtained in 2) (as it is when configured as a local server)  8) Check whether the Windows client app is running normally by executing the build (ALPRClient.exe) |
| --- | --- |
| Server | ########## How to install or set for the Project environment ##########  1. JAVA JDK(openjdk 11.0.12 2021-07-20) - https://jdk.java.net/18/  1-1. install JDK  Go to https://jdk.java.net/18/.  Select the appropriate JDK version and click Download.  1-2. setting for environment of JAVA (Set JAVA\_HOME)  Right click My Computer and select Properties.  On the Advanced tab, select Environment Variables,and then edit JAVA\_HOME to point to where the JDK software is located  for Example, C:\Program Files\ojdkbuild\java-1.8.0-openjdk  Check to set normally with the CMD ("java --version").  2. Eclipse(eclipse-jee-2022-06-R-win32-x86\_64) -  download IDEwith eclipse for the PROJECT.  https://www.eclipse.org/downloads/packages/  select package to "Eclipse IDE for Enterprise Java and Web Developers"  3. Lombok(1.18.24) - https://projectlombok.org/download  Download java from the site(https://projectlombok.org/download)  Copy jar to Eclipse installed path..  Open the CMD(shift + mouse right click and open power shell) and run the command below.  $ java -jar lombok.jar.  Select specify location and input the eclipse path.  EX) E:\\_Dev\eclipse-jee-2022-06-R-win32-x86\_64\eclipse  Click the button "install / update" and quit the installer.  4. SpringFramework(2.7.0) - can be installed through the eclipse market  Execute eclipse, and select Help -> Eclipse Marketplace..  Input the text "Spring tool" or "STS" and install package(option is stay with initial setting)  ########## How to import the PROJECT with eclipse##########  1. execute the IDE eclipse.  2. make "the workspace" after input the any path.  3. on the left side window, select "import project"  4. expand "Gradle" item on window, select "Existing Gradle Project"  5. find the project with "Browse.." and finish.  6. after importing project, refresh build.gradle with right click  7. Click Boot Dashboard icon on Eclipse tool(this button is green color and the "PowerOn" shape)  \*\* following tools don't have to set  \* Tomcat(9.0.63) - embedded in spring boot  \* update JAVA API and Libraries - Gradle Java library plugin  JWT / JPA (2.7.0) / H2DB (2.1.212)  ########## How to build and release (JAR)#########  1. select "RUN > Run Configuration" on the menu of Eclipse TOOL  2. Gradle Task double click and push the add button.  3. Input "bootjar" in the Gradle Task edit box.  4. Select workspace > Project(T5Defense\_Server)  5. apply and run  6. check the path "T5Defense\_Server\build\libs" in the project root path  ########## How to add a test account to the DB#########  The main DB file name of the server is <project>/T5Defense\_Server/local.mv.db  In the given DB file, license information identified by plate number and 1 user account are set by default. In our project, there is no sign-up process and it is assumed that the users to be authenticated are registered in advance. If you want to create a test account and see how it works, you need to proceed as follows.  1. open file <project>/T5Defense\_Server/src/main/java/com/defense/server/controller/LoginController.java  2. There is a signUp() method commented out. Uncomment it.  3. Rebuild and launch the server  4. Send http POST request to the server http request body example:  {  "userid":"your id",  "password":"your password",  "email":"your email"  }  5. If registration is successful, code 200 is returned in response.  The signUp() method is only added to help set the environment for Team 1. In the real environment, this code does not exist, and it is assumed that users are pre-registered in the DB. Therefore, DO NOT ASSUME THIS CODE TO BE A VULNERABILITY. |

## 11.2. User Guide

1. Insert the ID/PW
2. If the user inserts the correct ID/PW, the user will receive the OTP using registered email
3. Insert OTP. The input box for OTP will be shown after validation of ID/PW.



1. The user can choose which method(playback file or camera) is used to recognize the plate number using the menu button.
2. The user can receive the retributed information after sending the plate number to the server.



1. The user can save the retributed information to csv file by menu
2. The user can encrypt and decrypt the csv file.