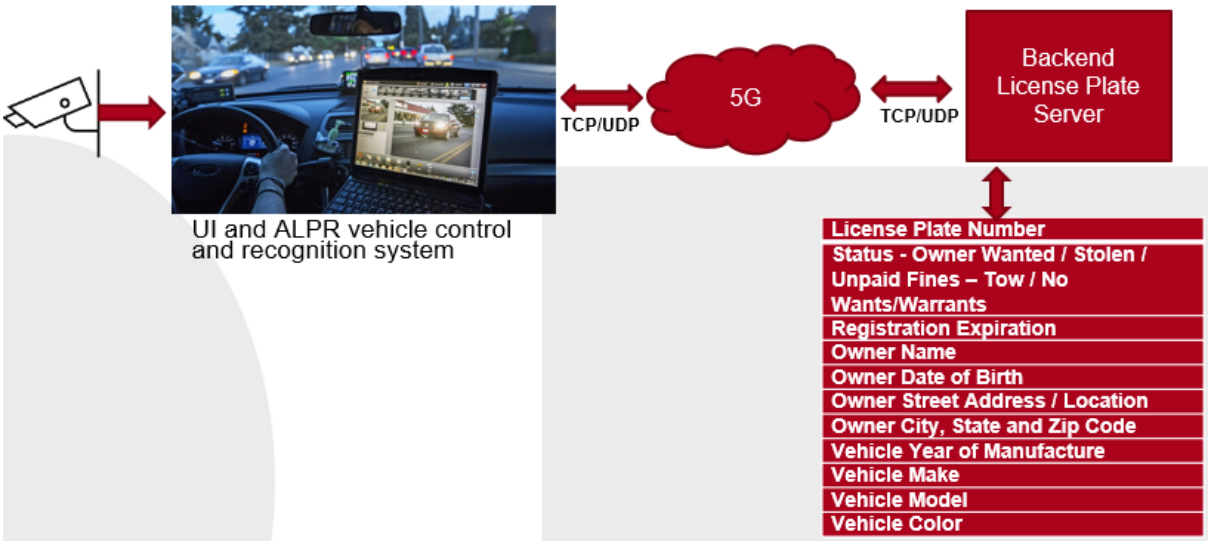
Automatic License Plate Recognition System (Team5)

Table of Contents

0. Introduction

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



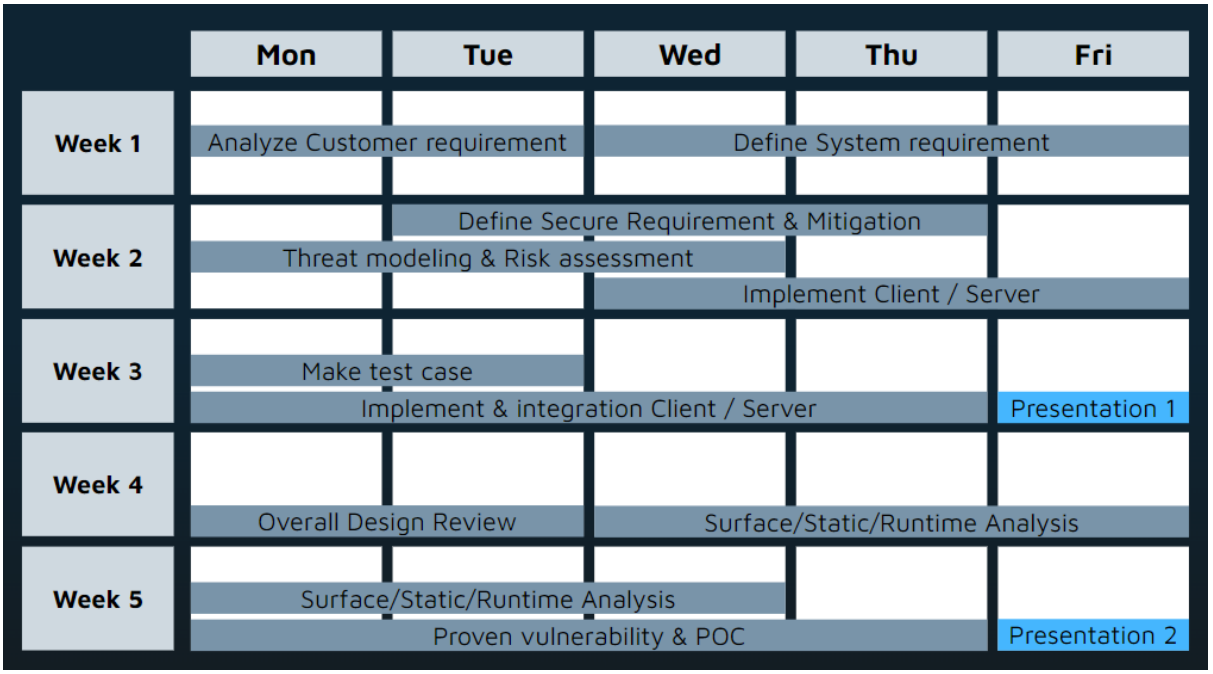
Role and Responsibilities

Name	Phase 1	Phase 2
Paul Lim	- Team Leader - Client App Development - Client Security	- Static Analysis - Reverse Engineering
Jong Soo Oh	- Architecture - System/Security Requirement	- Design Analysis - Runtime Analysis
Jinhwan Kim	- Threat Modeling - Risk Assessment - Validation	- Surface Analysis - Code Review
Sangwook Lee	- Server Security - 2FA (OTP) - DB, Mail management	- Static Analysis - Runtime Analysis
Dawoon Park	- Server Development - Server Security - TLS, Certificate	- Reverse Engineering - Code Review
Minyong Ha	- Server Security - Authentication Design - JWT, Cryptography	- Authentication /Cryptography Analysis

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



Phase 1: Secure Development

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.

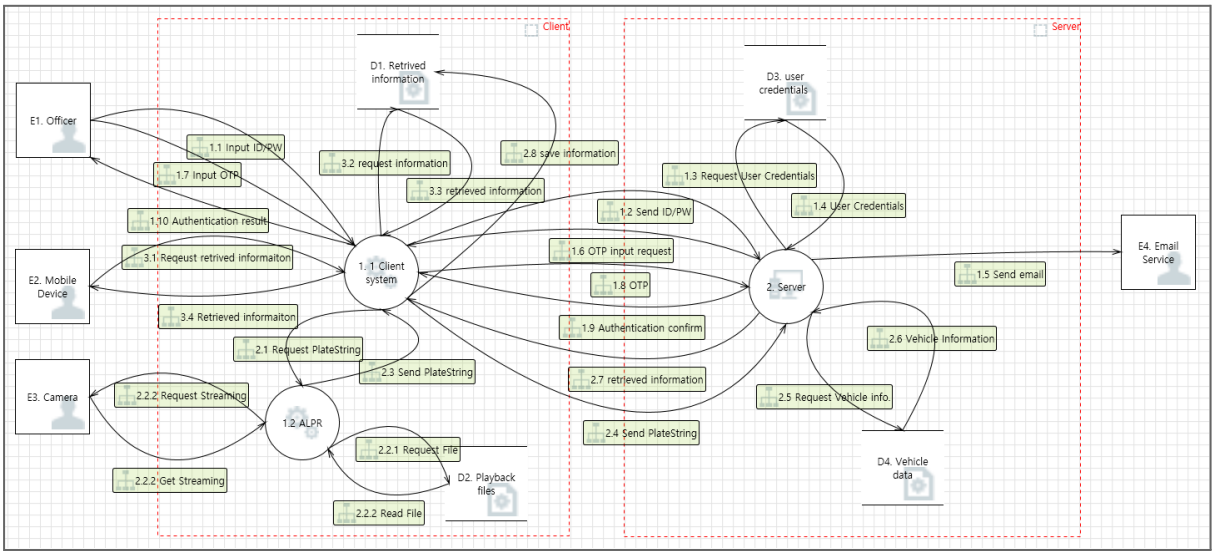
Assets
User credentials
Plate number
Vehicle information data
Client/Server connection

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	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

				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.	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	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

	Code Execution				
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	Client knows that client does not access Server.

				identify the destination process.	
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	2.4 Send	2. Server claims that it did not receive data from a source outside	Server repudiates even if client send the

	Repudiation by 2. Server		PlateString	the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	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	Denial Of Service	2.5 Request	Does 2. Server or D4. Vehicle data take explicit steps to control resource consumption? Resource	Not applicable, DB server is not separated

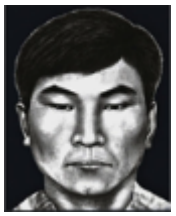
	Consumption for 2. Server or D4. Vehicle data		Vehicle info.	consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.	
35	Spoofing of Source Data Store D4. Vehicle data	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	2. Server may be spoofed by an attacker and this may lead to unauthorized access to 1. 1 Client	Client receives the unexpected vehicle information

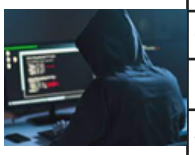
			information	system. Consider using a standard authentication mechanism to identify the source process.	
24	Spoofing the 1. 1 Client system Process	Spoofing	2.7 retrieved information	1. 1 Client system may be spoofed by an attacker and this may lead to information disclosure by 2. Server. Consider using 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	Denial Of Service	2.7 retrieved informa	An external agent interrupts data flowing across a trust boundary in either direction.	Client receives the unexpected vehicle information

	Interrupted		tion		
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.

Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#1- Spoofing the Server [Spoofing]	Threat Agent	Skill level	3 - Network and programming skills	6.5	HIGH	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	5.75	MEDIUM	High
	[Threat] Attacker could steal the user id/pw/ if Server may be spoofed, client send the ID/PW to fake server		Motive	4 - Possible reward				Loss of integrity	3 - Minimal seriously corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	5 - Easy				Reputation damage	5 - Loss of goodwill			
			Awareness	9 - Public knowledge				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	7 - Thousands of people			
Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#2- ID/PW are tampered [Tampering]	Threat Agent	Skill level	1 - Security penetration skills	6	HIGH	Technical Impact	Loss of confidentiality	0 -	4.5	MEDIUM	High
	[Threat] Client could not access to Server if ID/PW are tampered. Can't use the system.		Motive	4 - Possible reward				Loss of integrity	7 - Extensive seriously corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	5 -			
			Ease of exploit	3 - Difficult				Reputation damage	5 - Loss of goodwill			
			Awareness	9 - Public knowledge				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	0 -			
Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#3- [Repudiation]	Threat Agent	Skill level	3 - Network and programming skills	6.25	HIGH	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	4.75	MEDIUM	High
	[Threat] Server repudiates even if client send the ID/PW		Motive	4 - Possible reward				Loss of integrity	7 - Extensive seriously corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	3 - Minor effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	9 - Public knowledge				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	0 -			

Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#4- [Information Disclosure]	Threat Agent	Skill level	1 - Security penetration skills	6	HIGH	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	5.375	MEDIUM	High
	[Threat] Attacker could sniff user id/pw on connection.		Motive	4 - Possible reward				Loss of integrity	7 - Extensive seriously corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	3 - Minor effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	9 - Public knowledge				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	5 - Hundreds of people			
Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#5- [Denial Of Service]	Threat Agent	Skill level	3 - Network and programming skills	6.125	HIGH	Technical Impact	Loss of confidentiality	0 -	4.125	MEDIUM	High
	[Threat] Server does not work due to attacker's huge input.		Motive	9 - High reward				Loss of integrity	0 -			
			Opportunity	7 - Some access or resources required				Loss of availability	9 - All services completely lost			
			Group Size	9 - Anonymous Internet users				Loss of accountability	8 -			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	5 - Easy				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	3 - Logged and reviewed				Privacy violation	0 -			
Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#6- [Denial Of Service]	Threat Agent	Skill level	3 - Network and programming skills	6.125	HIGH	Technical Impact	Loss of confidentiality	0 -	3.25	MEDIUM	High
	[Threat] Attacket interrupts ID/PW flow. Server does not work.		Motive	4 - Possible reward				Loss of integrity	0 -			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	3 - Minor effect on annual profit			
			Ease of exploit	5 - Easy				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	0 -			

Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#7- [Information Disclosure]	Threat Agent	Skill level	6 - Some technical skills	6.125	HIGH	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	5.125	MEDIUM	High
	[Threat] Attacker get the user credential in server using remotely execute code with ID/PW		Motive	4 - Possible reward				Loss of integrity	1 - Minimal slightly corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	5 - Minimal primary services interrupted, extensive secondary services interrupted			
			Group Size	6 - Authenticated users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	5 - Easy				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	7 - Thousands of people			

Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#8- [Tampering]	Threat Agent	Skill level	3 - Network and programming skills	5.875	MEDIUM	Technical Impact	Loss of confidentiality	0 -	3.625	MEDIUM	Medium
	[Threat] Client does not send the platenumber if the user confirmed from server is tampered		Motive	4 - Possible reward				Loss of integrity	3 - Minimal seriously corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	3 - Minor effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	0 -			

Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#9- [Spoofing]	Threat Agent	Skill level	3 - Network and programming skills	5.875	MEDIUM	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	5.25	MEDIUM	Medium
	[Threat] Client send the platenumber to Attacker's Server. Client does not receive the retrived vehicle informations.		Motive	4 - Possible reward				Loss of integrity	0 -			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	7 - Thousands of people			

	Threat#10- [Repudiation]	Threat Agent	Skill level	3 - Network and programming skills	5.875	MEDIUM	Technical Impact	Loss of confidentiality	0 -	3.75	MEDIUM	Medium
	[Threat] Server repudiates even if client send the platenumber		Motive	4 - Possible reward				Loss of integrity	0 -			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	0 -			

Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#11- [Information Disclosure]	Threat Agent	Skill level	4 - Advanced computer user	6.125	HIGH	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	5.875	MEDIUM	High
	[Threat] Attacker get the vehicle information in server using remotely execute code with platenumber		Motive	7 -				Loss of integrity	1 - Minimal slightly corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	5 - Minimal primary services interrupted, extensive secondary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	9 - Completely anonymous			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	4 - Hidden				Non-compliance	7 - High profile violation			
			Intrusion detection	8 - Logged without review				Privacy violation	9 - Millions of people			

Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#12- [Spoofing]	Threat Agent	Skill level	3 - Network and programming skills	5.75	MEDIUM	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	5.25	MEDIUM	Medium
	[Threat] Attacker receive vehicle information from server when fake client send the platenumber to server		Motive	4 - Possible reward				Loss of integrity	0 -			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	5 -			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	4 -				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	7 - Thousands of people			

Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#16- [Information Disclosure]	Threat Agent	Skill level	3 - Network and programming skills	5.375	MEDIUM	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	4.5	MEDIUM	Medium
	[Threat] Attacker could sniff vehicle detail informaiton from server on connection		Motive	4 - Possible reward				Loss of integrity	0 -			
			Opportunity	7 - Some access or resources required				Loss of availability	1 - Minimal secondary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	3 - Difficult			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	7 - Thousands of people			
Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#17- [Information Disclosure]	Threat Agent	Skill level	3 - Network and programming skills	5.375	MEDIUM	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	4.5	MEDIUM	Medium
	[Threat] Attacker could sniff the email. Then complete the authentification.		Motive	4 - Possible reward				Loss of integrity	0 -			
			Opportunity	7 - Some access or resources required				Loss of availability	1 - Minimal secondary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	3 - Difficult			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	7 - Thousands of people			
Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#18- [Tampering]	Threat Agent	Skill level	3 - Network and programming skills	5.875	MEDIUM	Technical Impact	Loss of confidentiality	2 - Minimal non-sensitive data disclosed	5.25	MEDIUM	Medium
	[Threat] Server doesn't 2-factor authenficatation to user normally if attacker tampers the OTP.		Motive	4 - Possible reward				Loss of integrity	7 - Extensive seriously corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	9 - All services completely lost			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	1 -			
	Threat#13- [Tampering]	Threat Agent	Skill level	3 - Network and programming skills	5.875	MEDIUM	Technical Impact	Loss of confidentiality	4 - Minimal critical data disclosed, extensive non-sensitive data disclosed	5.25	MEDIUM	Medium
	[Threat] Client receive the wrong vehicle information if the platnumber is tampered		Motive	4 - Possible reward				Loss of integrity	7 - Extensive seriously corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	1 -			
Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#14- [Tampering]	Threat Agent	Skill level	3 - Network and programming skills	5.875	MEDIUM	Technical Impact	Loss of confidentiality	4 - Minimal critical data disclosed, extensive non-sensitive data disclosed	5.25	MEDIUM	Medium
	[Threat] Client receive the wrong vehicle information if the retrived vehicle informations is tampered		Motive	4 - Possible reward				Loss of integrity	7 - Extensive seriously corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	1 -			
Inteface	Threat Group	Factors for Estimating Likelihood					Factors for Estimating Impact					Overall Risk Severity
		Estimating Factors	Factors	Range	Likelihood Score	Likelihood Severity	Estimating Factors	Factors	Range	Impact Score	Impact Severity	
	Threat#15- [Repudiation]	Threat Agent	Skill level	3 - Network and programming skills	5.875	MEDIUM	Technical Impact	Loss of confidentiality	0 -	3.75	MEDIUM	Medium
	[Threat] Client repudiates even if server send the vehicle information		Motive	4 - Possible reward				Loss of integrity	0 -			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	7 - Significant effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	0 -			

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

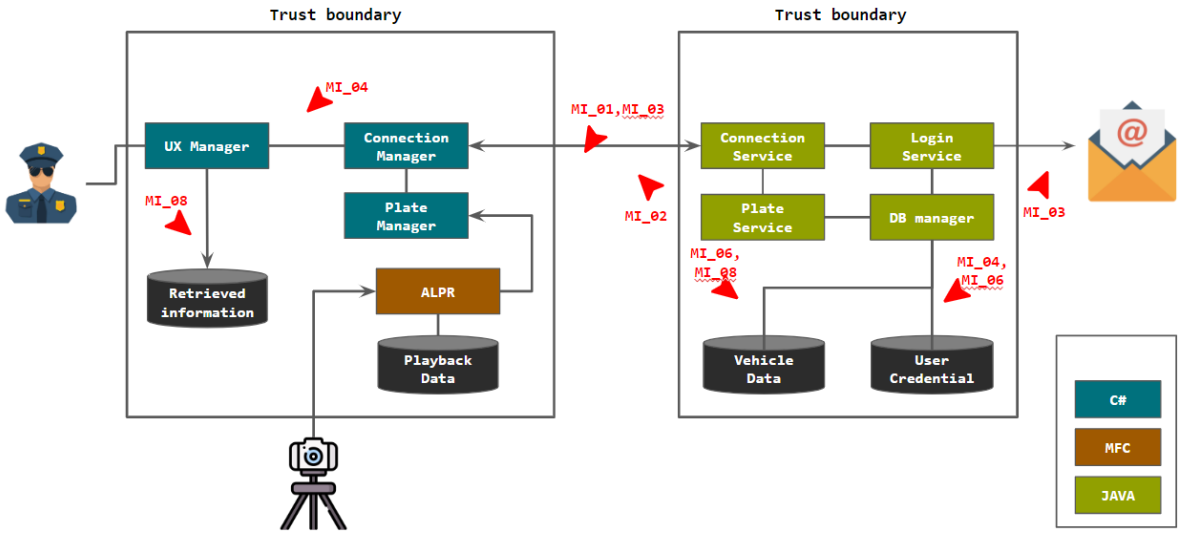
We describe the mitigations to satisfy the security requirements.

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.



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.

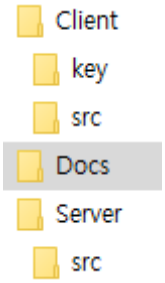
The structure of JWT is:

- 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

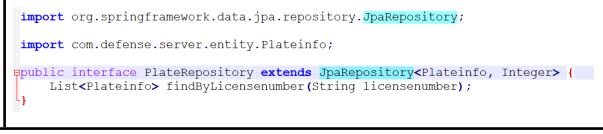
10.1. Implementation

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

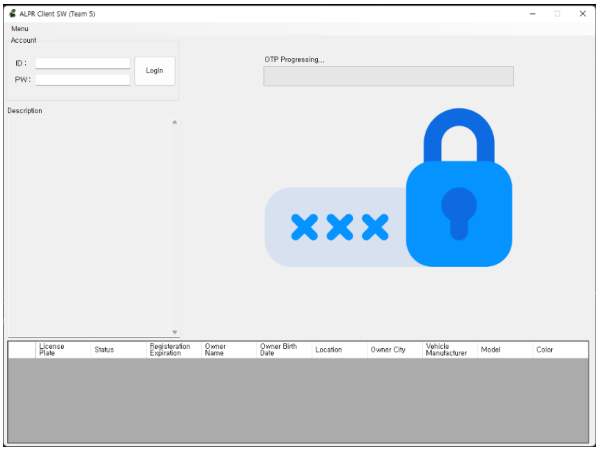
Client	- Execution environment 1. Visual Studio 2022 Community - .NET 4.7.2 Installation 2. Change the window display setting DPI to 100% 3. Build as below through ALPR_Client folder 1) Double-click the client key file to install as default (File : Client.pfx, Password: qwe123.. , Location: ..\ALPR_Client\Client_Key) 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#########

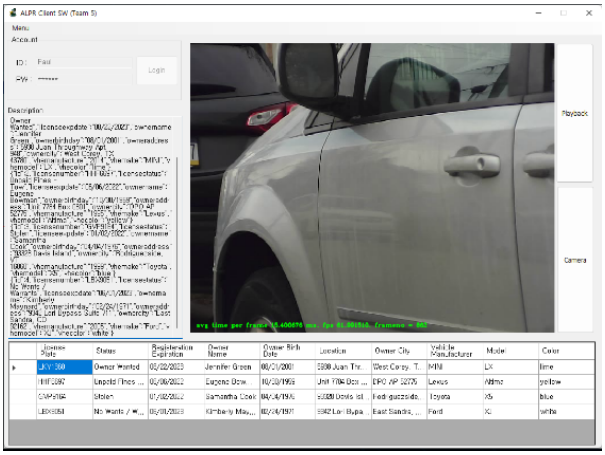
	The main DB file name of the server is <project>/T5Defense_Server/local.mv.db
	In the given DB file, license information identified by plate number and 1 user account are set by default. In our project, there is no sign-up process and it is assumed that the users to be authenticated are registered in 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.



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



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

Phase 2:
Security Analysis of Classmate System

We reviewed Team 4's output, identified security goals and assets, and figured out attack surfaces and found vulnerabilities through design reviews and code reviews. Then, vulnerabilities were assessed and classified. In addition, a method to attack each vulnerability was derived and actually verified.

In software requirements, Server has to be implemented as a Web Server. However, team4 was implemented with the Console server.
So we started with an Internal-cooperator scenario because access to the Console Server is realistically very difficult.

Artifacts of Team 4

Github: https://github.com/S4Best/team4

We got a one user credential from team 4

- ID : SecurityPolice_006
- PW : !dlwormsS4Best
- OTP : Check the number in the OTP program that has already been run.

12. Analysis

12.1. Secure Requirement & Mitigation

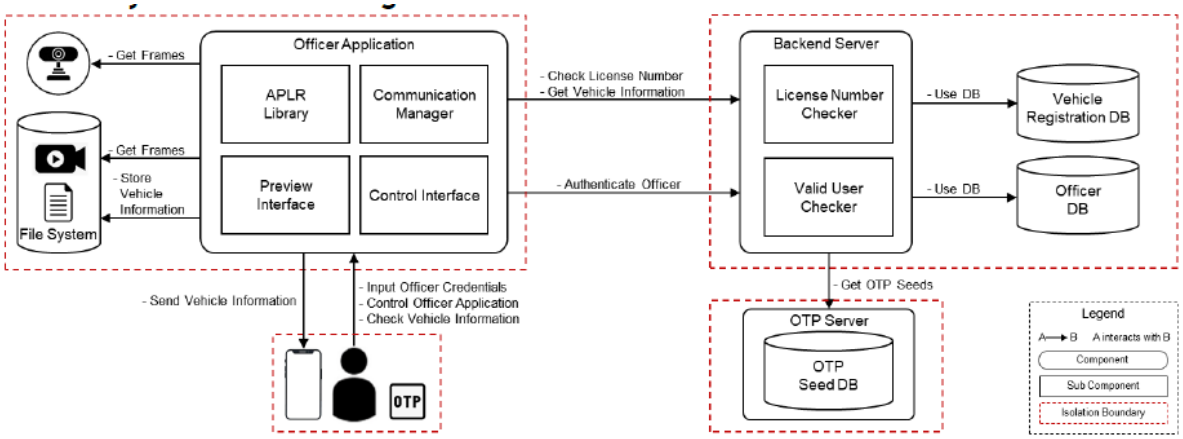
Reviewed the artifacts of Team 4. Extract the valuable data and attach it in following sections in order to identify the targets for assessment.

SR ID	Security Requirement	Mitigation
SR-S1	DB File should be protected and managed using ACL. The management policy for DB file should be applicable to Windows 10, Access, read, and write rights for DB files are limited to the backend server.	Vehicle Registration DB File ACL Management User DB File ACL Management
SR-S2	It is necessary to apply an appropriate authentication technique and go through the authentication process in order to guarantee the integrity and validity of the DB file. This can be done by MAC appended to file contents.	Vehicle Registration DB File Authentication User DB File Authentication
SR-S3	The user authentication should be applied with user ID and PW to check the validity of access rights to DB files.	Vehicle Registration DB User/Password User DB User/Password

SR-S4	Limit the number of simultaneous client connections to mitigate excessive access to the user DB. The maximum number of clients connected concurrently is defined as < N.	Limit the number of simultaneous client connections
SR-S5	Rate limit should be applied to the log-in process to limit the maximum number of errors in order to mitigate excessive access to the user DB,	Rate Limit : number of log-in errors
SR-C1	Playback files should be protected by managing ACL. Management policies for stored files should be applicable to Windows 10, Access and modification rights for saved files are limited to client applications.	ACL management regarding to play-back file
SR-C2	Vehicle registration data files should be saved after being encrypted. This is to prevent information leakage due to data sniffing by attackers.	Vehicle Registration Data File Encryption
SR-C3	Vehicle registration data files should be protected by managing ACL. Management policies for stored files should be applicable to Windows 10, Access and modification rights for saved files are limited to client applications.	ACL management regarding to vehicle registration data file
SR-C4	Rate limit should be applied to the save request to limit the maximum number of consecutive requests in order to mitigate excessive access to the file system.	Rate Limit : number of consecutive save request
SR-C5	Two factor authentication should be applied to log-in procedures.	Two Factor Authentication
SR-C6	Validity check for officer's input string should be done in the proper manner.	Input Validation : Input String
SR-N1	Communication between client application and backend server must be protected through TLS.	TLS
SR-N2	The communication process between client application and backend server should be logged for non-repudiation.	Logging
SR-N3	The connection between the client application and the backend server should go through a mutual authentication process by digital signature.	Digital Signature : Mutual Authentication
SR-N4	Communication between client application and backend server should be done by verifying the promised protocol and payload format.	Input Validation : Payload and Protocol
SR-N5	Client and Server System should protect DoS Attack using Firewall	Firewall : Syn-Flooding, ICMP

12.2. Design

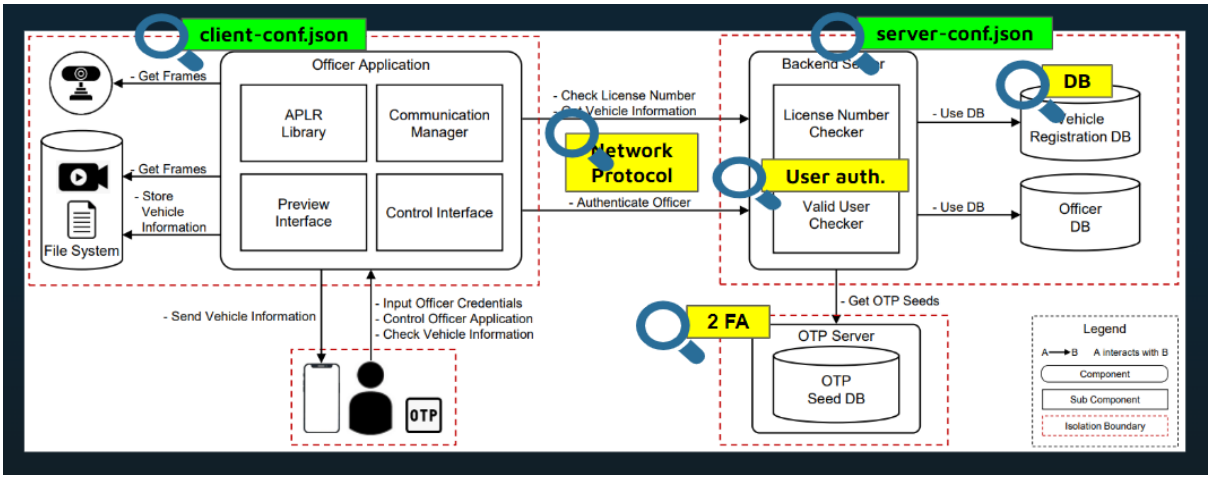
This is the overall architecture that is received from team 4.



we know after analysis design and code structure,

Used language: C++ for both client and server.
Configuration file: client-conf.json, server-conf.json
2 factor authentication method: OTP, OTP is a separate exe file.
DB: DB are separate files.

We decide to start analyze about below 6 items



- client-conf.json: Could we make a spoof app using this? .
- server-conf.json: Could we change any configuration?
- Network protocol: Could we sniff any information?
- User authentication: Could we find any backdoor or hole?
- DB: Could we access the DB?
- 2 Factor Authentication: Could we find any fault in the logic?

12.3. Runtime Analysis

12.3.1. nmap

We are checking the server with **nmap** for Penetration Testing

We have checking the server with nmap for Penetration Testing
Firewall ON, the Service name of each port was displayed as “tcpwrapped”.
So we have to retry it after Firewall OFF. After that we could check Service name of each port.
But 2222 port (server port) was still displayed as “tcpwrapped”
As a result, we can not find a metasploit database for an exploit related to "tcpwrapped".

Firewall ON

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV 10.58.3.229
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-13 03:04 EDT
Nmap scan report for 10.58.3.229
Host is up (0.0068s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT     STATE SERVICE    VERSION
80/tcp   open  tcpwrapped
135/tcp  open  tcpwrapped
139/tcp  open  tcpwrapped
445/tcp  open  tcpwrapped
2222/tcp open  tcpwrapped

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.70 seconds
```

Firewall OFF

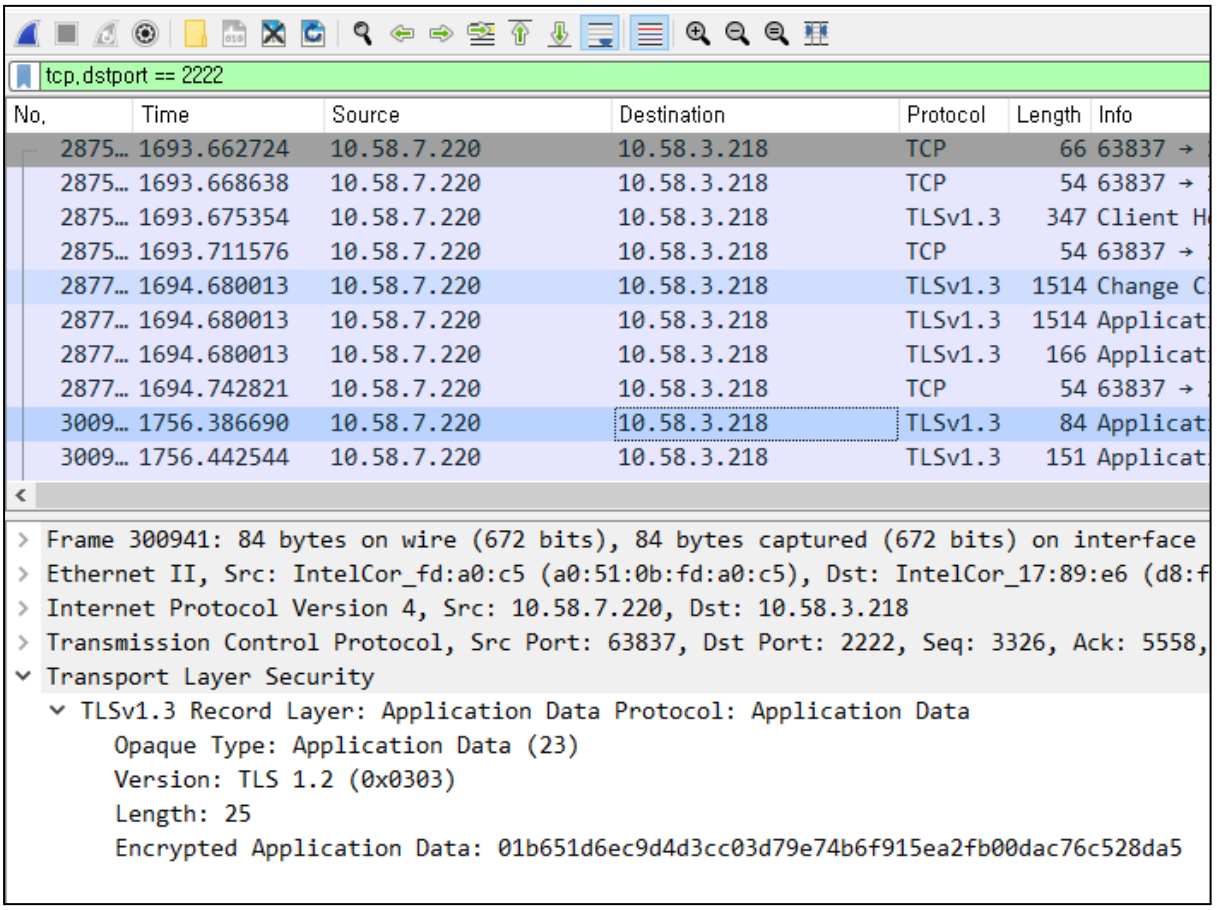
```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV 10.58.3.229
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-13 03:20 EDT
Nmap scan report for 10.58.3.229
Host is up (0.0087s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT     STATE SERVICE      VERSION
80/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: LGE)
2222/tcp open  tcpwrapped
Service Info: Host: MGKRD10-NA104GB; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

```
Nmap done: 1 IP address (1 host up) scanned in 17.88 seconds
```

12.3.2. Wireshark

We have checked the TCP packets of port 2222 with wireshark.

All data was changed to HASH because TLS1.3 was used.



12.4. Static Analysis

We used the Flawfinder and the Coverity Tool to identify the Static Analysis.
By performing Static Analysis, we wanted to find a vulnerability to attempting Exploit.

12.4.1. Flawfinder

Below are the results of the FlawFinder Tool.
All arrays with fixed sizes have been detected. However, validated values were being used in arrays.
In conclusion, it was judged as a false positive.

Stats from Flawfinder	Total	Open	closed	False Positive
# of vulnerabilities (client)	35	0	0	35
# of vulnerabilities (server)	15	0	0	15
# of vulnerabilities (common)	11	0	0	11

12.4.2. Coverity tool

Below are the results of Coverity Tool.

Stats from Coverity	Total	Open	closed	False Positive
# of vulnerabilities (client)	83	4	0	79
# of vulnerabilities (server)	8	1	0	7
# of vulnerabilities (common)	15	0	0	15

We found two critical issues through the Coverage Tool.

1) DATA RACE CONDITION -
   Data RACE Condition was detected in the Client's Thread.
   So we tried multi-access to confirm this, and contrary to expectations, we confirmed that the

client randomly disconnected from the server.

ClientMachine.cpp

```
261 void CClientMachine::client_state_machine_thread()
262 {
    1. Condition true /* 1 */, taking true branch.
263     while (1)
264     {
265         std::unique_lock<std::mutex> lk(cm->mtx_main);
266         cm->cv_main.wait(lk, [&] { return cm->stateChanged || cm->exitFlag; });
267
    2. Condition cm->stateChanged, taking true branch.
268         if (cm->stateChanged)
269         {
270             lgc_state_e changed_st = cm->getCliStatus();
    CID 37944 (#1 of 1): Data race condition (MISSING_LOCK)
    3. missing_lock: Accessing cm->stateChanged without holding lock std::unique_lock<std::mutex>._Pmtx. accesses strongly imply that it is necessary).
271             cm->stateChanged = false;
272             lk.unlock();
273
```

2) DEVIDED_BY_ZERO

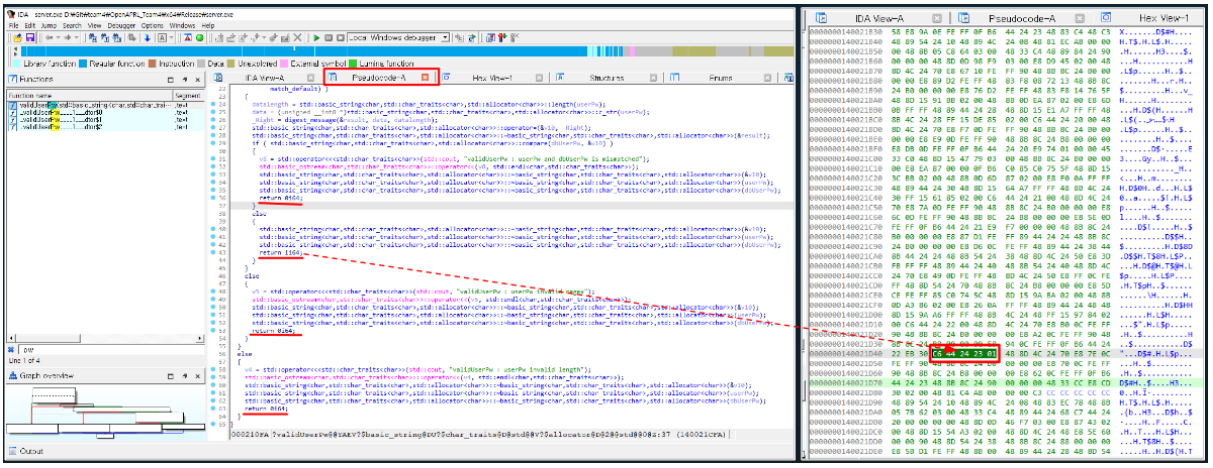
Team4's Code has a Backdoor ID that avoids 2FA.

DEVIDED_BY_ZERO was detected in the corresponding Logic, and when the Backdoor ID and Password 0 are actually entered, the server dies.

```
    23. Condition !strcmp(accountReq.userId.c_str(), "39354436618376475 79") , taking true branch.
410             else if (!strcmp(accountReq.userId.c_str(), "39354436618376475 79")) {
411                 string pw = accountReq.password.substr(0, accountReq.password.find('A'));
    24. zero_return: Function call std::stoull(pw, NULL, 10) returns 0. [show details]
    25. assignment: Assigning: numdiv = std::stoull(pw, NULL, 10). The value of numdiv is now 0.
412                 unsigned long long numdiv = std::stoull(pw);
413                 unsigned long long numorg = std::stoull(accountReq.userId);
414                 unsigned long long numdived = 0;
415
    26. Condition ConfData->debug, taking true branch.
416                 if (ConfData->debug) {
417                     cout << "id : " << numorg << std::endl;
418                     cout << "password : " << numdiv << std::endl;
419                 }
420
    CID 38661 (#1 of 1): Division or modulo by zero (DIVIDE_BY_ZERO)
    27. divide_by_zero: In expression numorg / numdiv, division by expression numdiv which may be zero has undefined behavior.
421                 numdived = numorg / numdiv;
422                 if (!(numorg == (numdived * numdiv)) ||
423                     numdived == 1 || numdived == numorg)
424                 {
```

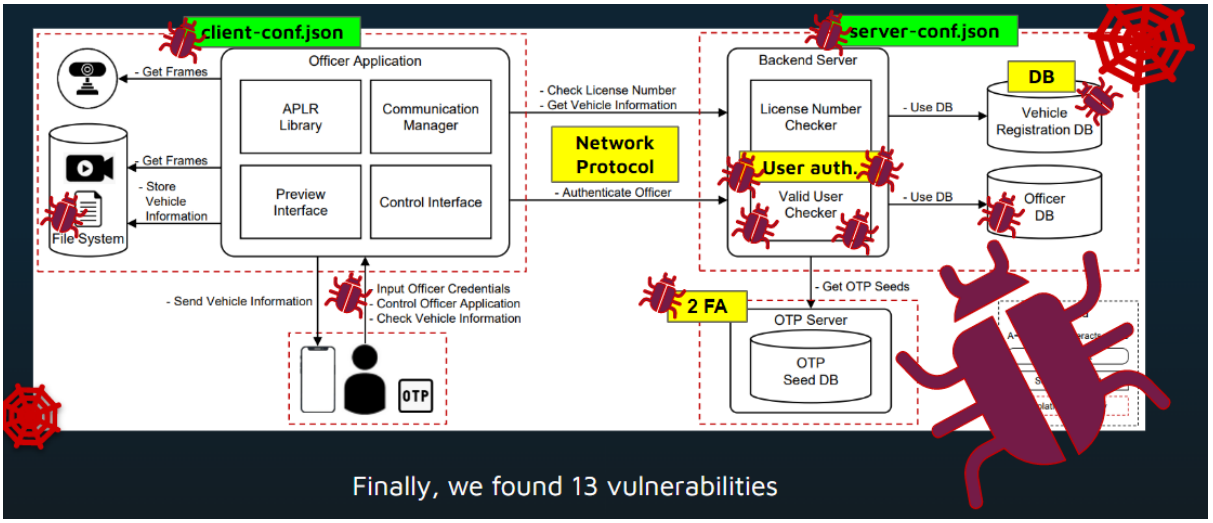
12.4.3. IDA

내용 내용



13. Exposed Vulnerabilities

There are vulnerabilities we've found. We marked them to diagram.



These vulnerabilities were found in several components, and especially, we focused on data encryption and authentication.

ID	Vulnerability	Approach	CIA	Impact
V01	Existence of BackDoor	[Code Review]	[CONFIDENTIALITY]	[HIGH]
V02	Weak passwords that can be easily exposed	[Tinkering]	[CONFIDENTIALITY]	[MEDIUM]
V03	Insecure logic is used and it is so dangerous to feed exploitation.	[Code Review]	[CONFIDENTIALITY]	[MEDIUM]
V04	Divided by Zero	[Static Analysis]	[CONFIDENTIALITY]	[MEDIUM]
V05	Authentication could be skipped in User Credential(PW) and 2FA(OTP) on server side through manipulating memory with the IDA tool.	[Reverse Engineering]	[INTEGRITY] [CONFIDENTIALITY]	[MEDIUM]
V06	Tempering configuration file(server-conf.json)	[Code Review], [Design Review]	[CONFIDENTIALITY]	[MEDIUM]
V07	Private key and encryption key files are exposed in plaintext.	[Code Review], [Design Review]	[CONFIDENTIALITY]	[MEDIUM]
V08	Retrieved information is exposed in plain text	[Code Review], [Design Review]	[CONFIDENTIALITY]	[LOW]
V09	The server prints sensitive data to the console	[Code Review]	[CONFIDENTIALITY]	[LOW]
V10	Password input via the keyboard is not protected.	[Code Review], [Design	[CONFIDENTIALITY]	[MEDIUM]

		Review]		
V11	Possible MITM attack using certificate change	[Reverse Engineering]	[INTEGRITY]	[MEDIUM]
V12	Tampering configuration file(client-conf.json)	[Code Review], [Design Review]	[CONFIDENTIALITY]	[MEDIUM]
V13	Tampering server DB	[Code Review], [Design Review]	[INTEGRITY]	[MEDIUM]

13.1. Criteria

13.1.1. Location

Location of the part that is related to the vulnerability.

13.1.2. Approach

Approaches to find vulnerabilities.

[Code Review]	Method for finding problems through review of source code.
[Design Review]	Method for finding problems through review of design documentation.
[Reverse Engineering]	Method to obtain the original source by reversely analyzing the program
[Social Engineering]	Method psychological manipulation of people into performing actions or divulging confidential information.
[SNIFFING]	Sniffing packets over the network
[SPOOFING]	So-called, man in the middle attack
[TAMPERING]	Modifying system components for a purpose
[Tinkering]	To try something driven by a whim, imagination, or curiosity.

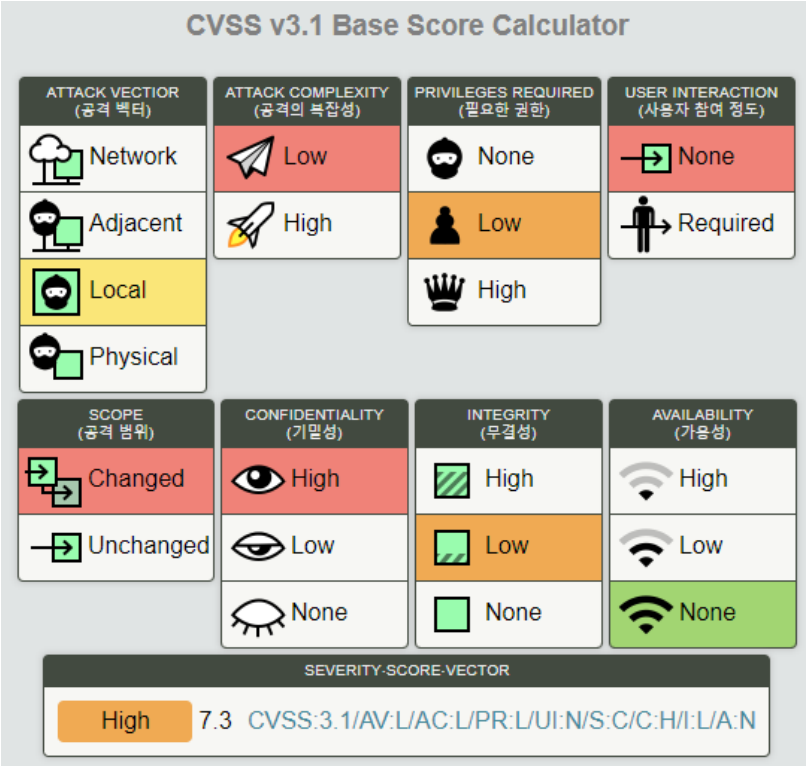
13.1.3. CIA

[CONFIDENTIALITY]	Compromises confidentiality
[INTEGRITY]	Compromises integrity
[AVAILABILITY]	Compromises availability

13.1.4. Impact

Using CVSS v3.1 base score calculator to determine the impact.

[CRITICAL] [HIGH] [MEDIUM] [LOW]	Higher than score 9.0 Higher than score 7.0 Higher than score 4.0 Lower than score 4.0



13.2. Details

13.2.1. V01 -Existence of backdoor in code

Summary	Success Log-In by insert specific ID/PW(BackDoor Log-In)

Location [server.ccp #409] Approach [Code Review] CIA [CONFIDENTIALITY] Impact [HIGH]
Description
System BackDoor Code: System BackDoor Code is found through code review. If a specific value is entered as ID/PW regardless of ID/PW, system log-in is possible without OTP knowledge. -> Login can be successful If input 3935443661837647579 in ID and the divisible value of 3935443661837647579(1324523323 or 2971215073 + Aa!) in PW with any OTP(6 numbers) -> However, PW/OTP must be made according to the PW/OTP rules. (Example: PW: 1324523323Aa! - [number + capital letter + small letter + special characters], OTP: 111111 - 6 numbers)
Perceived impact
If a specific value is entered as ID/PW, system LogIn is possible without knowing ID/PW/OTP.
Recommended Mitigations
Delete BackDoor code.
Reproduce Step(PoC)

[Precondition]

0. The attacker obtained the server's code through reverse engineering, etc.

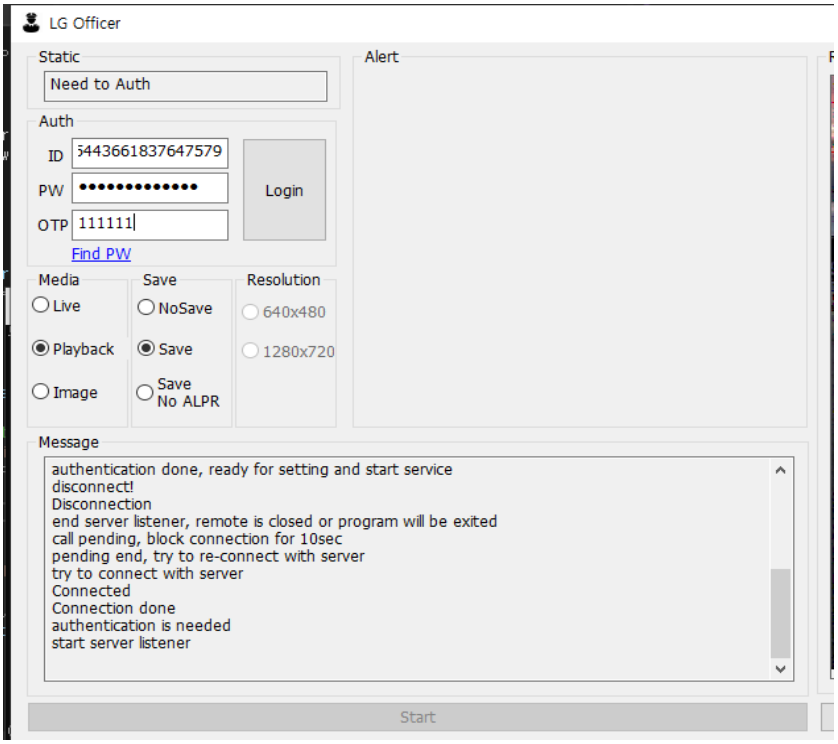
1. Confirm Log-In BackDoor value in sever.cpp “3935443661837647579”

```
else if (!strcmp(accountReq.userId.c_str(), "3935443661837647579")) {
    string pw = accountReq.password.substr(0, accountReq.password.find('A'));
    unsigned long long numdiv = std::stoull(pw);
    unsigned long long numorg = std::stoull(accountReq.userId);
    unsigned long long numdived = 0;

    if (ConfData->debug) {
        cout << "id : " << numorg << std::endl;
        cout << "password : " << numdiv << std::endl;
    }

    numdived = numorg / numdiv;
    if (!(numorg == (numdived * numdiv)) ||
        numdived == 1 || numdived == numorg)
    {
        // invalid user information so, reject code is needed
        std::cout << "log-in : invalid user information: " << std::endl;
        responseLoginResult(ConPort, "login_400_nok");
        goto free_resource;
    }

    if (ConfData->debug) {
        cout << "log-in : user login success: " + accountReq.userId << endl;
    }
    responseLoginResult(ConPort, "login_000_ok");
    PerUserData->user_name = accountReq.userId.c_str();
    PerUserData->state = 1;
    cur_connection++;
}
else {
    // invalid user information so, reject code is needed
```

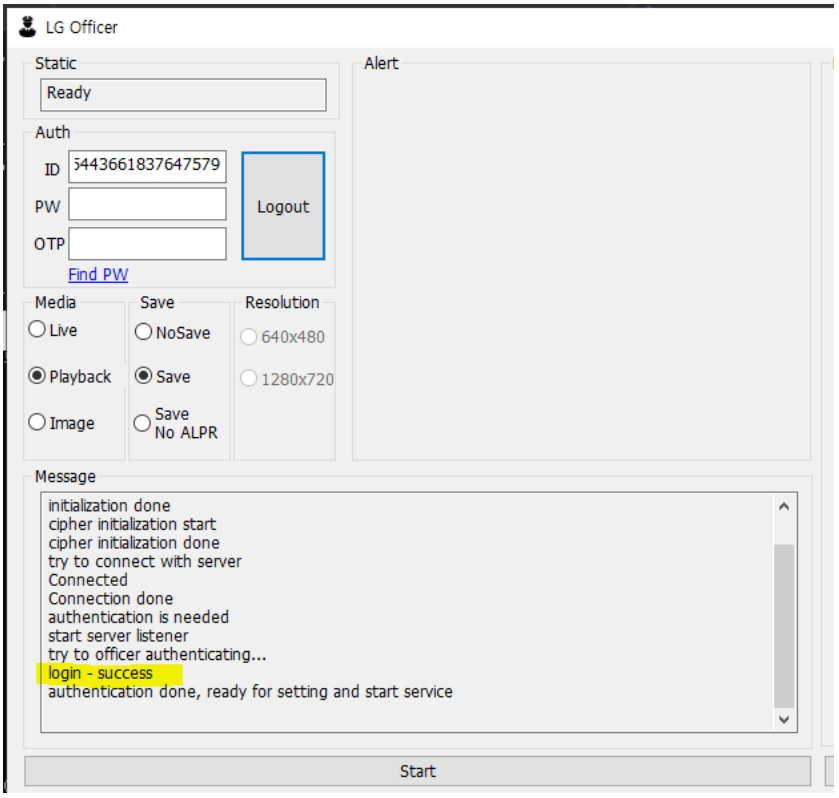
2. Find the divisible value of 3935443661837647579 to confirm P/W.
-> P/W = 1324523323 or 2971215073

3. Enter ID and PW/OTP according to the making rules.
ID(3935443661837647579), PW(1324523323 or 2971215073 + Aa!), OTP(111111)

4. Try Log-In



5. Confirm successful Log-In



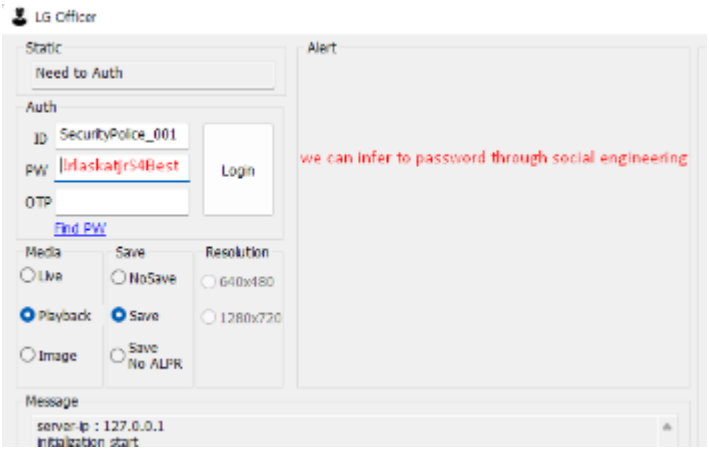
6. Finally we have successfully Log-In.

13.2.2. V02 - Infer to password of each account easily

Summary	Infer to password through Social Engineering methodology and success to authentication
Location [ID/PW Input UI] Approach [Tinkering] CIA [CONFIDENTIALITY] Impact [MEDIUM]	
Description	
userId = "SecurityPolice_006" password = "**4979f8a2e2bf4de925abca4244ce52328d4ee160851d8c7f7149375da6c14407**" otpKey = "SecurityPolice_006ZLgeZ4BestX006" **[Infer to password through Social Engineering methodology and success to authentication]** SecurityPolice_001 / !rlaskatjrS4Best /김남석 SecurityPolice_002 / !rlatjddmsS4Best / 김성은 SecurityPolice_003 / !rlacndfoS4Best / 김충래 SecurityPolice_004 / !djawldndS4Best / 엄지웅 SecurityPolice_005 / !chlwjdghS4Best / 최정호 **SecurityPolice_006 / !dlwormsS4Best /** 이재근 ex) rlaskatjrS4 -> Korean typing(김남석) = name of a member Condition : DB is not exposed and we can't modify and access to DB Through Social Engineering methodology, It is easy to infer their password, il is very dangerous in this password related personal information Consequently user account was stolen easily and misused it in the near future	
Perceived impact	
It is easy to infer their password, it is very dangerous in this password. Consequently user account was stolen easily and misused it in the near future	
Recommended Mitigations	
Make password not including related personal information not to infer password.	
Reproduce Step(PoC)	

we inferred to password through social engineering
As a result of inference, we can input the candidate password and check it whether login success or not

1. In case of target server environment, we can't access to DB and modify it because server is made of console server and not designed to add account in DB
2. We can't see credential information in DB however only know one person information
3. We knew name of team4 members so we tried to social engineering methodology
4. We have entered their password as his name related
5. As a result, we checked to match the password
6. It is easy to infer their password, it is very dangerous in this password



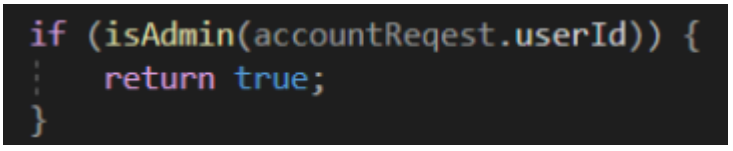
13.2.3. V03 - Included dangerous logic when checked authentication

Summary	Insecure logic is used and it is so dangerous to feed exploitation
Location	[server.cpp #230]
Approach	[Code Review]
CIA	[CONFIDENTIALITY]
Impact	[MEDIUM]
Description	

There is insecure logic and dangerous strings in code.

1) admin is exposed as string in code
   - admin ID : "SecurityPolice_Admin" --> included in code

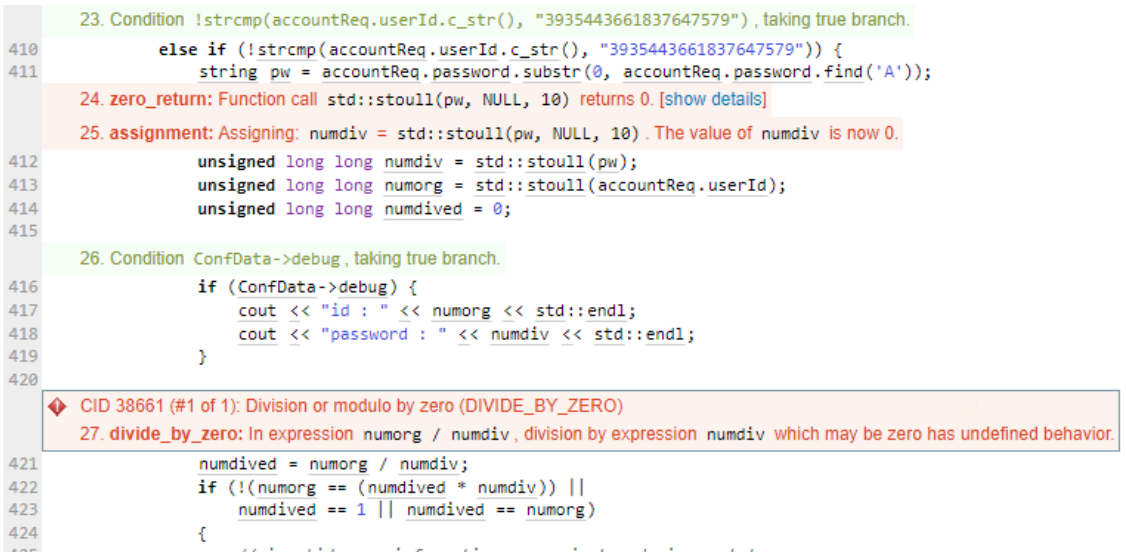
2) Insecure logic is used and it is so dangerous to feed exploitation. code snippet: if (isAdmin(accountReqest.userId)) { return true; }
Perceived impact
Insecure code may be fed to exploit and success to login without normal authentication.
Recommended Mitigations
Delete unnecessary and insecure code.
Reproduce Step(PoC)

we can see the dangerous code and it may be used to exploit with reverse engineering.
 	
1. In case of target server environment, we can't access to DB and modify it because server is made of console server and not designed to add account in DB
2. We can't see credential information in DB however only know one person information
3. We knew name of team4 members so we tried to social engineering methodology
4. We have entered their password as his name related
5. As a result, we checked to match the password
6. It is easy to infer their password, it is very dangerous in this password
 	
 
`if (isAdmin(accountReqest.userId)) {`
`  return true;`
`}`

13.2.4. V04 - Service attack by “Divided by Zero”

Summary	Server crashed when logged in with password “Zero Value’’
Location	[server.ccp #421]
Approach	[Static Analysis]
CIA	[AVAILABILITY]
Impact	[MEDIUM]
Description	

We can find this issue throw the Coverity Tool(Static Analysis)



So if we enter the ID(BackDoor Code) and PW(Zero valu), that occurs Divide by Zero.

Perceived impact

1. Denial of Service. Server terminated.

Recommended Mitigations

1. Delete BackDoor code.

Reproduce Step(PoC)

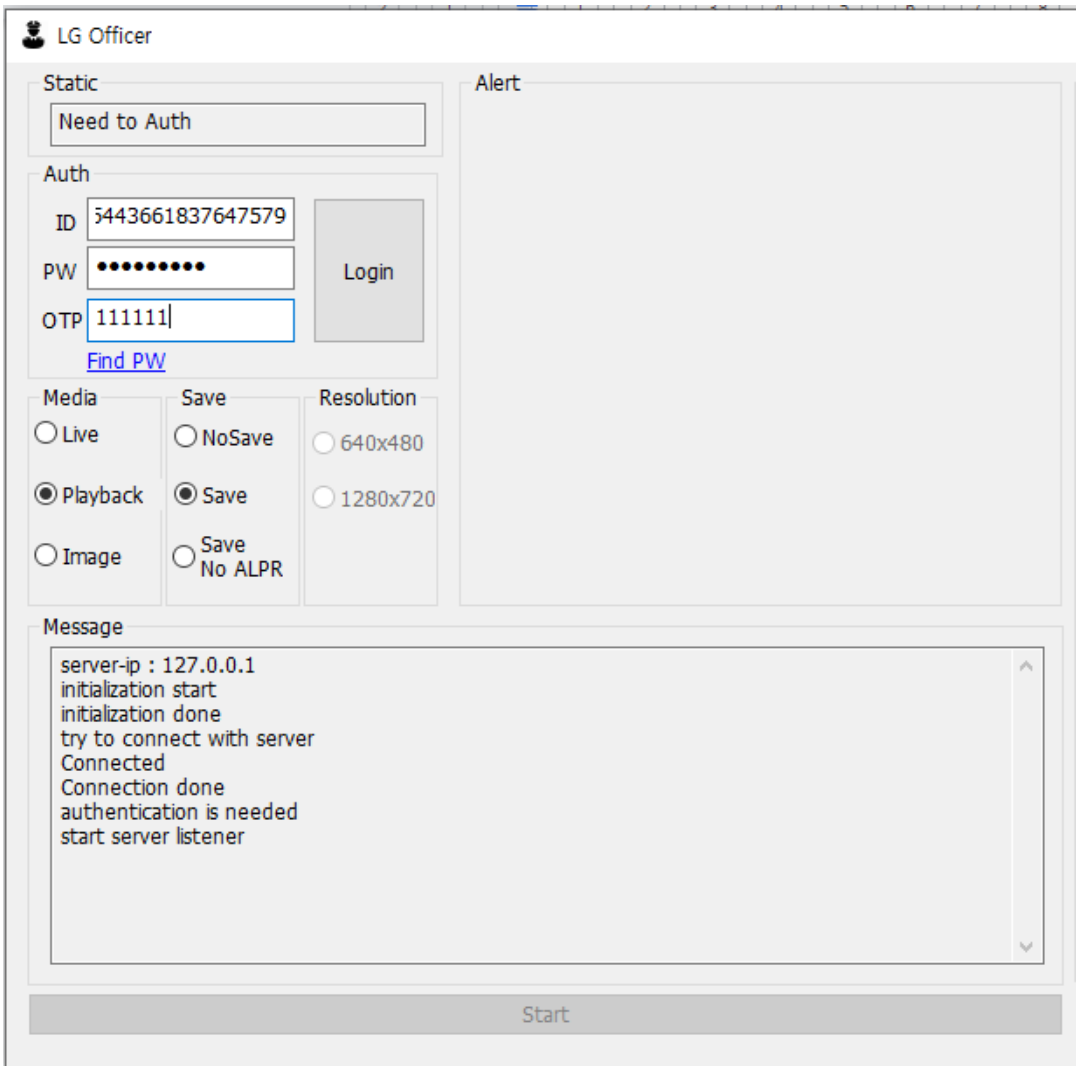
[Precondition]

The attacker obtained the server's code through reverse engineering, etc.

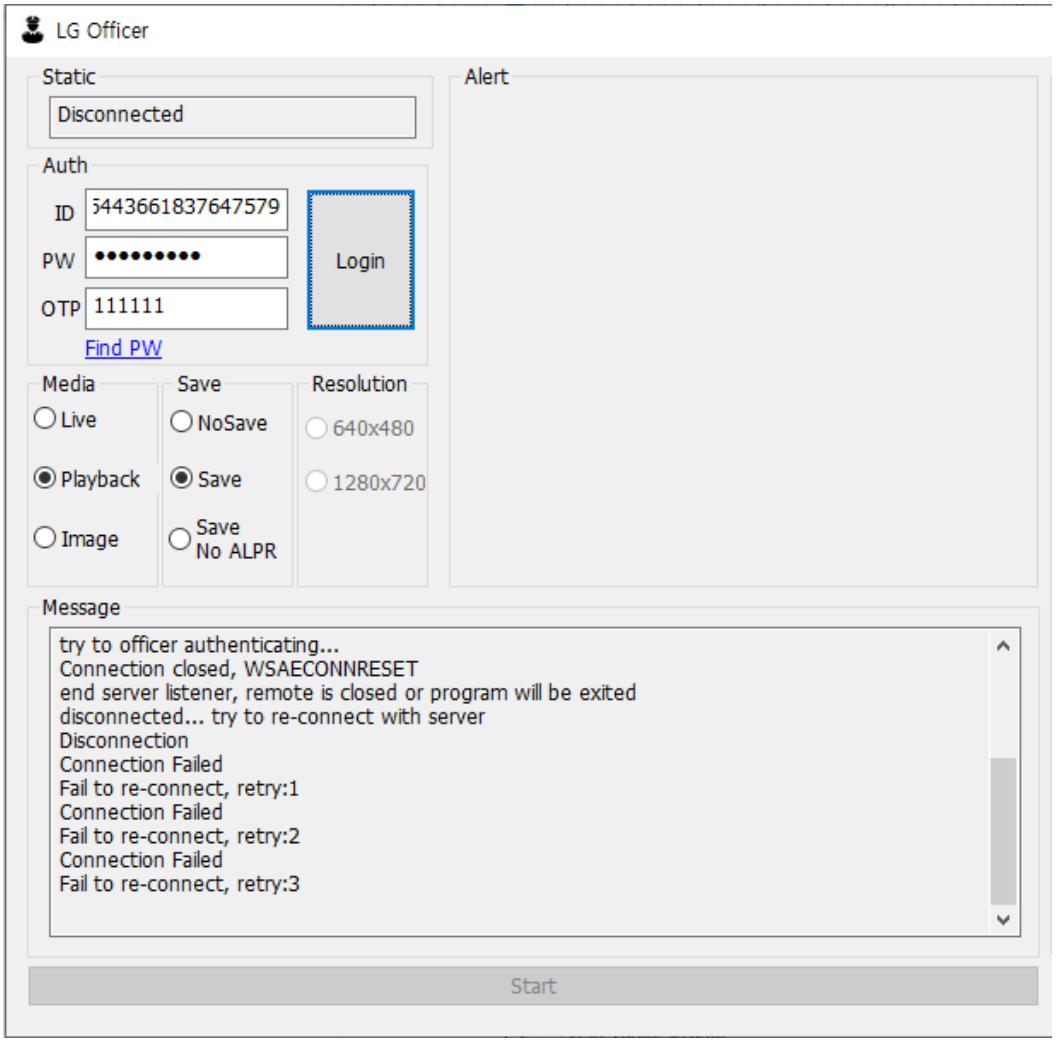
1. Confirm Log-In BackDoor value in sever.cpp “3935443661837647579”

2. Enter ID and PW(with Zero value)/OTP according to the making rules.

ID(3935443661837647579), PW(000000 + Aa!), OTP(111111)



3. Try to Log-In



4. Server CRASHED. Clients can not Log-In.

13.2.5. V05- Authentication could be skipped by manipulating binary

Summary	Authentication could be skipped in User Credential(PW) and 2FA(OTP) on server side through manipulating memory with IDA tool
Location Approach CIA Impact	[server.exe] [Reverse Engineering] [CONFIDENTIALITY], [INTEGRITY] [MEDIUM]

Description
Authentication could be skipped in User Credential(PW) and 2FA(OTP) on server side through manipulating memory with IDA tool. - precondition : PW and OTP values should entered according to the criteria(regex, length) - Each OTPGen matched in account is manipulated whatever you want.
Perceived impact
Authentication (2FA, Credential) could be skipped without progress on normal authentication Set the return value as TRUE processed after some condition by IDA tool.
Recommended Mitigations
Apply the digital sign for server.exe file. if occur corrupt memory, we can recognize failure to integrity
Reproduce Step(PoC)
Program could be manipulated through IDA First we find out related authentication code and then change the return value to TRUE Server can't validate authentication correctly and passed it. 0. we received server binary file by insider threat 1. open the server.exe with IDA tool 2. changed memory data related authentication return address 3. overwrite the server file

13.2.6. V06 - Tampering configuration file(server-conf.json)

Summary	Tampering configuration file(server-conf.json)
Location	[server.cpp], [server-conf.json]
Approach	[Code Review], [Design Review]
CIA	[CONFIDENTIALITY]
Impact	[MEDIUM]

Description
When the server starts, the configuration is performed by reading the server-conf.json file. When reading the server-conf.json file, there is no logic to detect the change. Therefore, if you enable debug by changing server-conf.json, user id, pw, and otp are output to the server log
Perceived impact
1. Attackers could gather user's ID, PW and OTP. There could be used another session. 2. sensitive information is revealed in log prin
Recommended Mitigations
Delete unnecessary and insecure code change the code not to use configuration file to debug
Reproduce Step(PoC)

After manipulating some configuration in file, we can see sensitive information in log printing
 	
Precondition: Attacker know the IP and PORT.
 	
1. insert a function to run(need to assembly code)
 	
{
 	
stream.open(conf_path);
 	
stream >> json_obj;
 	
json_obj["debug"] = TRUE;

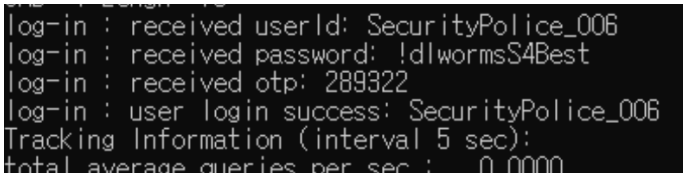
```
    stream.close();


    // write updated json object to file
    m_file.open(conf_path, std::ios::out);
    m_file << json_obj.toStyledString() << std::endl;
    m_file.close();
}
```

2. "debug": 1 is added to the server-conf.json file.

3. SW read and apply the server-conf.json to configuration

   (The debug is enabled)

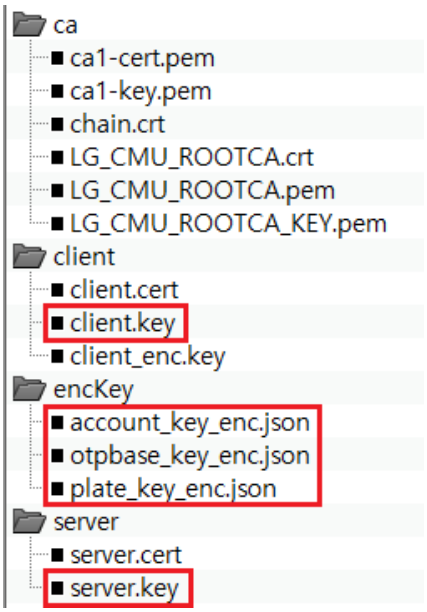
4. Get the user ID, PW and OTP from the log when the user inserts.



13.2.7. V07 - Sensitive data is exposed in plain text

Summary	Private key and encryption key files are exposed in plaintext.
Location	[keys/client/client.key], [keys/server/server.key], [keys/encKey/*]
Approach	[Code Review], [Design Review]
CIA	[CONFIDENTIALITY]
Impact	[MEDIUM]
Description	
The client and server each store their keys locally for encryption and TLS. However, no protection is applied to these keys and they are exposed in plaintext. Private keys and encryption keys are stored in their respective local storages on the client and server. However, if the data in the storage is leaked to the outside due to administrator's negligence, hacking attack or malware, physical theft etc., The keys may be exposed in plaintext and may be misused, which can lead to serious security risks. * NOTE : This vulnerability was discovered at the time of initial submission by team 4. However, it has been fixed with an additional patch after the submission deadline.	
Perceived impact	
The client and server perform TLS communication and mutual authentication through their own certificate, private key, and trusted CA chain. If an attacker can steal the certificate and private key, they can try various attacks through spoofing. For example, the attacker can spoof the server by using the server's key they stole and they can collect the ID/password of officers from the client. The server is protecting the db using an encryption key. If an attacker does not know the encryption key, they cannot know the contents of the db, but if they know the key, they can decrypt db and collect information.	
Recommended Mitigations	
Important keys used for TLS communication and encryption should be stored in the HSM or encrypted.	
Reproduce Step(PoC)	

- I checked all the keys used on the client and server.



"client.key" is a client's private key for TLS and "server.key" is for server.

- checked client.key and found that the key is pem format with plaintext.

```
$ openssl rsa -in client.key -check
RSA key ok
```

client.key

```
1  -----BEGIN RSA PRIVATE KEY-----
2  MIIEpAIBAAKCAQEAsOsRX90FzcMzyUh3xoFTlKChiyb0PAL2G11qmgdAD3ugNUd+
3  v2HwDL8bj1Z4UgBaFGPFeAEUJwPHJ2ukI5jZioSMPadTt/KBAFjm7mdtkx4HKhoH
4  P78fxkwYxJusWyWsxlTNS9HQXUYn7w01O7oXDSunJH+zgUqv8LW10bSM8TFlAz7f
5  O+bjtJ7k8mcsos3aiY4p3RferBRhbwRTjRtVJHOpJqLx39XaVSdqZ6LoFsTnitW9
6  e9A2ZlPDmVbvXIsJb+5nxJhOcJnZBG1Q/46FCHDuHpilpuoY2qRz7E8K0q2YmkDK
7  rhpCm7fTArQ0noZBYmBFNlWNFTgXlCL67hJj4QIDAQABAoIBAH9WwquwR1Qh0y1m
8  lqdvQHxy3bNtt09WhlON+oOhAk/imJ6gti8ETCawiyKxh2rmS+/vHBvTEvE9++N4
9  y6y/gJz96H8b59s3lfhtBNLF6Q2CCIAaOet9Il61OQKSVg1I4cHjGd0fyIzKSUX7
10 yLwTpjHZoLVZoRdHbbmg4JxvCx8yTtER8mHvL15boOWn/8DUjivuAgSk2tYyCyc9
```

- found that server.key is plaintext also

```
$ openssl rsa -in server.key -check
RSA key ok
```

server.key

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAuBuWgFIQUivSXKNI2wvW6ORM7zY13nX/PHqdtOHo+QQivruq
m2pZqMU0y5hgSEYGQgOLqMzfQ+rYMmaVtPqa3YEMxl2yxYtmoFGpIsACQFT00iWp
ZDRWcuccTEzZbzkkjaivD/8FKqZOJbOdAJjdD/M89qtd+QQUBEyJsO0/5zUENN40
Ti7LIMdeSO1v8FF+7HrrhQ8dAi9P1s/XbuaozZuhMVpVmrn+HHKly3oemTMmbJ8X
ftCG/niEHJUWtaP5arXJqtUK3XY+rLH6LM7bg6wCa+LRdaVkv9GrDtK4g5slMxaB
3Tzvb0dpYVLjYnNdSeEvv+TOEbHlcp3evzYobwIDAQABAoIBAQCklW0Om84hVzGA
RSxNPaFKo0Uvl7vCeqPIkijH2ijl68+PyL60nvc2C82tsDNqiMbxXrAMH9H7qkbJ
j7GHj3zARayA7KRzOrJs4XOdMB8TGZ+e+bIbdTpPD9HZV94NMv09ci3F1AnvXaqA
w7e4H++HYb//F6UzhnTKfw2cUAuPiOwcCEPtn+RbFjqUmUoU/ma2fmyIE39fjYTJ
```

- found that although names of encryption keys are enc, they are actually made up of plaintext.

account_key_enc.json

```
{
    "encKey": "QF1YlKiYifyulWO7sLHZ6Wi+IASWcvS5pQgxewIDAQABAoIBADSQxr+a8Yhn9oDk"
}
```

otpbase_key_enc.json

```
{
    "encKey": "MIIEowIBAAKCAQEA1HunpXTYwIhoGZDwCYajUoMIa59SvarMv09RV5HYYfcwowRN"
}
```

plate_key_enc.json

```
{
    "encKey": "uRHCUXqNfFpFlTuLeqrUb9Qr7Qstk8cuG69A8YPRJeBPoYx6ICjYo9kFALScYNqh"
}
```

- If an attacker can access the local storage and collect the key files,the attacker can spoof the client or server and decrypt the db file. So it is a vulnerability.

13.2.8. V08 - Retrieved information is exposed in plain text

Summary	Retrieved information is exposed in plaintext.
Location	[alert.log]
Approach	[Code Review], [Design Review]
CIA	[CONFIDENTIALITY]
Impact	[LOW]
Description	

The client stores retrieved vehicle information to file. However, no protection is applied to these information and they are exposed in plaintext.

The retrieved vehicle information is stored in local storage on the client. However, if the data in the storage is leaked to the outside due to officer's negligence, hacking attack or malware, physical theft etc., the retrieved information may be exposed in plaintext and may be misused.

NOTE : This vulnerability was discovered at the time of initial submission by team 4. However, it has been fixed with an additional patch after the submission deadline.

Perceived impact

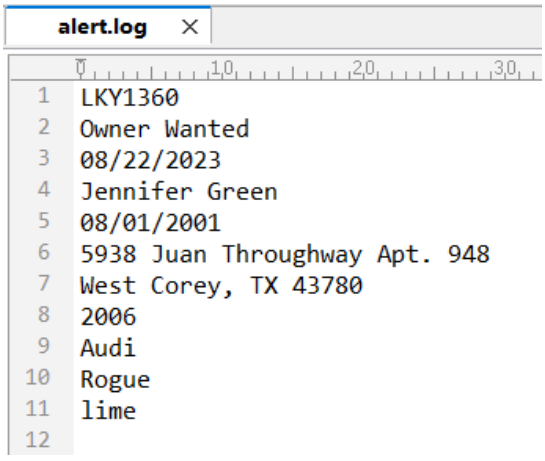
The retrieved vehicle information contains the vehicle that is stolen, the owner is wanted (criminal), or if it is a vehicle of interest (expired registration, unpaid tickets, owner is missing). Such sensitive personal information can be very useful to certain attackers.

Recommended Mitigations

The retrieved information should be encrypted.

Reproduce Step(PoC)

Retrieved vehicle information is stored as “alert.log” in the client's local storage.



It represents the user's sensitive information, including the plate number, name and address. Therefore, it can be a security threat if leaked.

13.2.9. V09 - The server prints sensitive data to the console

Summary	The server prints all request/response data including user password to the console.

Location [server.cpp #251, #331, #505]

Approach [Code Review]

CIA [CONFIDENTIALITY]

Impact [MEDIUM]

Description

The server prints officer's ID, password, otp and retrieved vehicle information to the console. If the attacker succeeded in checking the output of the server console (for example, malware is installed on the server that periodically takes screenshots and sends it to the attacker's server), the attacker could easily obtain confidential information.

NOTE : This vulnerability was discovered at the time of initial submission by team 4. However, it has been fixed with an additional patch after the submission deadline.

Perceived impact

Confidential information such as the officer's ID, password, and vehicle information may be leaked.

Recommended Mitigations

The server should not print any important information to the console.

Reproduce Step(PoC)

During source code review, I found that the server prints all request/response data including user password to the console.

server.cpp #251 : it prints response of log in

```
241  void responseLoginResult(TTcpConnectedPort* ConPort, string response) {
242      ssize_t result;
243      int sendlength = (int)(strlen(response.c_str()) + 1);
244      int SendMsgHdr[2];
245      SendMsgHdr[0] = ntohs(netshort: 1);
246      SendMsgHdr[1] = ntohs(netshort: sendlength);
247      if ((result = WriteDataTcp(TcpConnectedPort: ConPort, data: (unsigned char*)SendMsgHdr, length: sizeof(SendMsgHdr))) != sizeof(SendMsgHdr))
248          printf(_Format: "responseLoginResult : WriteDataTcp %d₩n", result);
249      if ((result = WriteDataTcp(TcpConnectedPort: ConPort, data: (unsigned char*)response.c_str(), sendlength)) != sendlength)
250          printf(_Format: "responseLoginResult : WriteDataTcp %d₩n", result);
251      printf(_Format: "sent ->%s₩n", (char*)response.c_str());
252  }
```

server.cpp #331 : it prints ALL outgoing responses

```
326          if (ReadDataTcp(TcpConnectedPort: ConPort, data: (unsigned char*)Payload, PayloadLength) != PayloadLength)
327          {
328              printf(_Format: "ReadDataTcp 2 error₩n");
329              goto free_resource;
330          }
331          printf(_Format: "Payload is : %s₩n", Payload);
```

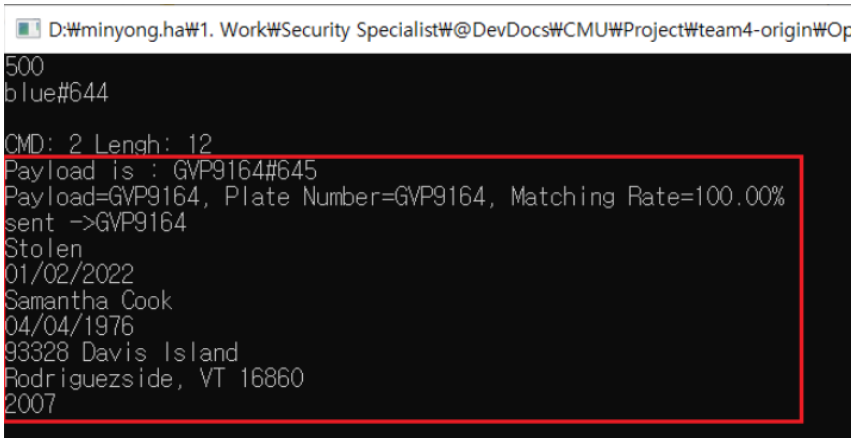
server.cpp #505 : it prints response that contains vehicle information

```
501                  if ((result = WriteDataTcp(TcpConnectedPort: ConPort, data: (unsigned char*)SendMsgHdr, length: sizeof(SendMsgHd
502                      printf(_Format: "WriteDataTcp %d₩n", result);
503                  if ((result = WriteDataTcp(TcpConnectedPort: ConPort, (unsigned char*)max_data.c_str(), sendlength)) != send
504                      printf(_Format: "WriteDataTcp %d₩n", result);
505                  printf(_Format: "sent ->%s₩n₩n", max_data.c_str());
506              }
```

found that user id, password, otp are printed to the console when user logs in.

```
D:₩minyong.ha₩1. Work₩Security Specialist₩@DevDocs₩CMU₩Project₩team4-origin₩OpenAPRL_Team4₩x64₩...
total average queries per sec :   0.0000
User[] average queies per sec :   0.0000
Tracking Information (interval 5 sec):
total average queries per sec :   0.0000
User[] average queies per sec :   0.0000
CMD: 1 Lengh: 75
Payload is : {"userId":"SecurityPolice_006","password":"!dlwormsS4Best","otp":"805296"}
log-in Payload raw: : {"userId":"SecurityPolice_006","password":"!dlwormsS4Best","otp":"805296"}
sent ->login_000_ok
Tracking Information (interval 5 sec):
total average queries per sec :   0.0000
User[SecurityPolice_006] average queies per sec :   0.0000
Tracking Information (interval 5 sec):
total average queries per sec :   0.0000
User[SecurityPolice_006] average queies per sec :   0.0000
Tracking Information (interval 5 sec):
```

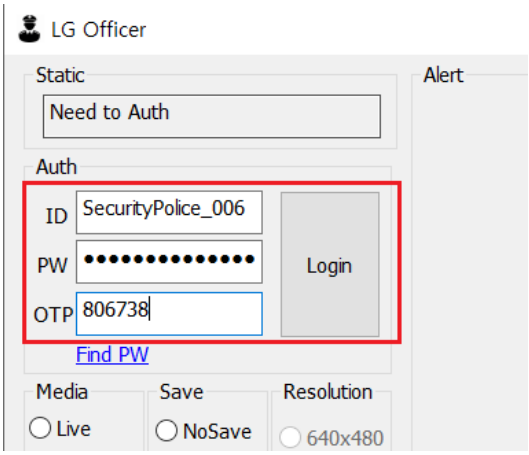
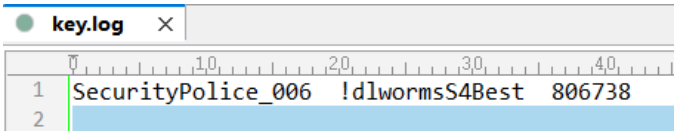
found that retrieved vehicle information is printed to the console when the server has successfully processed the client's vehicle inquiry request.



If the attacker succeeded in checking the output of the server console (for example, malware is installed on the server that periodically takes screenshots and sends it to the attacker's server), the attacker could easily obtain confidential information. So it is a vulnerability.

13.2.10. V10 - User input is not protected

Summary	Password input via the keyboard is not protected.
Location	[Client/IgofficerDlg.cpp #272]
Approach	[Code Review], [Design Review]
CIA	[CONFIDENTIALITY]
Impact	[MEDIUM]
Description	
In general, user input via the keyboard is not protected. Therefore, if a keylogger is installed without the user's knowledge and is logging input, an attacker can find out the ID and password.	
Perceived impact	

The officer's ID, password can be leaked.
Recommended Mitigations
Provide a virtual keypad that changes the layout of the keyboard every time it is executed.
Reproduce Step(PoC)
[Precondition] An attacker succeeds in installing a keylogger to the target device. The keylogger logs all input of the user and sends it to the attacker's server. 1. The officer attempts to log in from the client application.      2. The data that the officer just typed on the keyboard is logged by the keylogger.      3. The log including user's ID, password, otp will be sent to the attacker's server 4. The attacker analyzes the log and extracts the user ID and password

13.2.11. V11 - Execution Files are not digital signed for integrity

Summary	Files(server and client) are not digital signed for integrity. Binary would be

	target for reverse engineering and maybe changed the memory map.

Location	[server.exe], [lgofficer.exe]
Approach	[Reverse Engineering]
CIA	[INTEGRITY]
Impact	[MEDIUM]

Description

Files(server and client) are not digitally signed for integrity. Binary would be target for reverse engineering and maybe changed the memory map

Perceived impact

File could be modified without anyone knowing.

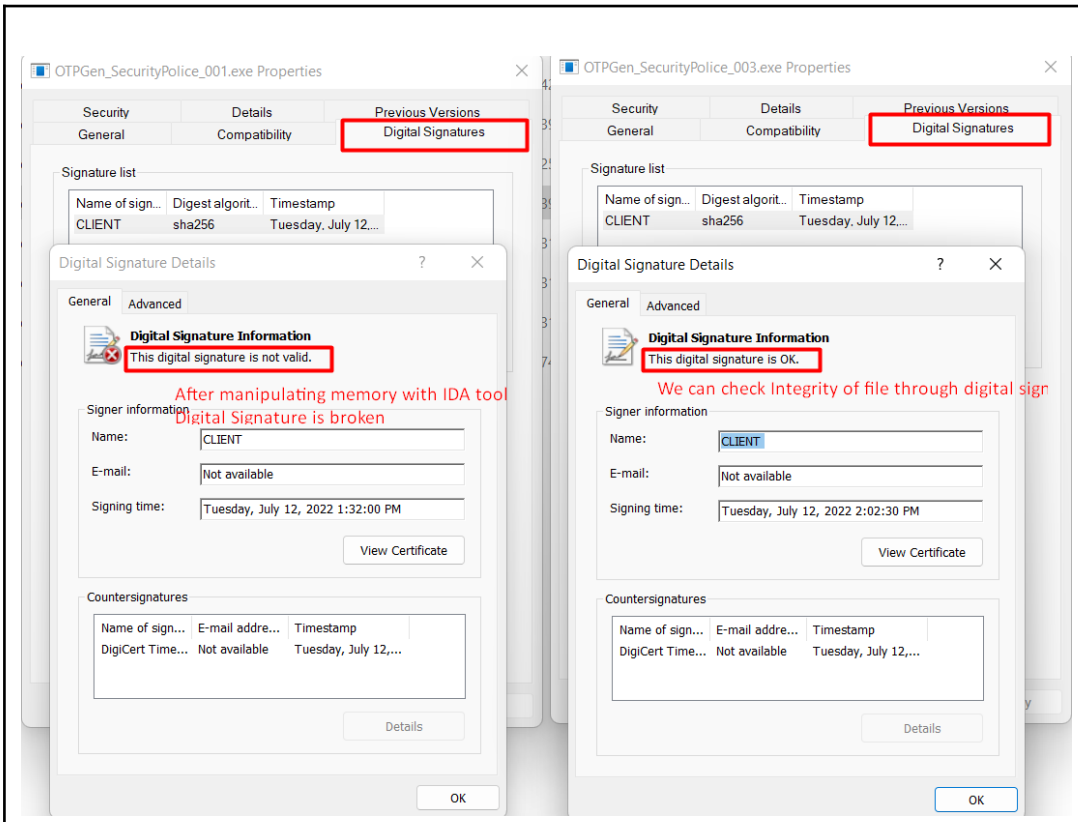
User just use program without validation

Recommended Mitigations

It is necessary to digital sign with your certification for PE binary (exe, dll)

Reproduce Step(PoC)

1. Signed file with signtool with private key
2. See the digital sign tab and sign is validation
3. Check to work normally
4. Manipulate the memory map in binary file
5. You can see that digital sign is not valid
6. We can check whether file is changed or not through result of sign validation



13.2.12. V12 - Tampering configuration file(client-conf.json)

Summary	Tampering configuration file(client-conf.json)
Location	[client-conf.json]
Approach	[Design Review, Code Review]
CIA	[INTEGRITY]
Impact	[MEDIUM]
Description	

The client has a configuration file(client-conf.json) containing the IP address of the target server and the location of the CA certificates to be used for server authentication. The configuration file is exposed in plain text, and if an attacker has access to the file, it is vulnerable to malicious modification.
Perceived impact
Any changes to the configuration file make the server vulnerable to spoofing. If the attacker succeeds in spoofing the server, the officer will try to log in by sending the ID and password to the attacker's server. In this case, the officer's password may be leaked.
Recommended Mitigations
The configuration file should be digitally signed to ensure its integrity.
Reproduce Step(PoC)
[Precondition] An attacker has gained write access to the configuration file. Attacker's server information - ip : 69.219.73.92 - CA cert : attacker_ca.crt - cert : attacker.crt - private key : attacker.key 1. open client-conf.json file 2. change the "server_ip" to the attacker server's ip (69.219.73.92) 3. change the "ca_cert_path" to attacker's ca path (keyes/ca/attacker.crt) 4. push the attacker's ca file in the keys/ca/ path of the target storage. 5. The officer did not notice this attack and runs the client application and logs in as usual. 6. The officer's ID, password will be sent to the attacker's server

13.2.13. V13 - Tampering server DB

Summary	Tampering server DB

Location	[server.cpp #377]
Approach	[Design Review, Code Review]
CIA	[INTEGRITY]
Impact	[MEDIUM]

Description

DB might use Oracle DB. The API is opened.
Can modify or delete using open API.

Perceived impact

DB file has not to be updated by server.exe code. Update DB scenario does not exist.

Recommended Mitigations

Database has to be set to read only because the DB update scenario does not exist. Special authority has to be applied to modify the database.

Reproduce Step(PoC)

[Precondition]

Attackers know the IP and PORT.
1. insert a function to run(need to assembly code)

```
{
    user_dbp->del(user_dbp, NULL, &key, 0);
}
```

When I insert the "ID : SecurityPolice_006"'s PW and OTP correctly, the login fails.
Because I removed the "ID : SecurityPolice_006" account in DB.
The licenseplate.DB could be modified or deleted.

14. Lessons Learned

1. We learned how important it is to know and be able to use a lot of testing tools when looking for vulnerabilities in cybersecurity. We will study good tools and increase our understanding so that we can use them for the next project.

2. We learned the importance of secure storage such as HSM, Intel SGX protection area. The secure storage area should be considered at the system design stage.

3. We learned how important integrity of Binary is. We need the following to solve this problem. → Certificate, digital signature

4. We tried to use the fuzzing test, but the fuzzing tool was not usefully utilized due to the difference in the environment of the platform. We will try to improve our understanding of fuzzing tools and learn how to use more types of fuzzing tools.