Automatic License Plate Recognition (ALPR) System

Security 4 Best

Document Information

Issuing authority	S4Best Team
Status of document	Released

Revision History

Version	Date	Comment	Author
P1-0.5	2022-07-06	Project Phase 1 Release	S4Best Team
P1-1.0	2022-07-10	Update Implementation, Mitigation, Lessons Learned	S4Best Team
P2-0.5	2022-07-13	Add Project Phase 2	S4Best Team
P2-1.0	2022-07-14	Project Phase 2 Release	S4Best Team

Conventions and Acronyms

Conventions

In this section, describe useful notes and important things audience should know as follows.

NOTE
Useful notes.

CAUTION
Important things

Acronyms

Acronym	Description
Abuse Use Case	Deliberate abuse of functional use cases in order to yield unintended results.
Accountability	The property that ensures that the actions of an entity may be traced uniquely to that entity.
Actor (Threat Agent)	Person who originates attacks, either with malice or by accident, taking advantage of vulnerabilities to create loss.
Application Programming Interface (API)	A source code interface that a computer system or program library provides to support requests for services to be made of it by a computer program [PCI HSM Security Req].
Asset	An asset is a resource of value. It varies by perspective. To a business, an asset might be the availability of information, or the information itself, such as customer data. It might be intangible, such as a company's reputation.
Attack (Exploit)	An attack is an action taken that utilizes one or more vulnerabilities to realize a threat.
Attack Surface	Logical area (browser stack, infrastructure components, etc.) or physical area (hotel kiosk) that an attack may occur or originate from.
Attack Vector	Point and channel for which attacks travel over (card reader, form fields, network proxy, client browser, etc.).
Authenticity	The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator [NIST SP 800-137, CNSSI 4009].
Authentication	The process of determining whether someone or something is, in fact, who or what it is declared to be “http://whatis.techtarget.com”
Authorization	The official management decision given by a senior organizational official to authorize operation of an information system and to explicitly accept the risk to organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, and the nation based on the implementation of an agreed-upon set of security controls [NIST SP 800-137, CNSSI 4009].
Availability	Ensuring timely and reliable access to and use of information [NIST SP 800-137, 44 U.S.C., Sec. 3542]. Capability of a product to provide a stated function if demanded, under given conditions over its defined lifetime [ISO 26262-1].
Confidentiality	Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information [NIST SP 800-137, 44 U.S.C., Sec. 3542].

Countermeasures (Control)	Countermeasures address vulnerabilities to reduce the probability of attacks or the impacts of threats. They do not directly address threats; instead they address the factors that define the threats.
Impact	Value of damage possibly sustained via an attack.
Integrity	Guarding against improper information modification or destruction and includes ensuring information non-repudiation and authenticity [NIST SP 800-137, 44 U.S.C., Sec. 3542].
Multi-tenant	An architecture in which a single computing resource is shared but logically isolated to serve multiple consumers [NIST.SP.500-322].
Non-repudiation	The ability to provide proof of the integrity and origin of data.
Privacy	The ability to provide protection against personal data discovery and misuse of that information by other users [Common Criteria Part 2].
Possession and/or control	the system and associated processes shall be designed, implemented, operated and maintained so as to prevent unauthorized control, manipulation or interference
Randomness	A random bit sequence could be interpreted as the result of the flips of an unbiased “fair” coin with sides that are labeled “0” and “1,” with each flip having a probability of exactly ½ of producing a “0” or “1.” Furthermore, the flips are independent of each other: the result of any previous coin flip does not affect future coin flips. The unbiased “fair” coin is thus the perfect random bit stream generator, since the “0” and “1” values will be randomly distributed (and [0,1] uniformly distributed). All elements of the sequence are generated independently of each other, and the value of the next element in the sequence cannot be predicted, regardless of how many elements have already been produced [NIST 800-22].
Safety	The design, implementation, operation and maintenance of the system and associated processes shall not jeopardize the health and safety of individuals, the environment or any associated assets. Absence of unreasonable risk due to hazards caused by malfunctioning behavior of E/E systems [ISO 26262-1].
Spoof	The term is used to describe a variety of ways in which hardware and software can be fooled. IP spoofing, for example, involves trickery that makes a message appear as if it came from an authorized IP address.
Tampering	The ability to change data in transit or in a data store.
Threat	A threat is an undesired event. A potential occurrence often best described as an effect that might damage or compromise an asset or objective. It is

	relative to each site, industry, company and is more difficult to uniformly define.
Trasnport Layer Security (TLS)	Transport Layer Security (TLS) is a cryptographic protocol designed to provide communications security over a computer network. The protocol is widely used in applications such as email, instant messaging, and voice over IP, but its use in securing HTTPS remains the most publicly visible.
Secure Socket Layer(SSL)	Netscape's Secure Socket Layer protocol [SSL3]. TLS is based on SSL Version 3.0. [RFC5246]

Table of Contents

Figures

Tables

1 Automatic License Plate Recognition (ALPR) System

1.1 Introduction of Automatic License Plate Recognition (ALPR) System

The Automatic License Plate Recognition (ALPR) System is an integrated solution of severs that provide license plate database lookups and remote camera applications designed to support law enforcement.
Our system consists of an application and a server operating in a Windows environment and provides easy license plate recognition and database inquiry using a remote camera to provide a quick and accurate vehicle inquiry service to police officers.

1.2 Team S4Best Role and Responsibility

Table 1 Team member Role and Responsibilities

Member	Role	Responsibility
Namseok Kim	Team Leader	Project Management DevOps
		Static Analysis
Seongeun Kim	System And Software Architect	Architecture Design Server Development
		Design Review
Choongrae Kim	Security Architect	Security Architecture Design Secure Communication Development
		Design Review
Jiwoong Eom	Server Engineer/Developer	Authentication & Database Development Risk Analysis
		Exploit Vulnerability
Jeongho Choi	Client Engineer/Developer	Authentication Development Threat Analysis
		Risk Assessment
Jaegeun Lee	Core Engineer/Developer	Core Engineering Client Development
		Exploit Vulnerability

2 Schedule

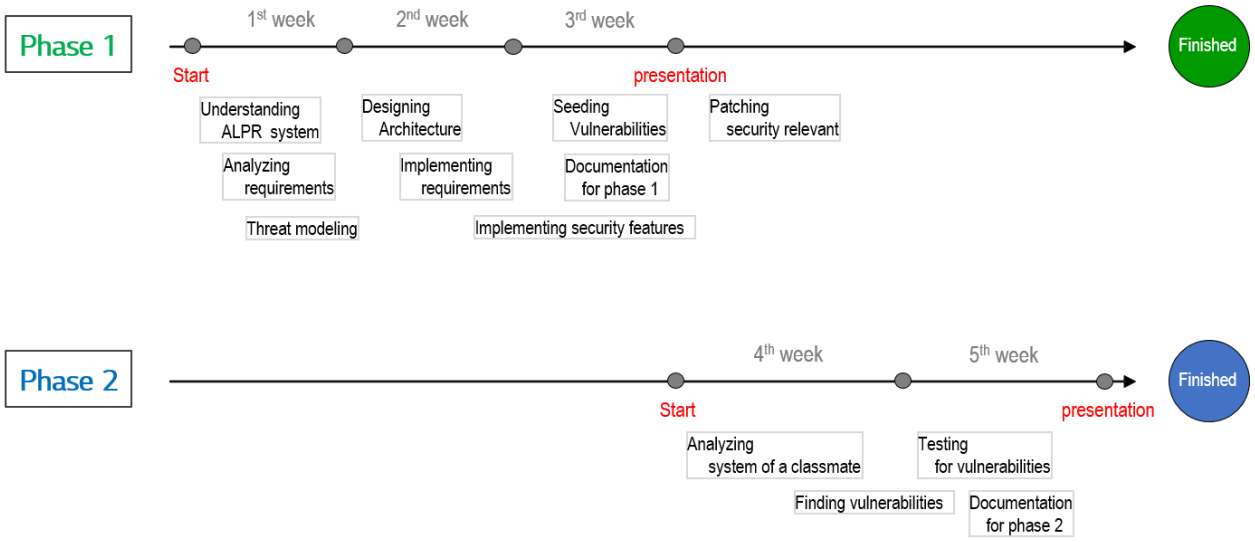


Figure 1 Plans and Schedule

Phase 1 : Secure Development

1 Requirement Engineering

1.1 Client – Functional Requirements

Table 2 Client – Functional Requirements

ID	Statement
REQ_CLI_FUNC_001	The system shall allow an officer to access the ALPR system through a secure web interface
REQ_CLI_FUNC_002	The system shall allow an officer to login and authenticate users locally and to the backend license plate database lookup. The system must use two factor authentication for sign on and user credentials must be protected.
REQ_CLI_FUNC_003	The system should allow a law enforcement officer to select and save retrieved information locally.
REQ_CLI_FUNC_004	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.
REQ_CLI_FUNC_005	The system should read images from the vehicle camera or a playback file and identify license plates for evaluation.
REQ_CLI_FUNC_006	The system should perform the ALPR function in real-time while maintaining a frame rate of at least 25fps.
REQ_CLI_FUNC_007	The system should query the backend license plate server for details about the vehicle. 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.
REQ_CLI_FUNC_008	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.
REQ_CLI_FUNC_009	The system should provide an area in the user interface that always contains the current camera /playback view.
REQ_CLI_FUNC_010	The system should allow officers to configure computed camera / playback frames per second, average time per frame, jitter and frame number.
REQ_CLI_FUNC_011	The system should allow the officer to choose between using a live camera and playback file in the UI.
REQ_CLI_FUNC_012	The system should alert officers of any communication errors or failures.

1.2 Client – Non-Functional Requirements

Table 3 Client - Non-Funtional Requirements

ID	Statement
REQ_CLI_NON_001	Lost or compromised credentials must be handled in a reasonable way.
REQ_CLI_NON_002	The system should provide secure communication between the client application and to the backend license plate database lookup system.
REQ_CLI_NON_003	The ability to detect network connectivity issues with the backend server within 5 seconds and automatically resolve the communication issue if possible.
REQ_CLI_NON_004	The system must fetch vehicle information in no more than 10 seconds as officers are often making queries in real time.

1.3 Sever – Functional Requirements

Table 4 Server - Functional Requirements

ID	Statement
REQ_SVR_FUNC_001	Support license plate queries.
REQ_SVR_FUNC_002	Authenticate remote laptop users.
REQ_SVR_FUNC_003	Support multiple users.
REQ_SVR_FUNC_004	Return the best match license plate if there is not an exact match that includes a configurable minimum confidence threshold to support a partial match.
REQ_SVR_FUNC_005	Track the average number of queries per second for each user and overall queries per second, for all users.
REQ_SVR_FUNC_006	Track the number partial matches and no matches for each user and all users
REQ_SVR_FUNC_007	Support configurable values via a configuration file.

1.4 Client – Non-Functional Requirements

Table 5 Client - Functional Requirements

ID	Statement
REQ_SVR_NON_001	Ensure secure communication with the client applications.

2 Security Goals

2.1 Business Goal

The system allows authorized users to make decisions based on the information provided by the image recognition system.
Earn our customer's trust.

2.2 Security Goals

Table 6 Security Goals

ID	Statement
G-01	Encrypts Sensitive Information
G-02	Provides Authentication
G-03	Provides integrity of sensitive data

3 Preliminary System Architecture

This section provide overall system description and strategy for ALPR system.

3.1 Preliminary System Architecture and Item boundary

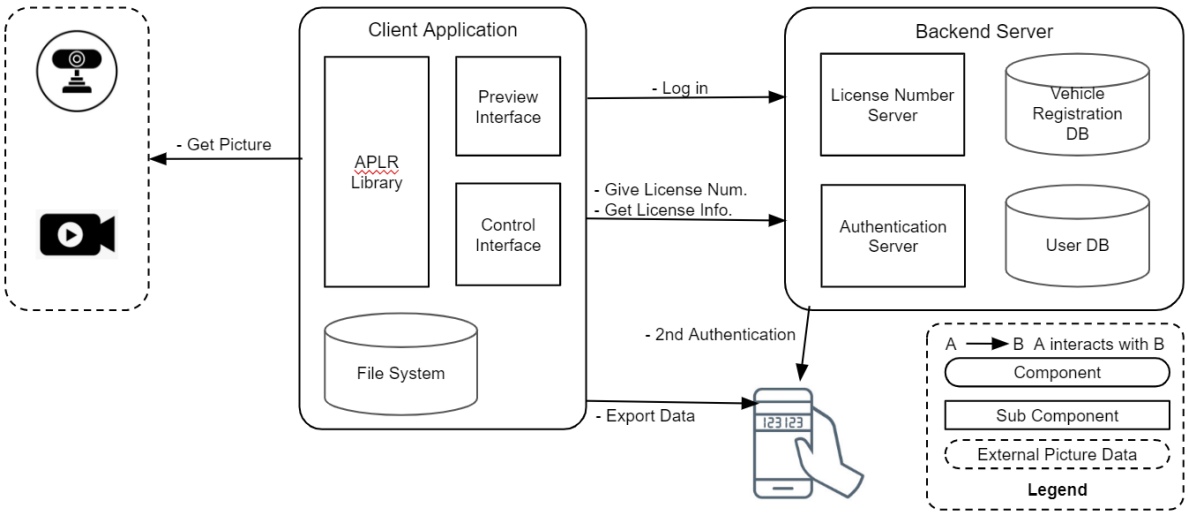


Figure 2 Preliminary System Architecture

3.2 System Architecture Element

Table 7 ALPR System Definition

	Client Application	Backend Server	Mobile Phone	External Picture Data
HW	x86 Base PC	x86 Base PC	NA	NA
Interface	WiFi, Cloud	WiFi, Cloud	WiFi, Cloud	USB, Block IO
OS	Windows 10	Windows 10	Android	NA
SW Module	APLR Lib., Frame Image Interface, Control Interface, File System	License Number Server, Vehicle Registration DB, Authentication Server, User DB	2nd Authentication Application, Vehicle Information Viewer	Live Camera, Playback
Data	User ID, PW Officer authentication result Image of vehicles on the road Image Frame Information	User ID, PW Two Factor Authentication Information Officer authentication result Vehicle license number	Two Factor Authentication Information Vehicle information corresponding to license number	Image of vehicles on the road

	(FPS, Average Time per Frame, Frame Number, Jitter) Vehicle license number Vehicle information corresponding to license number	Vehicle information corresponding to license number		

Table 8 ALPR System Definition

Component	Sub Component	Description
External Picture Data	Live Camera	Video frame image transmitted in real time through the camera
	Play-Back	Video frame image obtained from saved video file
Client Application	APLR Library	Recognizes the license plate area from a specific frame image and extracts the license number
	Preview Interface	Outputs frame images and corresponding frame information (FPS, Average Time per Frame, Frame Number, Jitter)
	Control Interface	Proceeds with officer's certification process for client application Sets operation mode of application(frame input selection) Outputs license information of cars
	File System	Saves license information obtained from backend server
Backend Server	License Number Server	Search and retrieve the vehicle information corresponding to license number delivered from client application from DB, and send those to client application again.
	Authentication Server	Proceed with user authentication using the two factor authentication with the user ID/PW delivered from client application.
	Vehicle Registration DB	Stores various vehicle information for each vehicle license number. The field of the saved record follows the predefined form in the assignment introduction.
	User DB	Stores officer's user credential, such as ID, PW, account recovery hint, etc. which are corresponding to each officer.
Mobile Phone	2nd Authentication App.	Mobile application for two factor authentication
	Vehicle Information Viewer	Mobile application for saving and checking vehicle information delivered from client application.

3.3 Operation Scenario

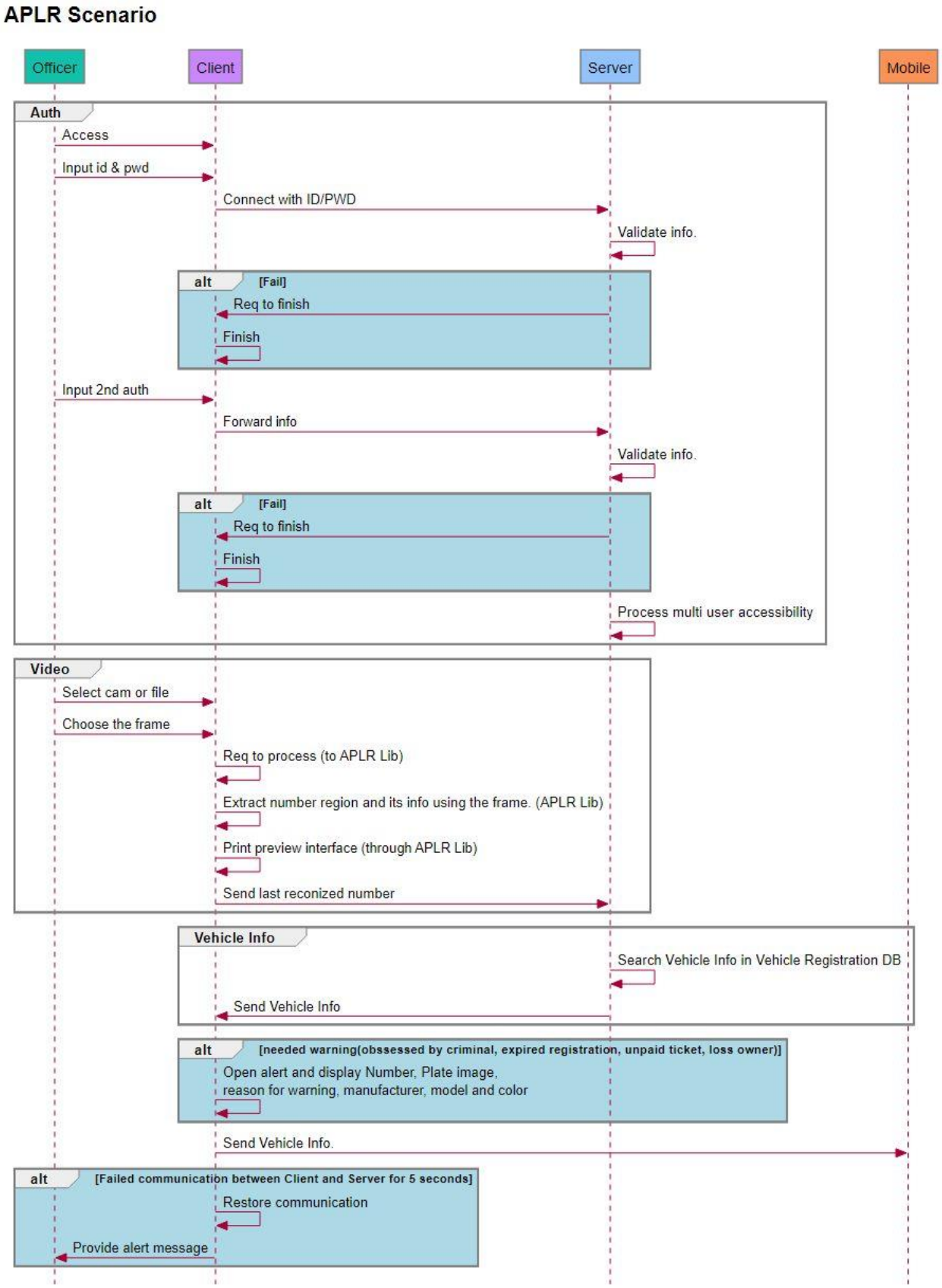


Figure 3 ALPR Operation Scenario

1. Client Application Execution and Connection to Server

1.1 The officer runs the client application and connects to the backend server by entering the user ID and password.

1.2 Backend server performs additional verification using two factor authentication infrastructure after validating user ID and password delivered from client application.

1.3 When the validation of step 1.1 and 1.2 is completed, the validation result is delivered to the client application.

1.4 If the result of user verification in step 1.3 is valid, proceed to step 2.1. If it is not valid, the application is terminated.

2. Selection of image of vehicles on the road in client application

2.1 The officer selects one of live camera and playback file as input of image of vehicles on the road.

2.2 Select the frame in the mode selected in step 2.1 as the input of the ALPR Library and proceed with the image recognition process.

3. License number recognition using APLR library

3.1 Extracts the area of the license plate and the license number inside area by using the frame selected in step 2.1 as input.

3.2 Extracts the frame information being processed and outputs it at the bottom of the preview interface.

3.3 Send the last recognized number to backend server

4. Search vehicle information in backend server

4.1 Backend server retrieves the vehicle information corresponding to the number delivered in step 3.3 from vehicle registration DB and delivers it to the client application.

4.2 Step 3.3 and 4.1 should be done within 10 seconds.

5. Alert output of client application

5.1 Alert is displayed when a warning is found in the vehicle information delivered in step 4.1.

Alert displays the vehicle number and license plate image, as well as the reason for the warning, make, model, and vehicle color.

5.2 The reasons for the warning in step 5.1 include theft and criminal possession, deregistration, unpaid tickets, and missing owner.

6. Storing vehicle information and forwarding to mobile phone in client application

6.1 The vehicle information delivered in step 4.1 is collected by the client application and delivered to the mobile phone if necessary.

7. Exception handling when communication fails

7.1 Client application should recover communication problem if communication with backend server fails for 5 seconds.

7.2 The proper alert should be provided to user if a communication problem occurs.

8. Backend server facilities

8.1 Multiple officers physically separated can be connected to backend server.

8.2 The average and total number of queries per second are stored for individual officers and all users, respectively.

8.3 Use the configuration file to set the server operation.

3.4 Assumptions

Table 9 Assumption list

Assumptions No.	Description
001	User Credentials such as ID/PW/OTP Seed Key are included in the server DB.
002	ACL is applied to User Credential DB and DB Keeper (statement) is applied.
003	Cryptographic Key (Sym/Asym) is stored in a reliable and secure device. ex) HSM.
004	OTP Seed Key for each user is created when creating an account and is ported to a unique OTP device for each user.
005	OTP device is owned exclusively by each user like a banking system.

4 Threat Modeling

4.1 STRIDE.

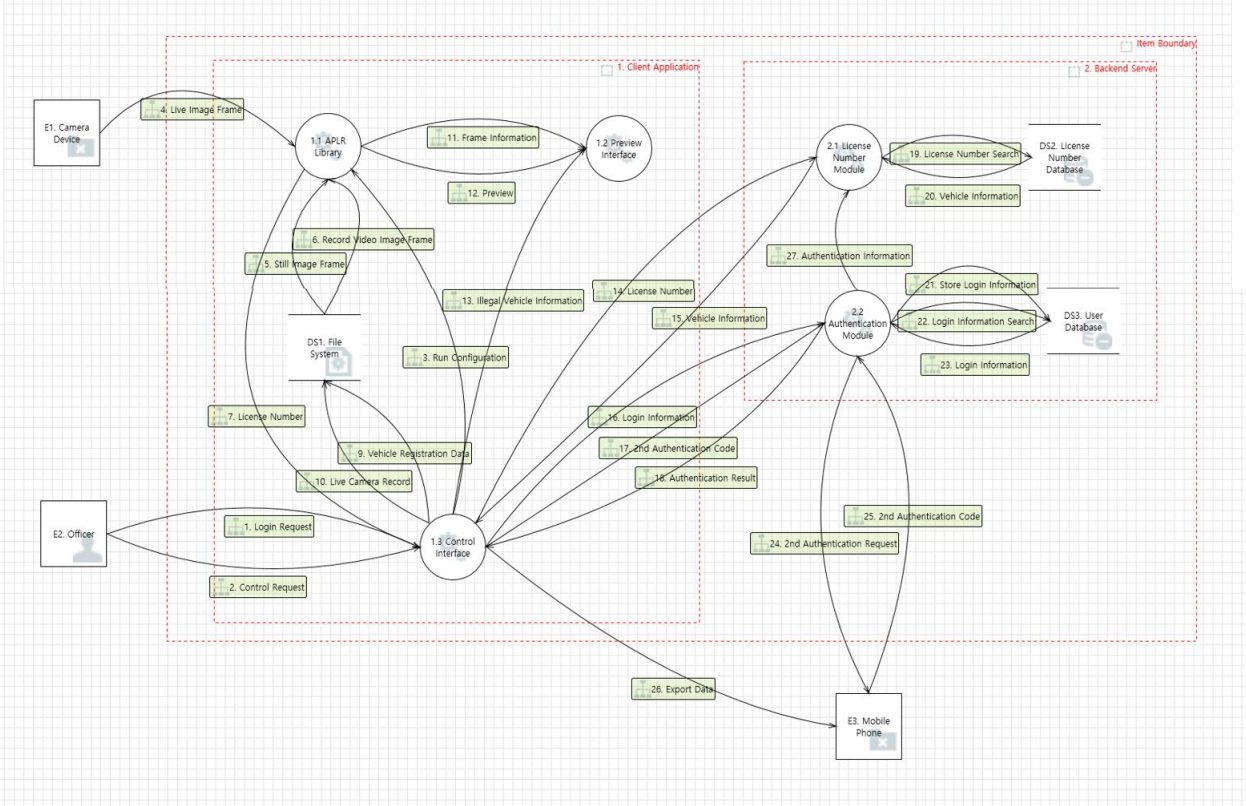


Figure 4 Data Flow Diagram

4.2 OWASP Risk Assessment

4.2.1 Risk Rating

Table 10 Risk Rating

Overall Risk Severity = Likelihood x Impact				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Likelihood and Impact Levels	
0 to <3	LOW
3 to <6	MEDIUM
6 to 9	HIGH

4.2.2 Server Threat List

Table 11 Server Threat 001

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	
19. License Number Search	Threat#1- Spoofing of Destination Data Store DS2. License Number Database[Spoofing] DS2. License Number Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of DS2. License Number Database. Consider using a standard authentication mechanism to identify the destination data store.	Threat Agent	Skill level	3 - Network and programming skills	5.125	MEDIUM	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	6.5	HIGH	High
			Motive	6 -				Loss of integrity	9 - All data totally corrupt			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	6 - Authenticated users				Loss of accountability	9 - Completely anonymous			
		Vulnerability	Ease of discovery	3 - Difficult			Business Impact	Financial damage	3 - Minor effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	9 - Brand damage			
			Awareness	4 - Hidden				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	5 - Hundreds of people			

Table 12 Server Threat 002

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	
20. Vehicle Information	Threat#2- Spoofing of Source Data Store DS2. License Number Database [Spoofing] DS2. License Number Database may be spoofed by an attacker and this may lead to incorrect data delivered to 2.1 License Number Module. Consider using a standard authentication mechanism to identify the source data store..	Threat Agent	Skill level	3 - Network and programming skills	5.125	MEDIUM	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	6.5	HIGH	High
			Motive	6 -				Loss of integrity	9 - All data totally corrupt			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	6 - Authenticated users				Loss of accountability	9 - Completely anonymous			
		Vulnerability	Ease of discovery	3 - Difficult			Business Impact	Financial damage	3 - Minor effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	9 - Brand damage			
			Awareness	4 - Hidden				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	5 - Hundreds of people			

Table 13 Server Threat 003

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	
27. Authentication Information	Threat#3- Elevation Using Impersonation [Elevation Of Privilege] 2.1 License Number Module may be able to impersonate the context of 2.2 Authentication Module in order to gain additional privilege.	Threat Agent	Skill level	3 - Network and programming skills	4.25	MEDIUM	Technical Impact	Loss of confidentiality	9 - All data disclosed	5.375	MEDIUM	Medium
			Motive	4 - Possible reward				Loss of integrity	4 -			
			Opportunity	7 - Some access or resources required				Loss of availability	1 - Minimal secondary services interrupted			
			Group Size	6 - Authenticated users				Loss of accountability	9 - Completely anonymous			
		Vulnerability	Ease of discovery	2 -			Business Impact	Financial damage	3 - Minor effect on annual profit			
			Ease of exploit	2 -				Reputation damage	7 -			
			Awareness	1 - Unknown				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	5 - Hundreds of people			

Table 14 Server Threat 004

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	
21. Store Login Information	Threat#4- Spoofing of Destination Data Store DS3. User Database [Spoofing] DS3. User Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of DS3. User Database. Consider using a standard authentication mechanism to identify the destination data store.	Threat Agent	Skill level	3 - Network and programming skills	4.25	MEDIUM	Technical Impact	Loss of confidentiality	9 - All data disclosed	5.5	MEDIUM	Medium
			Motive	4 - Possible reward				Loss of integrity	3 - Minimal seriously corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	1 - Minimal secondary services interrupted			
			Group Size	6 - Authenticated users				Loss of accountability	9 - Completely anonymous			
		Vulnerability	Ease of discovery	2 -			Business Impact	Financial damage	3 - Minor effect on annual profit			
			Ease of exploit	2 -				Reputation damage	9 - Brand damage			
			Awareness	1 - Unknown				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	5 - Hundreds of people			

Table 15 Server Threat 005

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	
21. Store Login Informaiton	Threat#5 - Denial of Service Data Store DS3. User Database [Denial of Service] Does 2.2 Authentication Module or DS3. User Database 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.	Threat Agent	Skill level	4 - Advanced computer user	6.25	HIGH	Technical Impact	Loss of confidentiality	1 -	4	MEDIUM	High
			Motive	1 - Low or no reward				Loss of integrity	3 - Minimal seriously corrupt data			
			Opportunity	9 - No access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	9 - Anonymous Internet users				Loss of accountability	9 - Completely anonymous			
		Vulnerability	Ease of discovery	6 -			Business Impact	Financial damage	1 - Less than the cost to fix the vulnerability			
			Ease of exploit	6 -				Reputation damage	4 - Loss of major accounts			
			Awareness	6 - Obvious				Non-compliance	2 - Minor violation			
			Intrusion detection	9 - Not logged				Privacy violation	5 - Hundreds of people			

Table 16 Server Threat 006

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	
22. Login Information Search	Threat#6- Spoofing of Destination Data Store DS3. User Database [Spoofing] DS3. User Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of DS3. User Database. Consider using a standard authentication mechanism to identify the destination data store.	Threat Agent	Skill level	3 - Network and programming skills	4.625	MEDIUM	Technical Impact	Loss of confidentiality	1 -	3.25	MEDIUM	Medium
			Motive	4 - Possible reward				Loss of integrity	1 - Minimal slightly corrupt data			
			Opportunity	8 -				Loss of availability	1 - Minimal secondary services interrupted			
			Group Size	6 - Authenticated users				Loss of accountability	9 - Completely anonymous			
		Vulnerability	Ease of discovery	3 - Difficult			Business Impact	Financial damage	3 - Minor effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	1 - Unknown				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	2 -			

Table 17 Server Threat 007

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	
22. Login Information Search	Threat#7- Potential Excessive Resource Consumption for 2.2 Authentication Module or DS3. User Database[Denial Of Service] Does 2.2 Authentication Module or DS3. User Database 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.	Threat Agent	Skill level	4 - Advanced computer user	5.25	MEDIUM	Technical Impact	Loss of confidentiality	1 -	3.75	MEDIUM	Medium
			Motive	1 - Low or no 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	8 -				Loss of accountability	9 - Completely anonymous			
		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	1 - Unknown				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	2 -			

Table 18 Server Threat 008

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	
23. Login Information	Threat#8- Spoofing of Source Data Store DS3. User Database [Spoofing] DS3. User Database may be spoofed by an attacker and this may lead to incorrect data delivered to 2.2 Authentication Module. Consider using a standard authentication mechanism to identify the source data store.	Threat Agent	Skill level	3 - Network and programming skills	4.125	MEDIUM	Technical Impact	Loss of confidentiality	9 - All data disclosed	5.625	MEDIUM	Medium
			Motive	4 - Possible reward				Loss of integrity	5 - Extensive slightly corrupt data			
			Opportunity	4 - Special access or resources required				Loss of availability	1 - Minimal secondary services interrupted			
			Group Size	6 - Authenticated users				Loss of accountability	9 - Completely anonymous			
		Vulnerability	Ease of discovery	3 - Difficult			Business Impact	Financial damage	3 - Minor effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	8 -			
			Awareness	1 - Unknown				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	5 - Hundreds of people			

Table 19 Server Threat 009

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	
23. Login Information	Threat#9- Weak Access Control for a Resource[Information Disclosure] Improper data protection of DS3. User Database can allow an attacker to read information not intended for disclosure. Review authorization seattings.	Threat Agent	Skill level	3 - Network and programming skills	4.125	MEDIUM	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	4.75	MEDIUM	Medium
			Motive	4 - Possible reward				Loss of integrity	1 - Minimal slightly corrupt data			
			Opportunity	4 - Special access or resources required				Loss of availability	1 - Minimal secondary services interrupted			
			Group Size	6 - Authenticated users				Loss of accountability	9 - Completely anonymous			
		Vulnerability	Ease of discovery	3 - Difficult			Business Impact	Financial damage	3 - Minor effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	9 - Brand damage			
			Awareness	1 - Unknown				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	5 - Hundreds of people			

4.2.1 Client Threat List

Table 20 Client Threat 001

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	
6. Record Video Image Frame (DS1. File System → APLR Lib)	Threat#2 - Weak Access Control for a Resource [Information Disclosure] Improper data protection of DS1. File System can allow an attacker to read information not intended for disclosure. Review authorization settings.	Threat Agent	Skill level	4 - Advanced computer user	6.375	HIGH	Technical Impact	Loss of confidentiality	2 - Minimal non-sensitive data disclosed	4.125	MEDIUM	High
			Motive	4 - Possible reward				Loss of integrity	3 - Minimal seriously 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	1 - Less than the cost to fix the vulnerability			
			Ease of exploit	5 - Easy				Reputation damage	9 - Brand damage			
			Awareness	9 - Public knowledge				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	1 -			

Table 21 Client Threat 002

6. Record Video Image Frame (DS1. File System → APLR Lib)	Threat#3 - Spoofing of Source Data Store DS1. File System [Spoofing] DS1. File System may be spoofed by an attacker and this may lead to incorrect data delivered to 1.1 APLR Library. Consider using a standard authentication mechanism to identify the source data store.	Threat Agent	Skill level	4 - Advanced computer user	6.375	HIGH	Technical Impact	Loss of confidentiality	2 - Minimal non-sensitive data disclosed	3.875	MEDIUM	High
			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,			
			Group Size	6 - Authenticated users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	1 - Less than the cost to fix the vulnerability			
			Ease of exploit	5 - Easy				Reputation damage	9 - Brand damage			
			Awareness	9 - Public knowledge				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	1 -			

Table 22 Client Threat 003

9. Vehicle Registration Data (Control Interface → DS1. File System)	Threat#6 - Spoofing of Source Data Store DS1. File System [Spoofing] DS1. File System may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of DS1. File System. Consider using a standard authentication mechanism to identify the destination data store.	Threat Agent	Skill level	4 - Advanced computer user	6.375	HIGH	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	4.75	MEDIUM	High
			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			
			Group Size	6 - Authenticated users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	1 - Less than the cost to fix the vulnerability			
			Ease of exploit	5 - Easy				Reputation damage	9 - Brand damage			
			Awareness	9 - Public knowledge				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	5 - Hundreds of people			

Table 23 Client Threat 004

9. Vehicle Registration Data (Control Interface → DS1. File System)	Threat#7 - Potential Excessive Resource Consumption for 1.3 Control Interface or DS1. File System [Denial Of Service] Does 1.3 Control Interface or DS1. File System 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.	Threat Agent	Skill level	4 - Advanced computer user	6.25	HIGH	Technical Impact	Loss of confidentiality	2 - Minimal non-sensitive data disclosed	5.125	MEDIUM	High
			Motive	4 - Possible reward				Loss of integrity	3 - Minimal seriously corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	9 - All services completely lost			
			Group Size	6 - Authenticated 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	9 - Brand damage			
			Awareness	9 - Public knowledge				Non-compliance	7 - High profile violation			
			Intrusion detection	8 - Logged without review				Privacy violation	1 -			

Table 24 Client Threat 005

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	
1. Login Request/ 2. Control Request	Threat# - Spoofing the E2. Officer External Entity [Spoofing] E2 Officer may be spoofed by an attacker and this may lead to unauthorized access to 1.3 Control Interface. Consider using a standard authentication mechanism to identify the external entity.	Threat Agent	Skill level	9 - No technical skills	5.5	MEDIUM	Technical Impact	Loss of confidentiality	9 - All data disclosed	6.5	HIGH	High
			Motive	4 - Possible reward				Loss of integrity	3 - Minimal seriously corrupt data			
			Opportunity	4 - Special access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	6 - Authenticated 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	9 - Brand damage			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	5 - Hundreds of people			

Table 25 Client Threat 006

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	
1. Login Request/ 2. Control Request	Threat# - Data Flow Sniffing [Information Disclosure] Data flowing across 1. Login Request 2. Control Request 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.	Threat Agent	Skill level	9 - No technical skills	6.125	HIGH	Technical Impact	Loss of confidentiality	9 - All data disclosed	6.5	HIGH	Critical
			Motive	9 - High reward				Loss of integrity	3 - Minimal seriously corrupt data			
			Opportunity	4 - Special access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	6 - Authenticated 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	9 - Brand damage			
			Awareness	6 - Obvious				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	5 - Hundreds of people			

Table 26 Client Threat 007

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	
1. Login Request/ 2. Control Request	Threat# - Potential Process Crash or Stop for 1.3 Control Interface [Denial Of Service] 1.3 Control Interface crashes, halts, stops or runs slowly; in all cases violating an availability metric.	Threat Agent	Skill level	4 - Advanced computer user	6	HIGH	Technical Impact	Loss of confidentiality	2 - Minimal non-sensitive data disclosed	3.25	MEDIUM	High
			Motive	1 - Low or no reward				Loss of integrity	1 - Minimal slightly corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	6 - Authenticated 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	9 - Public knowledge				Non-compliance	2 - Minor violation			
			Intrusion detection	9 - Not logged				Privacy violation	0 -			

Table 27 Client Threat 008

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	
1. Login Request/ 2. Control Request	Threat# - Elevation by Changing the Execution Flow in 1.3 Control Interface [Elevation Of Privilege] An attacker may pass data into 1.3 Control Interface in order to change the flow of program execution within 1.3 Control Interface to the attacker's choosing.	Threat Agent	Skill level	1 - Security penetration skills	5.25	MEDIUM	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	5.75	MEDIUM	Medium
			Motive	9 - High 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	6 - Authenticated 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	9 - Brand damage			
			Awareness	4 - Hidden				Non-compliance	5 - Clear violation			
			Intrusion detection	9 - Not logged				Privacy violation	3 - One individual			

4.2.2 Network Threat List

Table 28 Network Threat 001

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	
License Number	Threat#1- Spoofing the 1.3 Control Interface Process [Spoofing] License Number Module may be spoofed by an attacker and this may lead to information disclosure by 1.3 Control Interface. Consider using a standard authentication mechanism to identify the destination process.	Threat Agent	Skill level	3 - Network and programming skills	5	MEDIUM	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	5.125	MEDIUM	Medium
			Motive	4 - Possible reward				Loss of integrity	7 - Extensive seriously corrupt data			
			Opportunity	4 - Special access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	6 - Authenticated 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	8 -				Non-compliance	5 - Clear violation			
			Intrusion detection	3 - Logged and reviewed				Privacy violation	3 - One individual			

Table 29 Network Threat 002

License Number	Threat#3-Potential Data Repudiation by 2.1 License Number Module [Repudiation] License Number Module 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.	Threat Agent	Skill level	9 - No technical skills	7.25	HIGH	Technical Impact	Loss of confidentiality	4 - Minimal critical data disclosed, extensive non-sensitive data disclosed	3.625	MEDIUM	High
			Motive	4 - Possible reward				Loss of integrity	3 - Minimal seriously 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	4 -			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	1 - Less than the cost to fix the vulnerability			
			Ease of exploit	5 - Easy				Reputation damage	4 - Loss of major accounts			
			Awareness	9 - Public knowledge				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	3 - One individual			

Table 30 Network Threat 003

License Number	Threat#4-Data Flow Sniffing [Sniffing] License Number 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.	Threat Agent	Skill level	4 - Advanced computer user	6.25	HIGH	Technical Impact	Loss of confidentiality	9 - All data disclosed	5.875	MEDIUM	High
			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	6 - Authenticated 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	9 - Public knowledge				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	5 - Hundreds of people			

Table 31 Network Threat 004

License Number	Threat#5-Potential Process Crash or Stop [DoS]	Threat Agent	Skill level	3 - Network and programming skills	6.125	HIGH	Technical Impact	Loss of confidentiality	4 - Minimal critical data disclosed, extensive non-sensitive data disclosed	4.75	MEDIUM	High
	License Number Module crashes, halts, stops or runs slowly; in all cases violating an availability metric.		Motive	4 - Possible reward				Loss of integrity	3 - Minimal seriously corrupt data			
			Opportunity	4 - Special 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	4 - Loss of major accounts			
			Awareness	9 - Public knowledge				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	1 -			

Table 32 Network Threat 005

16. Login Information	Threat#6-Spoofing the 1.3 Control Interface Process [Spoofing]	Threat Agent	Skill level	4 - Advanced computer user	5.375	MEDIUM	Technical Impact	Loss of confidentiality	9 - All data disclosed	6.125	HIGH	High
	1.3 Control Interface may be spoofed by an attacker and this may lead to unauthorized access to 2.2 Authentication Module. Consider using a standard authentication mechanism to identify the source process.		Motive	4 - Possible reward				Loss of integrity	7 - Extensive seriously corrupt data			
			Opportunity	5 -				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	6 - Authenticated				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	7 - Easy			Business Impact	Financial damage	5 -			
			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	5 - Hundreds of people			

Table 33 Network Threat 006

16. Login Information	Threat#7-Potential Lack of Input Validation for 2.2 Authentication Modul [Tampering]	Threat Agent	Skill level	3 - Network and programming skills	4.125	MEDIUM	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	4.875	MEDIUM	Medium
	Data flowing across 16. Login Information may be tampered with by an attacker. This may lead to a denial of service attack against 2.2 Authentication Module or an elevation of privilege attack against 2.2 Authentication Module or an information disclosure by 2.2 Authentication Module. 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.		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	6 - Authenticated users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	3 - Difficult			Business Impact	Financial damage	3 - Minor effect on annual profit			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	4 - Hidden				Non-compliance	5 - Clear violation			
			Intrusion detection	3 - Logged and reviewed				Privacy violation	1 -			

Table 34 Network Threat 007

16. Login Information	Threat#8-Data Flow Sniffing[Information Disclosure]	Threat Agent	Skill level	9 - No technical skills	7.25	HIGH	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	4.875	MEDIUM	High
	Data flowing across 16. Login 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..		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	5 - Easy				Reputation damage	4 - Loss of major accounts			
			Awareness	9 - Public knowledge				Non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	1 -			

Table 35 Network Threat 008

16. Login Information	Threat#9-Potential Process Crash or Stop for 2.2 Authentication Module[Denial Of Service]	Threat Agent	Skill level	3 - Network and programming skills	5.75	MEDIUM	Technical Impact	Loss of confidentiality	0 -	3.875	MEDIUM	Medium
	2.2 Authentication Module crashes, halts, stops or runs slowly; in all cases violating an availability metric.		Motive	1 - Low or no reward				Loss of integrity	1 - Minimal slightly corrupt data			
			Opportunity	9 - No access or resources required				Loss of availability	9 - All services completely lost			
			Group Size	9 - Anonymous Internet users				Loss of accountability	9 - Completely anonymous			
		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	9 - Public knowledge				Non-compliance	5 - Clear violation			
			Intrusion detection	3 - Logged and reviewed				Privacy violation	0 -			

Table 36 Network Threat 009

16. Login Information	Threat#10-Elevation by Changing the Execution Flow in 2.2 Authentication Module[Elevation Of Privilege] An attacker may pass data into 2.2 Authentication Module in order to change the flow of program execution within 2.2 Authentication Module to the attacker's choosing.	Threat Agent	Skill level	3 - Network and programming skills	4.125	MEDIUM	Technical Impact	Loss of confidentiality	9 - All data disclosed	4.75	MEDIUM	Medium
			Motive	4 - Possible reward				Loss of integrity	5 - Extensive slightly corrupt data			
			Opportunity	7 - Some access or resources required				Loss of availability	7 - Extensive primary services interrupted			
			Group Size	6 - Authenticated users				Loss of accountability	7 - Possibly traceable			
		Vulnerability	Ease of discovery	3 - Difficult			Business Impact	Financial damage	1 - Less than the cost to fix the vulnerability			
			Ease of exploit	3 - Difficult				Reputation damage	4 - Loss of major accounts			
			Awareness	4 - Hidden				non-compliance	5 - Clear violation			
			Intrusion detection	3 - Logged and reviewed				Privacy violation	0 -			

Table 37 Network Threat 010

16. Login Information	Threat#11-Potential Data Repudiation by 2.2 Authentication Module[Repudiation] 2.2 Authentication Module 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.	Threat Agent	Skill level	9 - No technical skills	7.25	HIGH	Technical Impact	Loss of confidentiality	5 - Extensive critical data disclosed	4.875	MEDIUM	High
			Motive	4 - Possible reward				Loss of integrity	7 - Extensive seriously			
			Opportunity	7 - Some access or				Loss of availability	7 - Extensive primary			
			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			
			Ease of exploit	5 - Easy				Reputation damage	4 - Loss of major accounts			
			Awareness	9 - Public knowledge				non-compliance	5 - Clear violation			
			Intrusion detection	8 - Logged without review				Privacy violation	1 -			

4.3 PnG Risk Assessment

4.3.1 Identify the PnG types, goals, motivations, skills

Table 38 PnG 1

PnG 1	Type	server developer
	**Goal**	Cause ALPR system malfunction, to ruin the Tartan's reputation
	Motivation	He is anger and want to revenge, because there are too much work, lower reward and he is finally fired
	Skill	Extensive knowledge of ALPR system, especially server side, computers, relevant programs, DB, PKI system and access to server system with administrator authority.
	Misuse case	1. Using a backdoor of server system, modify sensitive database file causing system malfunction 2. DDoS/DRDoS attack to known server's IP

Table 39 PnG 2

PnG 2	Type	License revoked driver
	**Goal**	Manipulating driver's license record
	Motivation	This person's license was revoked for multiple traffic violations. He wants to get his license back and drive normally.
	Skill	Trying to login with brute force
	Misuse case	1. Buying hackers or police officers with money to tamper with license records 2. Attempts to login with a brute force and succeeds in tampering with the license record

Table 40 PnG 3

PnG 3	Type	Police Officer
	**Goal**	Modify the plate information
	Motivation	Receive money and change plate information
	Skill	1. takeover a backdoor account accessing to Backend Server 2. repudiate that he has never accessed the backend server
	Misuse case	1.Using a second ID 2. remove audit

Table 41 PnG 4

PnG 4	Type	Hacker
	**Goal**	Show the anti-government slogans on the client screen.
	Motivation	He is an anarchist and hates the power of the nation.
	Skill	Code Injection, Buffer Overflow Attack, DDOS
	Misuse case	Hacker steals the system admin privilege and injects code to display slogans, then executes that code. He can also do DDOS attack.

Table 42 PnG 5

PnG 5	Type	Criminal
	**Goal**	Retrieve and tamper with traffic violation information. and it is used for crime.
	Motivation	to get financial gain
	Skill	subsumption ability and network of criminal engineers in various fields.
	Misuse case	He stole personal privacy information and used them for fraudulent crime.

Table 43 PnG 6

PnG 6	Type	System Manager
	**Goal**	Sneaking all information
	Motivation	Making a fortune
	Skill	Data replication using covert channel
	Misuse case	

4.3.2 Comparison with STRIDE

Indicate whether they discovered threats that did not appear with STRIDE or whether it reinforced the STRIDE results

Table 44 Comparision With STRIDE

Threat		Comparison PnG
Spoofing	TID-S1, TID-S2	Similar with PnG case 4 & 5
Tampering	TID-N6 Login Information	Similar with PnG case 1 & 2 & 3 & 4
Repudiation	TID-N2, TID-N10, Repudiation User Authentication, License Number	Similar with PnG case 3
Information disclosure	TID-N3, TID-N7 User ID/PW, License Number	Similar with PnG case 4 & 5
Denial of Service	TID-S5, TID-N8, TID-N4 16. Login Information 21. Store Login Informaiton License Number	Similar with PnG case 1 & 4
Elevation of Privilege	TID-S3 TID-C8 TID-N9	Similar with PnG case 1 & 4

5 Architecture Design

5.1 System Architecture

5.1.1 System Context Diagram

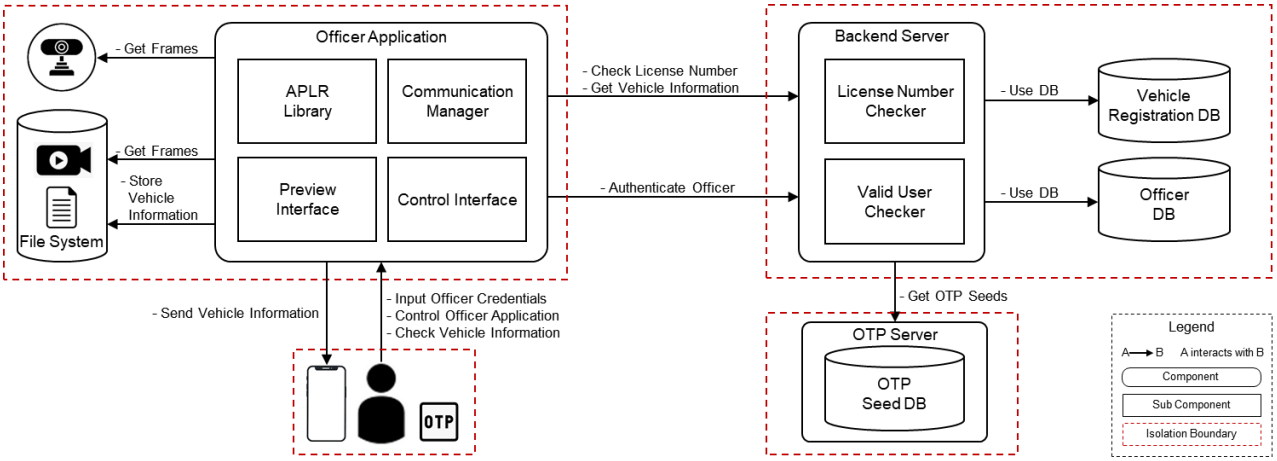


Figure 5 System Context Diagram

Table 45 Entity Description

1st Level	2nd Level	Description
Officer Application	APLR Library	Recognizes the license plate area from a specific frame image and extracts the license number
	Preview Interface	Outputs frame images and corresponding frame information
	Control Interface	Proceeds with officer's certification process for client application Sets operation mode of application(frame input selection) Outputs license information of cars
	Comm. Manager	Connect to backend server
User	Officer	Controls officer application
	OTP Device	Used for 2-factor authentication
	Mobile Phone	Officer checks vehicle information sent from officer app.
Backend Server	License Number Checker	Search and retrieve the vehicle information corresponding to license number delivered from client application from DB, and send those to client application again.
	Valid User Checker	Proceed with user authentication using the two factor authentication with the user ID/PW delivered from client application.
	Vehicle Registration DB	Stores various vehicle information for each vehicle license number.
	Officer DB	Stores officer's user credential, such as ID, PW, account recovery hint, etc
	OTP Seeds DB	Stores seed information of each physical OTP device.

5.1.2 State Transition

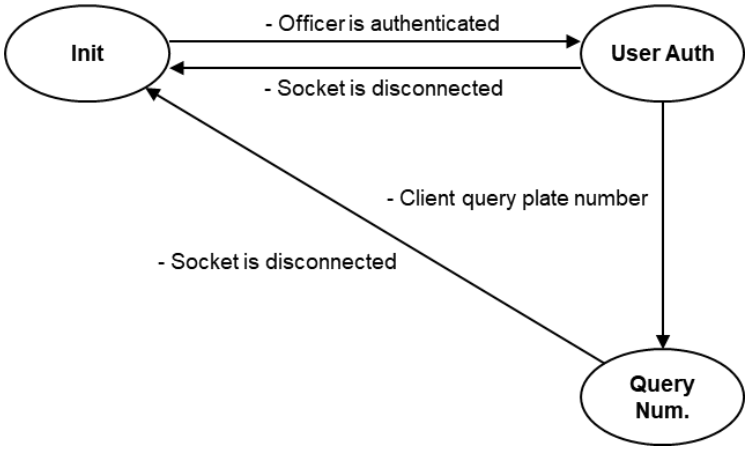


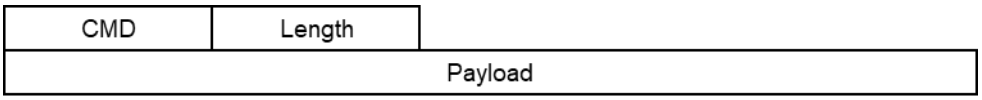
Figure 6 State Transition

Table 46 State Transition

State	Operation Description
Init	- Server rejects all packets except for user authentication commands. - Once finishing the user authentication, state is switched to "User Auth" state. - If the number of currently connected client exceeds maximum number configured, server rejects all attempts to log in.
User Authentication	- Server accepts only plate number query command. - If user authentication commands arrives, server considers multiple log-in attempts with the same ID and closes the connection.
Query Number	- Server responds every query of client. - If connection is closed, state is switched to "Init" state

5.1.3 Communication Packet

- Two packets are delivered sequentially : Header + Payload



- CMD is used to check whether client packet corresponds with server state
- Payload consists of contents corresponding to CMD.

5.1.4 System Service Sequence

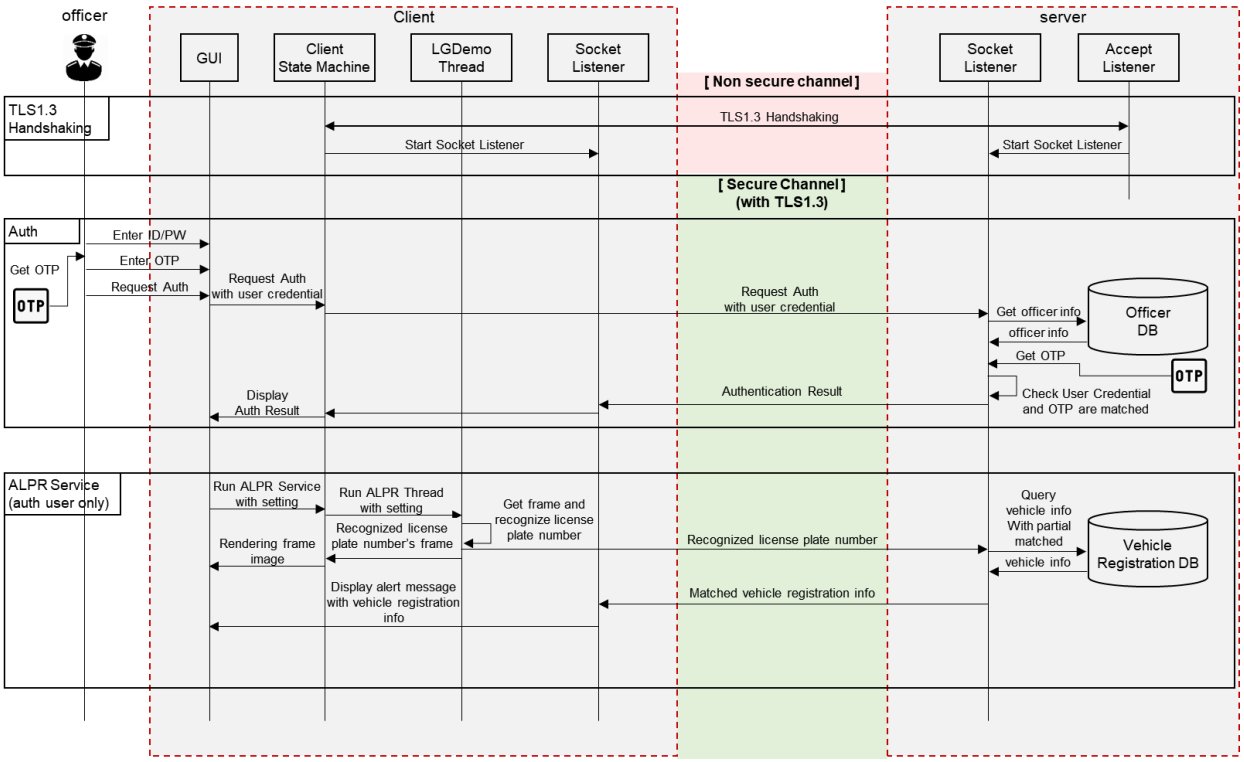


Figure 7 System Service Sequence

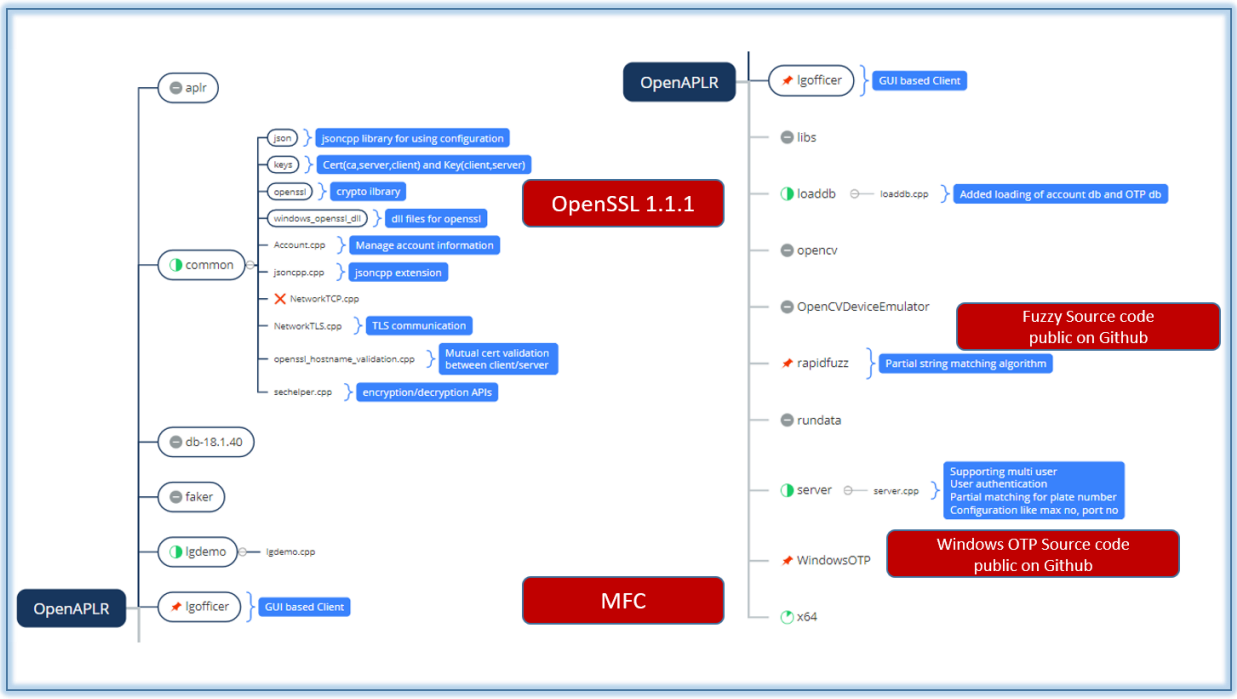


Figure 8 Source Tree

5.2 Client Architecture

5.2.1 Client State Diagram

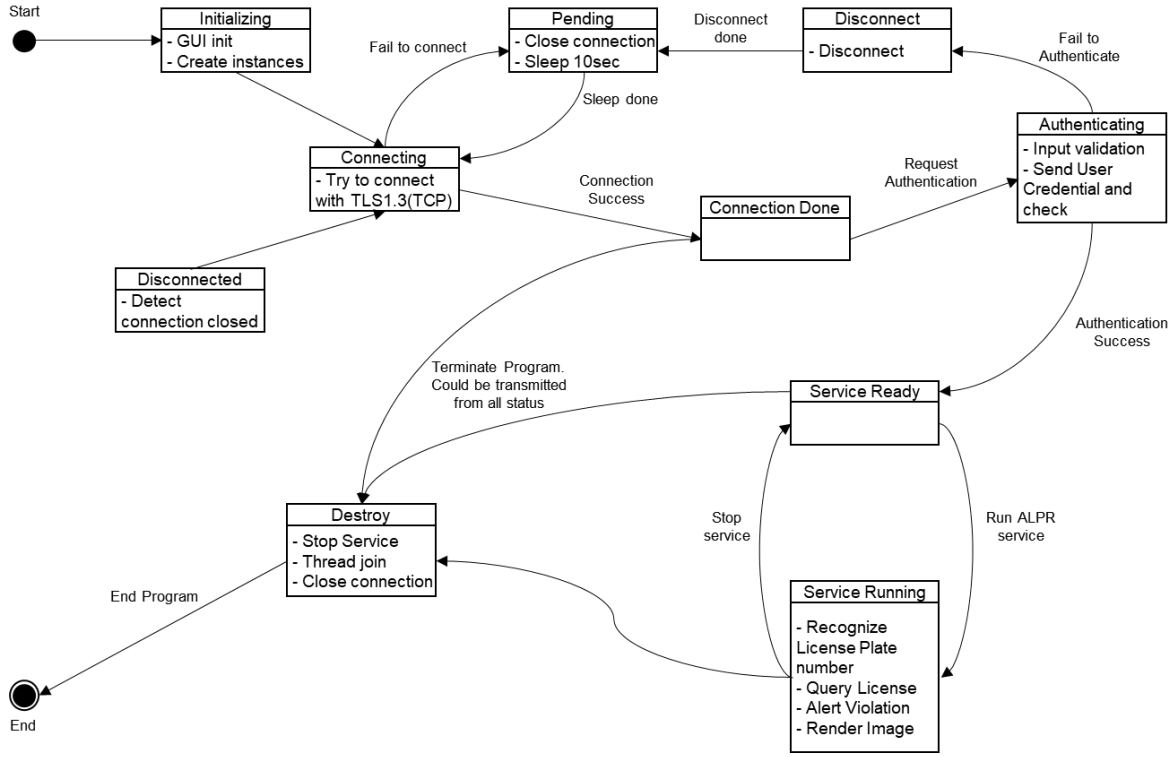


Figure 9 Client State Diagram

6 Mitigation

6.1 2 Factor Authentication.

6.1.1 Basic Concept

Figure 10 2FA Basic Concept

6.1.2 Cross-verifies users with two different forms of identification

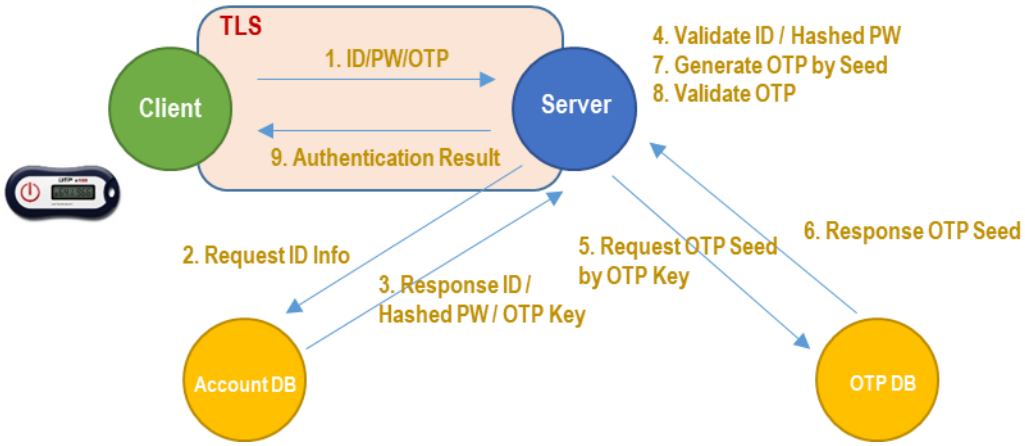


Figure 11 Cross-verifies users

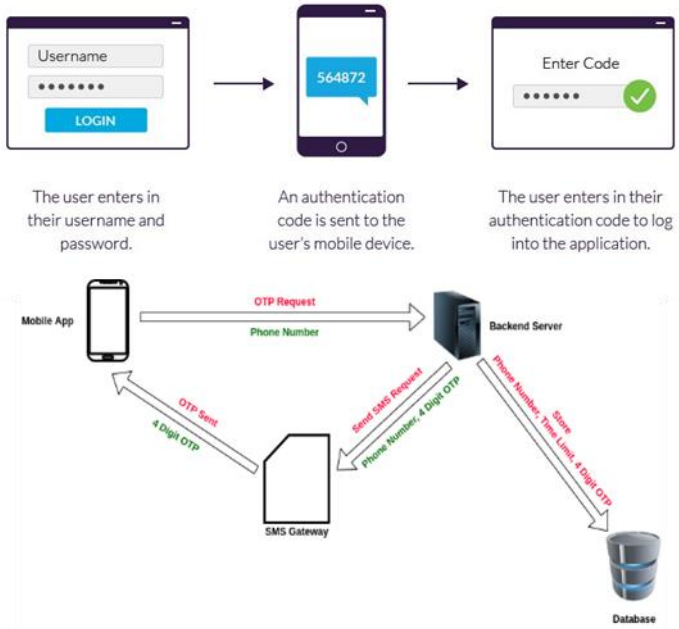


Figure 12 OTP identification

6.1.3 2FA Basic Scenario

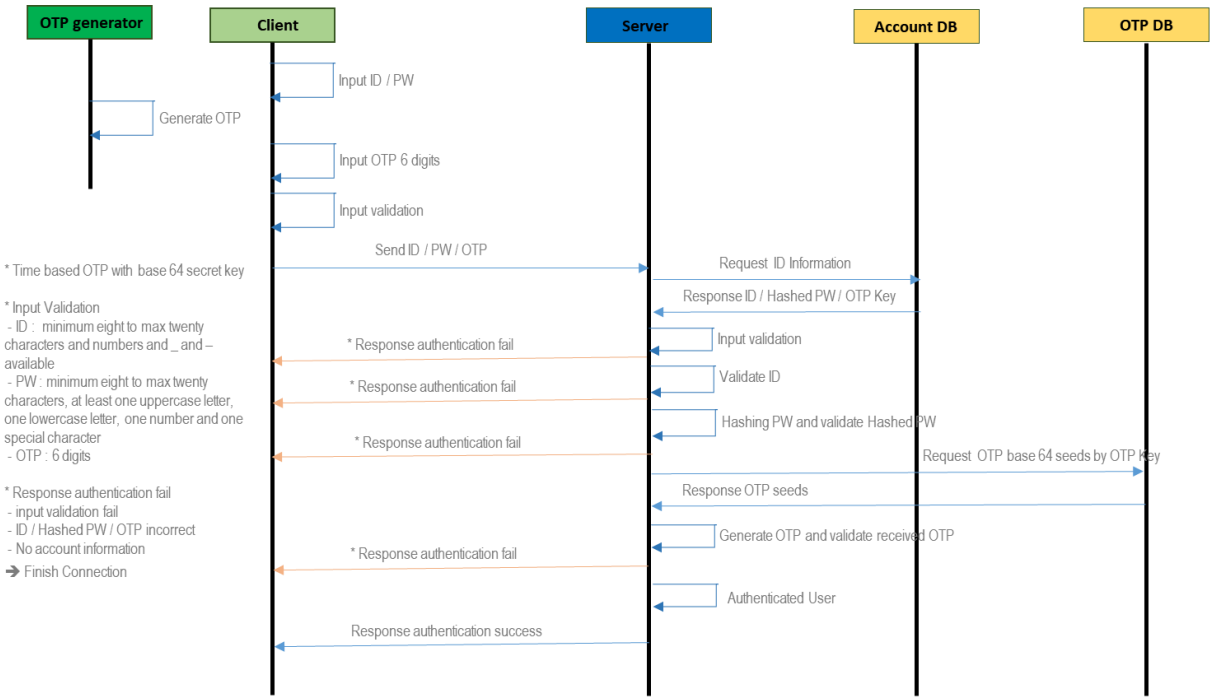


Figure 13 2FA Basic Scenario

6.2 TLS/SSL

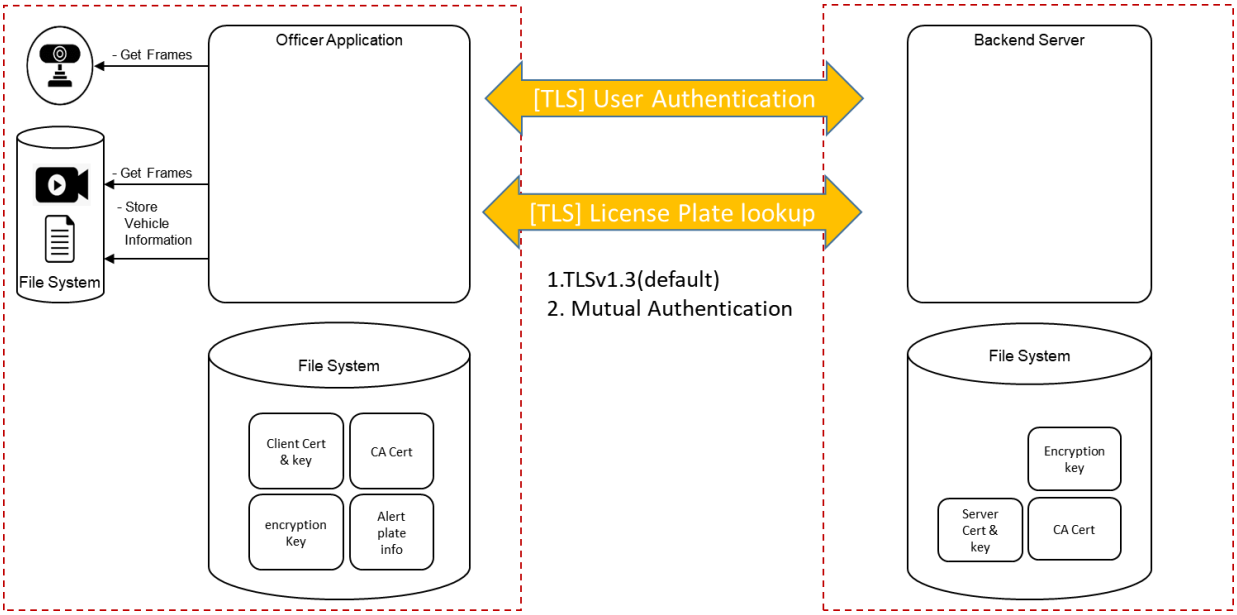


Figure 14 TLS/SSL

6.3 Input Validation

6.3.1 Possible Attack Cases

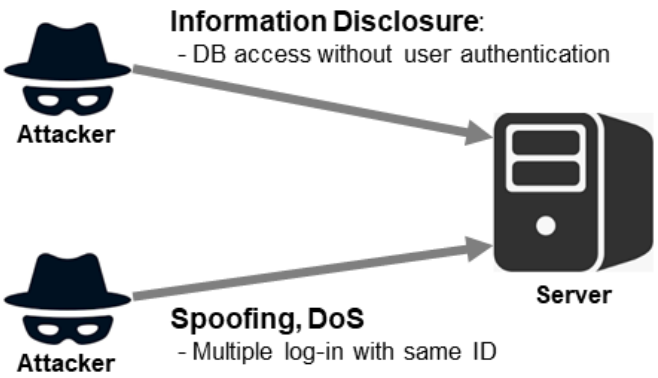


Figure 15 Possible Attack Cases

6.3.2 Mitigation

- Keep state consistency between server and client

- Prevent multiple log-in with same ID

- Limit the number of concurrent connected client

7 Implementation

7.1 Build Project

7.1.1 Pre-Condition

1) Copy "opencv" directory and all files in it to the root directory of "OpenAPLR_Team4" project. (You can find it from the original source codes (OpenAPLR-3.1.1_Vs2022/opencv))

2) Install MFC to visual studio as below picture.

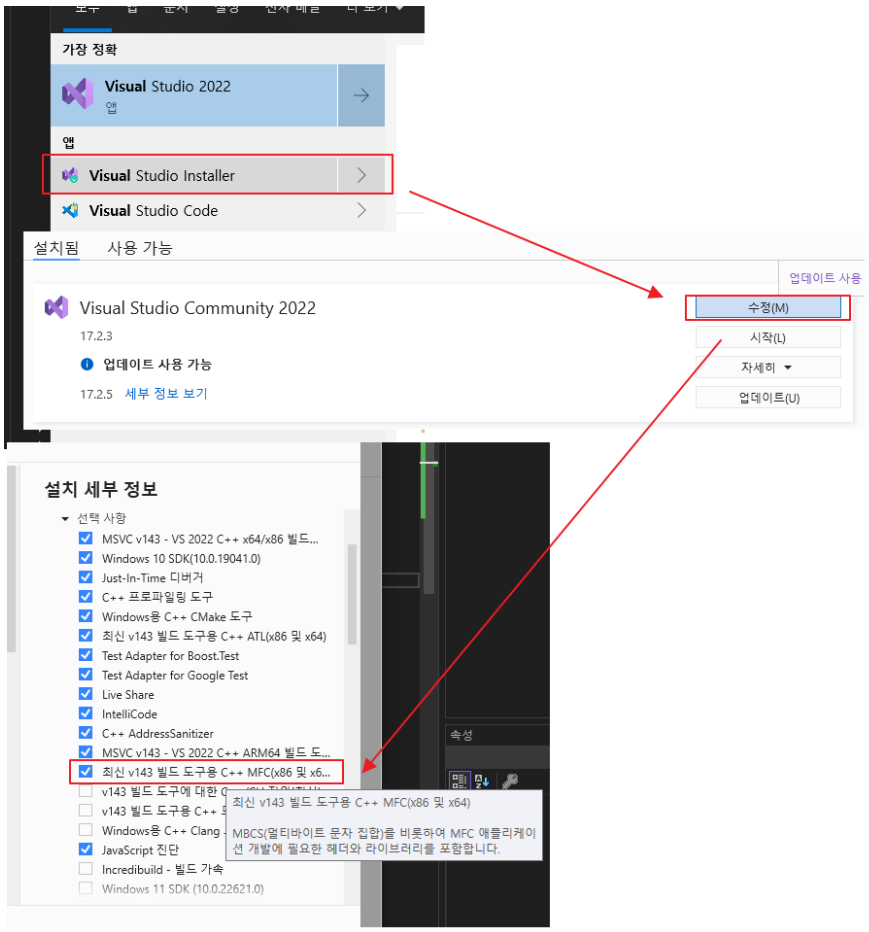


Figure 16 Pre-Condition for build

7.1.2 Build

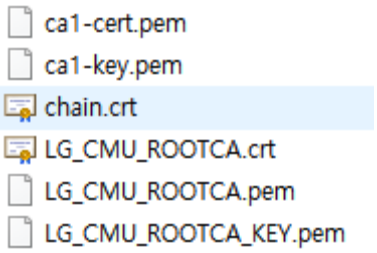
1) Open "OpenALPR.sln" solution file with visual studio
2) Make sure "Release/x64" is selected as a build configuration.
3) Execute "solution build"

7.2 Execute Program

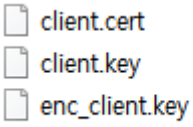
7.2.1 Pre-Condition

1) Execute "setup.bat"
   - "setup.bat" will move all files required to run, including key and cert to "x86/release".

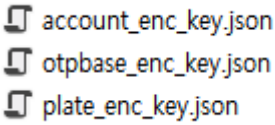
2) Check if all files described below exist normally.
   - "x64/release" directory
     - configuration files : server-conf.json, client-conf.json
     - db data files : db/account.db, licenseplate.db, otpbase.db
     - "keys" directory
       - "ca" directory



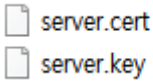
- "client" directory



- "encKey" directory



- "server" directory



7.2.2 Run

If you want to run server or client on other windows computer, Copy "Release" in x64 directory directory to the target device.

1) Run Server
   - Execute "server.exe"(located in x64/Release) before running "lgofficer" as a client.

2) Run Client
   - Excute lgofficer
     - If the client and server are running on different devices,
       ① check ip address of server.
       ② open x64/Release/client-conf.json then modify server_ip.
   - Execute OTPGen Application (See the "OTPGen" directory)
     - Let's try User #6 (one of several users in the account database) in this guide.
     - Run "OTPGen_SecurityPolice_006.exe" and you can see the otp number and it ill be changed periodically.

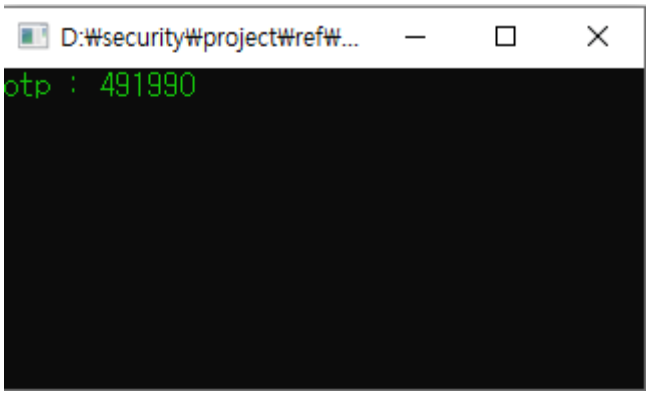


Figure 17 OTP Generator

- Pass Log-in information on the "lgofficer" application.
  - For blank text box, fill in like below.

    ID : SecurityPolice_006

    PW : !dlwormsS4Best

    OTP : Check the number in the OTP program that has already been run.

- Start process
  - If user authentication is succeeded, Start button will be enabled like below.
  - Push the "Start" button.

7.3 Test

Table 47 Test Cases

SRS ID	TC ID	TC Name	Procedure	Expected	Result	Note
REQ_CLI_01	TC-01	User login with two factor authentication	1. run lgofficer.exe 2. Check status in lgofficer diaglog 3. Insert ID/PW and OTP number from OTP generator which is dedicated to each user 4. Click Login Button 5. Check login success message in message box	2. Need to Auth 5. login - success	Pass	
REQ_CLI_02	TC-02	Handle lost or compromised credentials	1.run lgofficer.exe 2. Click Find PW below OTP edit control 3. Call to IT service	2. Popup message is displayed	Pass	IT service's number is virtual number, do not call

REQ_CLI_03	TC-03	select and save retrieved information locally.	1. Select radio buttons for media, save policy and resoultion for ALPR service 2. Click Start Button 3. Check below combination 3-1-1) Playback/NoSave/640x480 3-1-2) Playback/Save/1280x720 3-1-3) Playback/Save No ALPR/640x480	3-1-1) Run Playback. and recognize certain license plate number and there is no output.avi 3-1-2) Run Playback. and recognize certain license plate number and check output.avi which is saved 3-1-3) Run Playback. and do not recognize certain license plate and check output.avi which is saved and recognized information is not displayed in video	Pass	
REQ_CLI_04	TC-04	secure communicatio n(TLS1.3)	1. Run lgofficer.exe 2. Check status in lgofficer diaglog 3. Check Wireshark packet to verify TLS1.3 establishment 4. Input user id/pw/OTP(refer to TC-10) and click login button to check user credential is encrypted or not	2. Need to Auth 3. Check TLS1.3 handshaking Client Hello Server Hello, Change Cipher Spec, Change Cipher Spec, 4. User Credentail is not exposed. we can only see Application Data Packet	Pass	
REQ_CLI_05	TC-05	Recognize license plate number	1. Start ALPR service with Live/Playback/Image	1. Check recognized number in "Recognized Image" section's image	Pass	
REQ_CLI_06	TC-06	ALPR function maintaining a frame rate of at least 25fps.	1. Start ALPR service with playback and check displayted fps text	1. Check fps	Pass	1. If PC's has poor CPU performance, fps could be lower.
REQ_CLI_07	TC-07	query the backend license plate through server's db	1. Start ALPR service with playback 2. If when detecting license number which registered in server plate DB, wait for alert message is displayed	2. Alert Message with vehicle registeration information is displayed	Pass	
REQ_CLI_08	TC-08	If a license plate does not generate an alert, then the user interface must display the	1. Start ALPR service with playback 2. If when detecting license number which registered in server plate DB, display recognized frame image 3. if next recognized license is not registed in DB, keep previous recognized image in UI	2. Recognized Image 3. Previous recognized image	Pass	

		last recognized plate image				
REQ_CLI_09	TC-09	provide the current camera /playback view.	1. Start ALPR service with camera/playback 2. Check new window for current carmera/playback view	2. new window for current carmera/playback view is displayed	Pass	
REQ_CLI_10	TC-10	display computed camera / playback frames per second, average time per frame, jitter and frame number	1. Start ALPR service with camera/playback 2. Check fps, average time per frame, jitter, frame number	2. Check fps, average time per frame, jitter, frame number on bottom of recognized image	Pass	
REQ_CLI_11	TC-11	The system should allow the officer to choose between using a live camera and playback file in the UI.	1. Select radio buttons for media, save policy and resoultion for ALPR service 2. Click Start Button 3. Check below combination 3-1-1) Live/Save/1280x720 3-1-2) Playback/Save (Need to select video file)	3-1-1) Run Live Camera with 1280x720 resolution. and recognize certain license plate number and check output.avi which is saved 3-1-2) Run playback video and recognize licence plate in video and check output.avi which is saved	Pass	
REQ_CLI_12	TC-12	detect network connectivity issues with the backend server within 5 seconds and automatically resolve the communicatio n	1. Terminate server.exe and check status in UI 2. Restart server.exe as soon as possible wihin 5sec 3. Check status	1. Disconnected 3. Need to Auth (it means retry connection is succeed)	Pass	

REQ_CLI_13	TC-13	alert communication errors or failures.	1. Terminate server.exe and check message box	1. disconnect! message is printed	Pass	
REQ_CLI_14	TC-14	The system must fetch vehicle information in no more than 10 seconds as officers are often making queries in real time.	1. Start ALPR service with camera/playback 2. Check time duration by wireshark pakcet time diff = recevie time - send time	2. check do not exceed 10sec	Pass	vehicle information is recieved only if there is (partial)matched license number
REQ_SVR_01	-	Support license plate queries.				Ref to TC-08
REQ_SVR_02	-	secure communication(TLS1.3)				Ref to TC-05
REQ_SVR_03	-	Authenticate remote laptop users.				Ref to TC-01/TC-05
REQ_SVR_04	TC-16	Support multiple users.	1. run logofficer.exe and login with user1 2. run logofficer.exe and login with user2	1./2. user1,2 are authenticated	Pass	Server supports max 10 users
REQ_SVR_05	TC-17	the best match license plate if there is not an exact match that includes a configurable minimum confidence threshold to support a partial match.	1. Start ALPR service with camera/playback	1. If when recieve license query from client and there is no exact match Displayed like below in server's console log Payload=LKY136D, Plate Number=LKY1360, Matching Rate=85.71%	Pass	threshold is 70%

REQ_SVR_06	TC-18	query statistics	1. Start ALPR service with camera/playback	1. query statistics is displayed every 5sec User[SecurityPolice_006] average queies per sec : 0.2000	Pass	
REQ_SVR_07	TC-19	partial match statistics	1. Start ALPR service with camera/playback	1. If when recieve license query from client and there is no exact match Displayed like below in server's console log Payload=LKY136D, Plate Number=LKY1360, Matching Rate=85.71%	Pass	
REQ_SVR_08	TC-20	Configuable values for server setting	1. oepn server-conf.json 2. change port to 2223 3. run server.exe 4. run lgofficer.exe 5. check status	5. loop below status Conecting Wait...(10sec)	Pass	

8 Seeded Vulnerabilities

8.1 VU01 - Null pointer dereference could have occurred when an alert message is written in a log file after service is stopped

Table 48 Seeded Vulnerability 01

Description	[VU01] Null pointer dereference could have occurred when an alert message is written in a log file after service is stopped.
STRIDE	Denial of Service
Detail	socket listener is always on and displaying and writing an alert message is could be done regardless of it state "file pointer of the alert message file is opened when running the service and closed when stopping the service " it is a synchronization issue. stop first, the file pointer is closed then an alert message is came in this case, null pointer dereferencing could have occurred
PoC	
1. load network traffic on server to increase packet latency(ping, iperf, etc)	

2. login and start ALPR service with playback video(beaver1.avi)
3. try to stop as soon as license recognition packet query is sended
4. iterate until application is crashed. The frequency of occurrence is about once among 20 attempts (5%)
5. application crash is occurred due to absent of null pointer check
do {
written += fwrite(payload, 1, remain, fp);
remain -= written;
}
while (written < sizeof(payload));

CVSS Score	*CVSS Base Score: 3.3* *Impact Subscore: 1.4* *Exploitability Subscore: 1.8* *CVSS Temporal Score: 3.3* *CVSS Environmental Score: 3.3* *Modified Impact Subscore: 1.4* *Overall CVSS Score: 3.3* *Severity : Low*	Source Code	ClientMachine.cpp line : 757~761 line : 1064~1070
Consequence / Impact			
It cause denial of service in client application due to system crash			

8.2 VU02 - Backdoor - ID/PW based on factorization of big number

Table 49 Seeded Vulnerability 02

Description	[VU02] Backdoor - ID/PW based on factorization of big number
STRIDE	Elibation of Privilege
Detail	- Backdoor for Developer ID = (Prime A) * (Prime B) PW = Prime A or Prime B ID/PW Authentication : if (ID / PW) == (Integer), then Authenticate For the number to be used as the ID, choose a number large enough to take a week or two with the current NotePC computing power.
PoC	
1. Attacker can check the ID for backdoor by analyzing shared codes. 2. Attacker can check the procedure of backdoor authentication by using shared codes. 3. Attacker gets the password by factorization of ID number 4. Attacker can infer the order of two prime numbers by considering the order of the ID number, and	

proper prime numbers can be selected sequentially by using - Brute-Force - A list of known prime numbers (http://compoasso.free.fr/primelistweb/page/prime/liste_online_en.php)			
CVSS Score	*CVSS Base Score: 5.9* *Impact Subscore: 3.6* *Exploitability Subscore: 2.2* *CVSS Temporal Score: 5.6* *CVSS Environmental Score: 7.3* *Modified Impact Subscore: 5.4* *Overall CVSS Score: 7.3* *Severity : High*	Source Code	server.cpp line : 411~440
Consequence / Impact			
An attacker without access rights can gain access to the system			

8.3 VU03 - Tampering configuration file results in TCP connection (not TLS)

Table 50 Seeded Vulnerability 03

Description	[VU03] Tampering configuration file results in TCP connection (not TLS)		
STRIDE	Tampering		
Detail	- tcp connection is conducted through OpenTcpConnection function which we implemented - this function has argument const char* ca_pem, const char* cert_pem, const char* key_pem from configure file but when fail to read CA path from configure file, it returns NULL - when NULL is passed, it tries to TCP connection rather than TLS - it results in non-secure channel establishment		
PoC			
1. restart the server by deleting the ""server_cert_path"", ""server_key_path"", ""ca_cert_path""value to the server`s config file 2. On the server side, it changes from tls socket communication to unprotected socket communication.			
CVSS Score	*CVSS Base Score: 5.7* *Impact Subscore: 5.2* *Exploitability Subscore: 0.5* *CVSS Temporal Score: 5.4*	Source Code	NetworkTLS.cpp line :261~263 line : 542~708 line : 745~758

	CVSS Environmental Score: 6.1 *Modified Impact Subscore: 5.9* *Overall CVSS Score: 6.1* *Severity : Medium*		line : 780~828 line : 851~860 server.cpp 729~740
Consequence / Impact			
Mutual authentication based on PKI certificates is not supported.			

8.4 VU04 - Sniffing ID/Password/OTP/Plate Information

Table 51 Seeded Vulnerability 04

Description	[VU04] Sniffing ID/Password/OTP/Plate Information		
STRIDE	Information Disclossure		
Detail	- The communication channel for the user credentials is not secure by changing server and client configuration settings		
PoC			
1. restart the server by deleting the ""server_cert_path"", ""server_key_path"", ""ca_cert_path""value to the server`s config file 2. On the server side, it changes from tls socket communication to unprotected socket communication.			
CVSS Score	*CVSS Base Score: 5.7* *Impact Subscore: 5.2* *Exploitability Subscore: 0.5* *CVSS Temporal Score: 5.4* *CVSS Environmental Score: 6.1* *Modified Impact Subscore: 5.9* *Overall CVSS Score: 6.1* *Severity : Medium*	Source Code	NetworkTLS.cpp line :261~263 line : 542~708 line : 745~758 line : 780~828 line : 851~860 server.cpp 729~740
Consequence / Impact			
The socket communication data is not protected and it becomes possible to sniff for plate number and driver's license information.			

8.5 VU05 - Problem of certificate validation

Table 52 Seeded Vulnerability 05

Description	[VU05] Problem of certificate validation		
STRIDE	Spoofing		
Detail	-RootCA doesn't be used for CA certificae validation. So, if server key, cert and CA cert are chaged, a spoong attack is possible		
PoC			
1. generate server key, certificate and self-signed CA certificate for attack 2. replace the new generated server private key, certificate and ca certificate 3. try to connect TLS with the new forged certificate. 4. TLS channel is successfully established			
CVSS Score	*CVSS Base Score: 8.1* *Impact Subscore: 5.2* *Exploitability Subscore: 2.8* *CVSS Temporal Score: 7.7* *CVSS Environmental Score: 7.1* *Modified Impact Subscore: 5.9* *Overall CVSS Score: 7.1* *Severity : High*	Source Code	NetworkTLS.cpp 206 line
Consequence / Impact			
the certificate chain must be validated by rootca			

8.6 VU06 - Buffer overflow of Alert.log data

Table 53 Seeded Vulnerability 06

Description	[VU06] Buffer overflow of Alert.log data		
STRIDE	Spoofing		
Detail	- the maximum buffer size is 32768 - if the alert.log size over the maximum, the client is abnormally terminated.		
PoC			
1. make a big size(over 33kb) alert.log() file on the Release folder. 2. push the button ""Encrypt""" 3. lgofficer will be died and restarted.			
CVSS Score	*CVSS Base Score: 3.3* *Impact Subscore: 1.4* *Exploitability Subscore: 1.8* *CVSS Temporal Score: 3.2*	Source Code	lgofficerDig.cpp 435~460line sechelper.cpp 342~349 line

	CVSS Environmental Score: 3.2 *Modified Impact Subscore: 1.4* *Overall CVSS Score: 3.2* *Severity : Low*		sechelper.h 25,50 line
Consequence / Impact			
attacker can use this vulnerability for DoS attack			

8.7 VU07 - Display id and password in the log of server-side

Table 54 Seeded Vulnerability 07

Description	[VU07] Display id and password in the log of server-side		
STRIDE	Information Disclossure		
Detail	- Admin account which skips 2FA can be used to neutralize 2FA. - ID and password of admin account cab be checked via over-the-shoulder sniffing. - Attacker can use dictionary attack as well.		
PoC			
1. Obtain administrator account information(ID/Password) via over-the-shoulder or dictionary attck or restart the server by adding the { ""debug"" : ""true""} value to the server`s config file 2. Login through the administrator`s id/password in the client side 3. Login succeeds without 2 authentication factor by OTP.			
CVSS Score	*CVSS Base Score: 4.9* *Impact Subscore: 3.6* *Exploitability Subscore: 1.2* *CVSS Temporal Score: 4.7* *CVSS Environmental Score: 6.3* *Modified Impact Subscore: 5.4* *Overall CVSS Score: 6.3* *Severity : Medium*	Source Code	server.cpp line : 230~232 line : 352~356
Consequence / Impact			
Two authentication factor using OTP is disabled in the user authentication process. so attacker is possible to get plate number and driver's license information.			

8.8 VU08 - No input validation method for partial matching threshold in the configuration file

Table 55 Seeded Vulnerability 08

<table>
<tr><td>Description</td><td colspan="3">[VU08] No input validation method for partial matching threshold in the configuration file</td></tr>
<tr><td>STRIDE</td><td colspan="3">Tampering, Denial of Service</td></tr>
<tr><td>Detail</td><td colspan="3">- Attacker can manipulate the partial matching value in the config file to not be detected illegal vehicles.
- If the partial matching threshold is very high then every vehicle wouldn’t be detected even if it is the illegal vehicle</td></tr>
<tr><td colspan="4">PoC</td></tr>
<tr><td colspan="4">1. Access the config file that manages server settings
2. Modify the ""num_thr_partial"" value to 100 or more to set the part recognition criteria of the license plate
3. Restart the server to apply the config.
4. Even if it is a vehicle that violates the license plate, the license plate cannot be matched any more.</td></tr>
<tr><td>CVSS Score</td><td>CVSS Base Score: 4.4
Impact Subscore: 3.6
Exploitability Subscore: 0.8
CVSS Temporal Score: 4.3
CVSS Environmental Score: 6.1
Modified Impact Subscore: 5.4
Overall CVSS Score: 6.1

Severity : Medium</td><td>Source Code</td><td>server.cpp
line : 721~724</td></tr>
<tr><td colspan="4">Consequence / Impact</td></tr>
<tr><td colspan="4">It is possible to avoid enforcement by interfering with illegal vehicle detection in the server for the license plate recognized by the client.</td></tr>
</table>

8.9 VU09 - Weak point of protocol about packet command processing

Table 56 Seeded Vulnerability 09

Description	[VU09] Weak point of protocol about packet command processing
STRIDE	Elibation of Privilege
Detail	- Attacker can make and send the pseudo packet pretending the normal communication.

<table>
<tr><td></td><td colspan="3">- Currently, ‘3’ cmd is not used for any purpose, but can be used to neutralize the user authentication procedure.</td></tr>
<tr><td colspan="4">PoC</td></tr>
<tr><td colspan="4">make and send the pseudo packet pretending the normal communication.</td></tr>
<tr><td>CVSS Score</td><td>CVSS Base Score: 5.9
Impact Subscore: 3.6
Exploitability Subscore: 2.2
CVSS Temporal Score: 5.8
CVSS Environmental Score: 7.5
Modified Impact Subscore: 5.4
Overall CVSS Score: 7.5

Severity : High</td><td>Source Code</td><td>server.cpp
Should have put ""break"" in next line of 313

Should have put like below in next line of 314
if (!(PacketCmd == 2 || PacketCmd == 3))
{
printf(""Rx invalid cmd : cur stat(%d) cmd(%d)\n"", PerUserData->state, PacketCmd);
goto free_resource;
}</td></tr>
<tr><td colspan="4">Consequence / Impact</td></tr>
<tr><td colspan="4">After making communication session, if attacker sends ‘3’ cmd pretending normal packet, server can receive ‘2’ cmd for getting plate number information.</td></tr>
</table>

8.10 VU10 - Tampering with the file that is used for storing alert information (TOCTOU)

Table 57 Seeded Vulnerability 10

<table>
<tr><td>Description</td><td colspan="3">[VU10] Tampering with the file that is used for storing alert information (TOCTOU)</td></tr>
<tr><td>STRIDE</td><td colspan="3">Tampering</td></tr>
<tr><td>Detail</td><td colspan="3">- Client application saves vehicle information to a text file for illegal vehicles
- Attacker can modify or delete text file by TOCTOU attack.</td></tr>
<tr><td colspan="4">PoC</td></tr>
<tr><td colspan="4">1. Access the file path that stores the illegal vehicle detection history in the client
2. Open the file and modify or delete the detected list.</td></tr>
<tr><td>CVSS Score</td><td>CVSS Base Score: 7.1
Impact Subscore: 5.2</td><td>Source Code</td><td>ClientMachine.cpp
line : 1064~1070</td></tr>
</table>

	Exploitability Subscore: 1.8 *CVSS Temporal Score: 6.9* *CVSS Environmental Score: 7.6* *Modified Impact Subscore: 5.9* *Overall CVSS Score: 7.6* *Severity : High*		
Consequence / Impact			
Recognized contents are altered by deleting or manipulating files that store illegal vehicles information			

9 Phase-1 Lessons Learned

9.1 Lesson #1 - *Consider it in the early phases*

- ✓ **Importance of abstracting security requirement from initial phase.**
- ✓ **In the process of converting a program based on traditional TCP communication to TLS-based communication, many difficulties were encountered.**
- ✓ **This is an obvious technical debt, and it is a cost that would not have been incurred if TLS was considered from the initial requirements analysis stage.**

9.2 Lesson #2 - *Opportunity to learn and experience the secure development methodology*

- ✓ **Opportunity to learn the secure development methodology,**
  such as extracting requirements from asset, threat, risk analysis and applying secure coding.

9.3 Lesson #3 - *Having secondary security measures*

- ✓ **2FA offer better protection than passwords alone.**

9.4 Lesson #4 - *Consider security development process*

- ✓ **Organizations that do not implement a vulnerability management process are at significantly increased risk of experiencing a security incident.**

Phase 2 : Security Analysis of Classmate System

1 Phase 2 Introduction

We reviewed and analyzed Team 3's artifacts to identify assets and security goals, identified attack surfaces, and discovered vulnerabilities through design and code reviews. The vulnerabilities were then assessed and classified. In addition, methods for attacking each vulnerability were derived and verified.

Artifacts of Team 3 : https://github.com/kibongs/3team_studio_project

1.1 Phase 2 Goals

- Breaking Confidentiality
  - Steal the Credentials and Plate information
- Breaking Integrity
  - Pollute user DB
- Breaking Availability
  - Make the system unavailable

2 Design Review

This section describes a review of Team Purple's design documentation.

2.1 Security Goals

Table 58 Security Goals

ID	Statement
G-01	System should maintain confidentiality and integrity of databases when a system admin or an officer reads and writes databases.
G-02	The confidentiality of the network communication should be maintained when the server and a client communicate.
G-03	System needs to detect a breach of personal identifiable information of databases when an admin or an officer read and write databases.

2.2 Security Requirements

Table 59 Security Requirements

No	Preliminary Requirement	Component	Requirement Specification
SR1	All users accessing the DB of the system must be authenticated.	server	server authentication is commonly performed by submitting a username or ID and OTP of private information that only a given user should know.
SR2	The DB of the system must be encrypted	server	Use 128-bit AES to encrypt data in the user DB and the driver/car Information DB.
SR3	User permissions to access DB should be classified and managed.	server	only allow Read permission - Ensure that least-privileged accounts are used to connect to Database server - Sysadmin role should only have valid necessary users
SR4	The server must be capable of responding to a large number of queries.	server	Immediate recovery Deploy server as a container (fault tolerance)
SR5	Communication history management using communication channel between client and server should be performed.	server & client	Ensure that auditing and logging is enforced on the application The client application records and saves requests sent to the server in a file.
SR6	The query statement through the communication channel between the client and the server must be verified.	server & client	Perform input validation and filtering on all string type Model properties All the input parameters must be validated before they are used in the application to ensure that the application is safeguarded against malicious user inputs. Validate the input values using regular expression validations on server side with a allowed list validation strategy. Unsanitized user inputs / parameters passed to the methods can cause code injection vulnerabilities.
SR7	The communication channel between the client and the server must be encrypted.	server & client	Enable HTTPS - Secure Transport channel The application configuration should ensure that HTTPS is used for all access to sensitive information.
SR8	The client system can only use digitally signed queries.	server & client	Apply authentication token

SR9	The client user should use a strong password.	client	client application support Proper Password Strength Controls. - Password limit: 8 to 20 characters - Password string limit: Must contain at least one lowercase letter, uppercase letter, and number - Password special characters: ~!@$%^&*(),<.>/?]+
SR10	The server must manage the digital signature.	server	Change the token generation key periodically.

2.3 System Architecture Design

2.3.1 Server

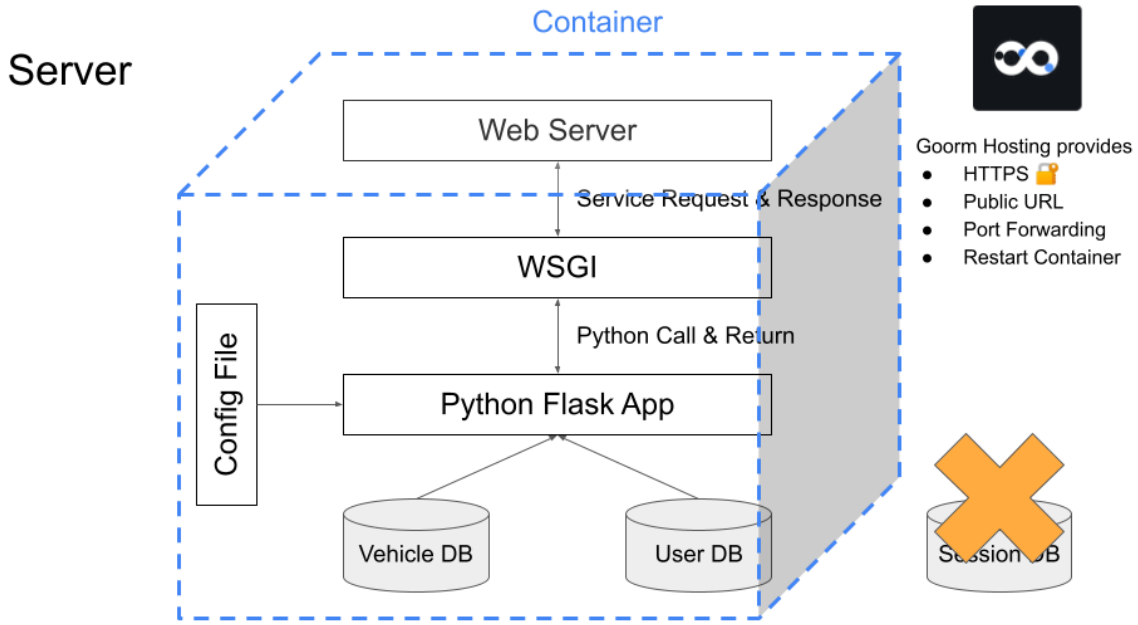


Figure 18 Server System Architecture Design

2.3.2 Client

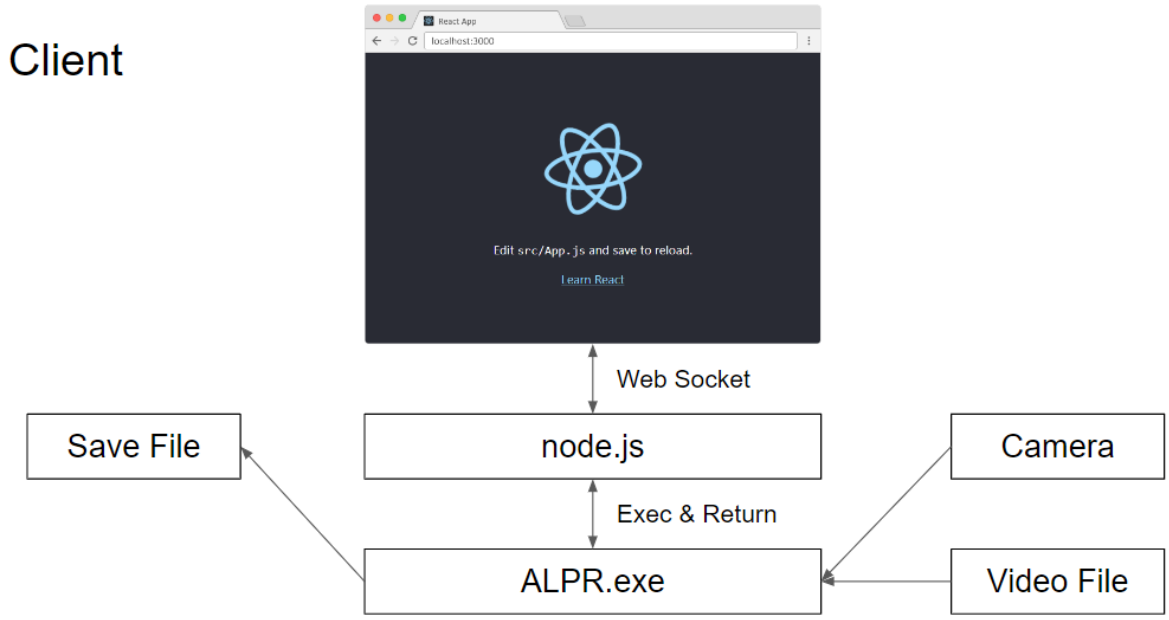


Figure 19 Client System Architecture Design

2.4 Software Architecture Design

2.4.1 Server

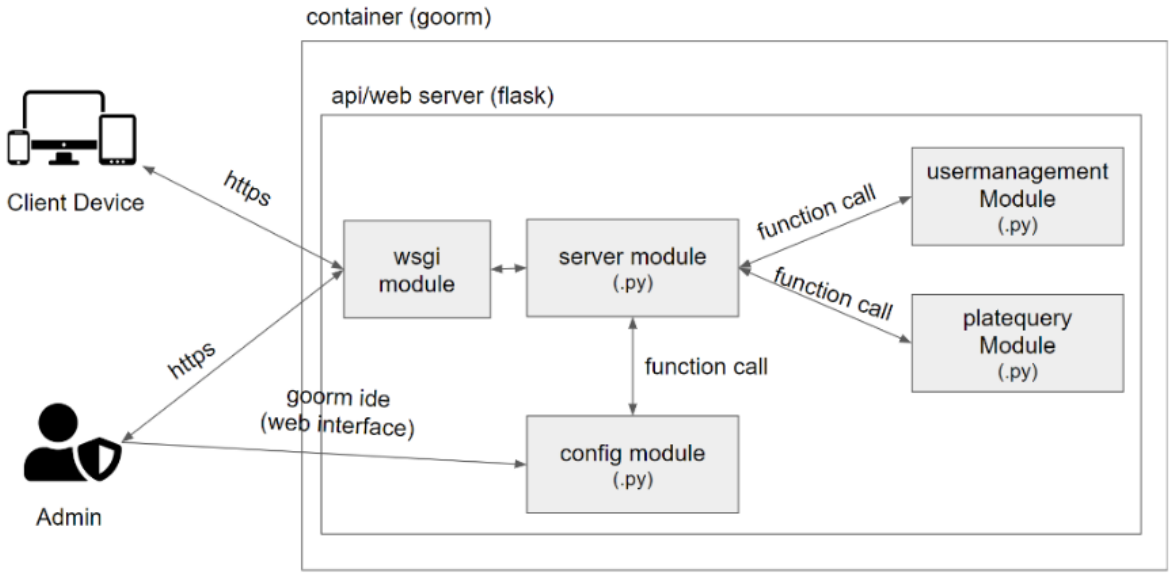


Figure 20 Server Software Architecture Design

2.4.2 Client

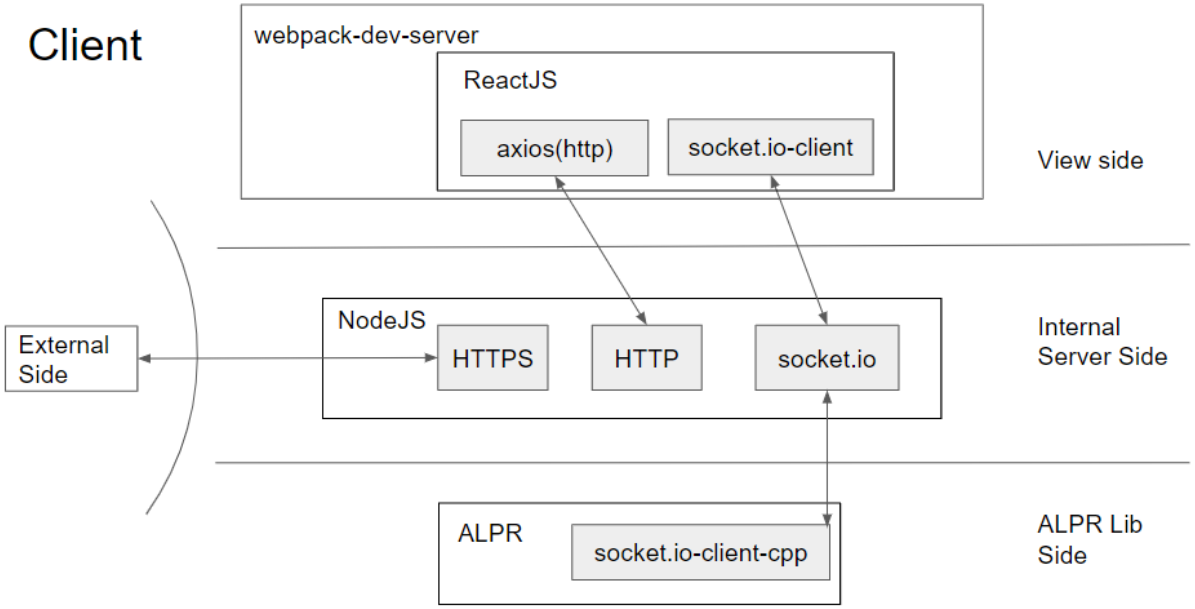


Figure 21 Client Software Architecture Design

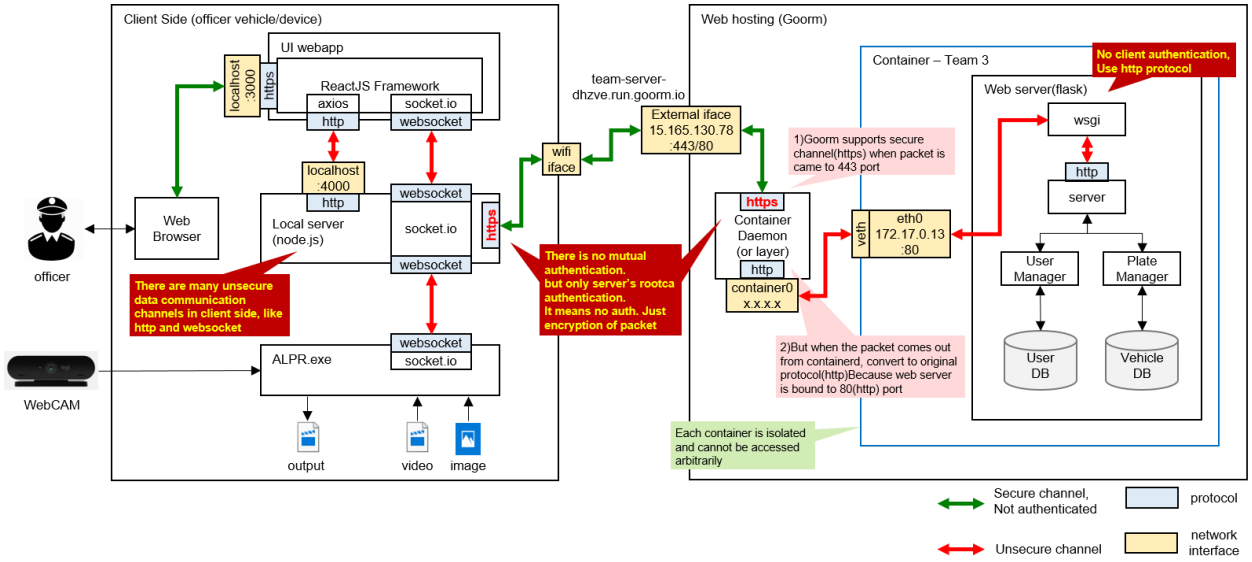
3 Analysis

This section describes source code analysis and static analysis to discover vulnerabilities in Team Purple's ALPR system.

3.1 Architecture Analysis

We analyzed the design and source code of Team 3's and drew the detailed architecture again. Through this process, the interface and protocol used for communication between each component were identified, and whether it was a secure channel or not was determined.

Based on this, we found the point and method to attack, and evaluated impact it had on the system by actually performing the attack.



- Pros
  - Remote web server is protected by container isolation
    → container protects data and prevents from illegal access
- Cons
  - On client side, there are many unsecured data communication channel (http, websocket)
    → Packets are easily sniffed by packet capturing tool
  - Web server use http protocol natively, and packets are wrapped with https (it is supported by Groom hosting service)
    → There is no authentication of server/client, it means that it is vulnerable to spoofing

3.2 Reconnaissance

- target : team-server-dhzve.run.goorm.io (remote server)
- there are two ports 80(http), 443(https)

Figure 22 Result of Reconnaissance

3.3 Static Analysis

In Team 3's System, most of the assets are located in the Server, and the Server is web-based. They designed a system similar to that used in commercial servers and developed mainly using JavaScript, so we could not use tools such as Flawfinder or Cppcheck, which are often used for static analysis of C/C++ source code.

We checked for known vulnerabilities in the source code using tools called DeepScan and sonatypelift.

3.3.1 DeepScan

✓ 6 issues, Poor grade

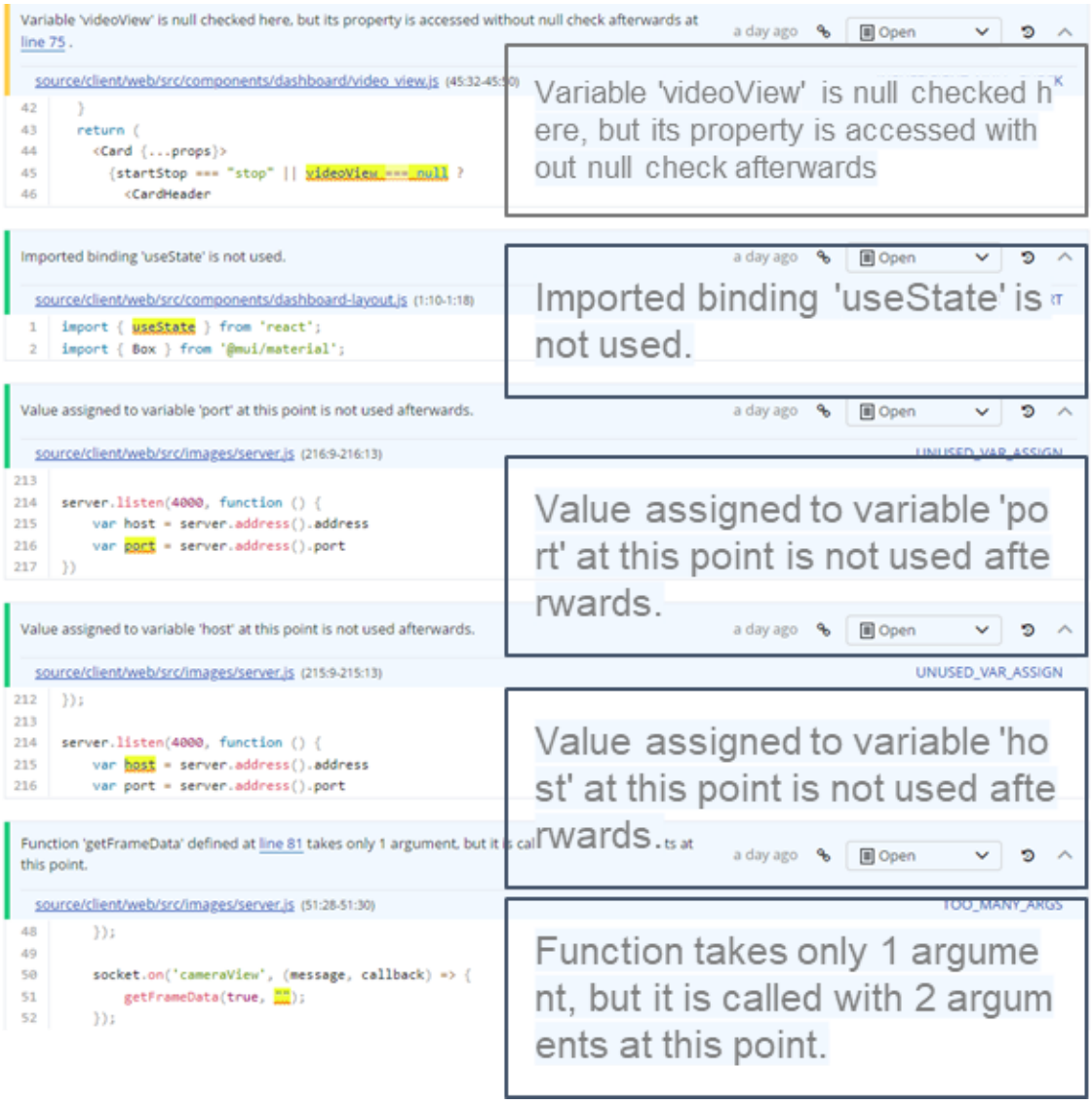


Figure 23 Result of DeepScan

3.3.2 sonatypelift

- ✓ 15 issues (8 critical / 7 severe)
- ✓ 13 components Violation

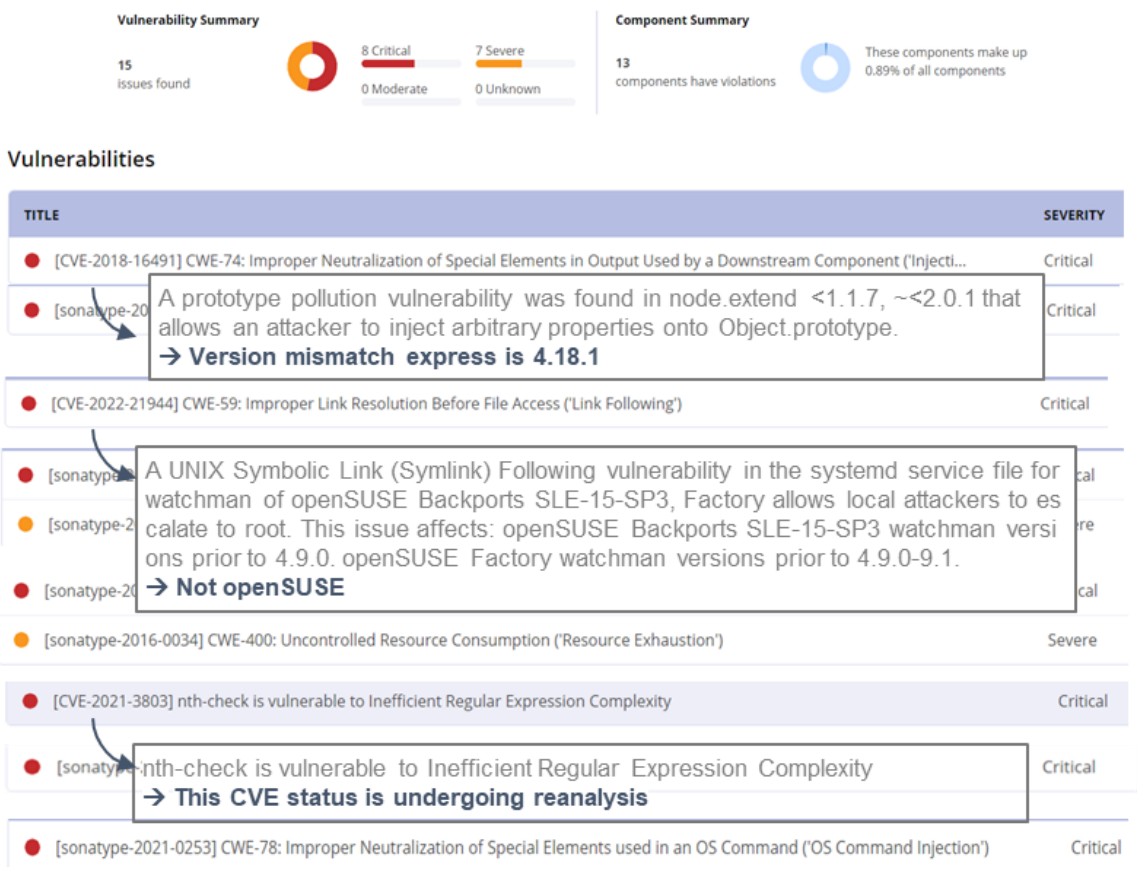


Figure 24 Result of sonatypelift

3.4 Fuzzing Test

3.4.1 Testing of Rest API using ZAP

We could not find any critical issue is Fuzz

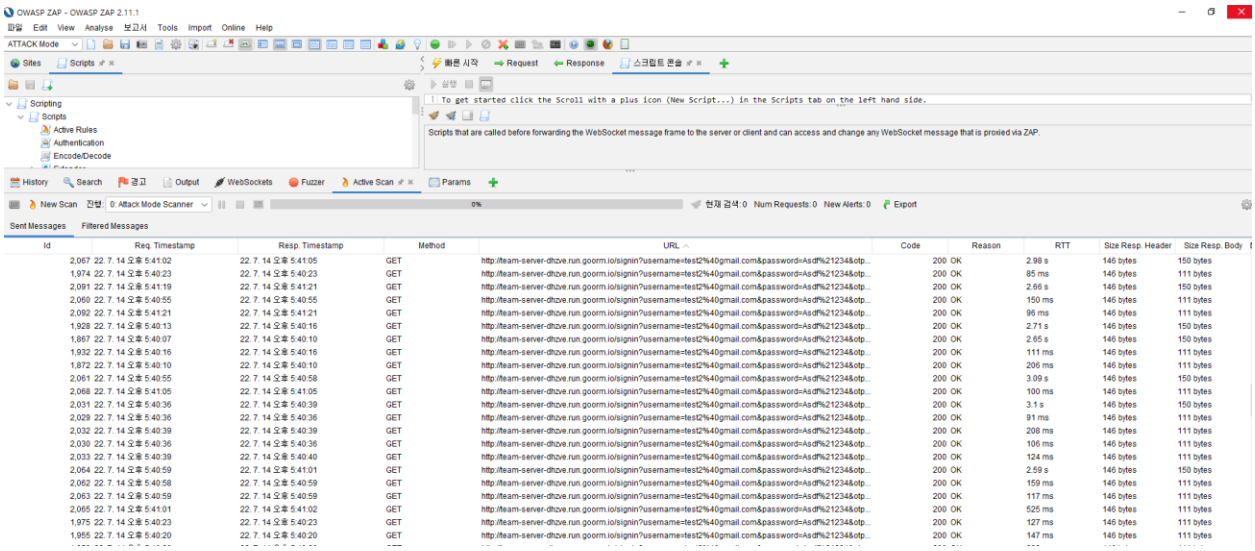


Figure 25 Result of Fuzz - login API

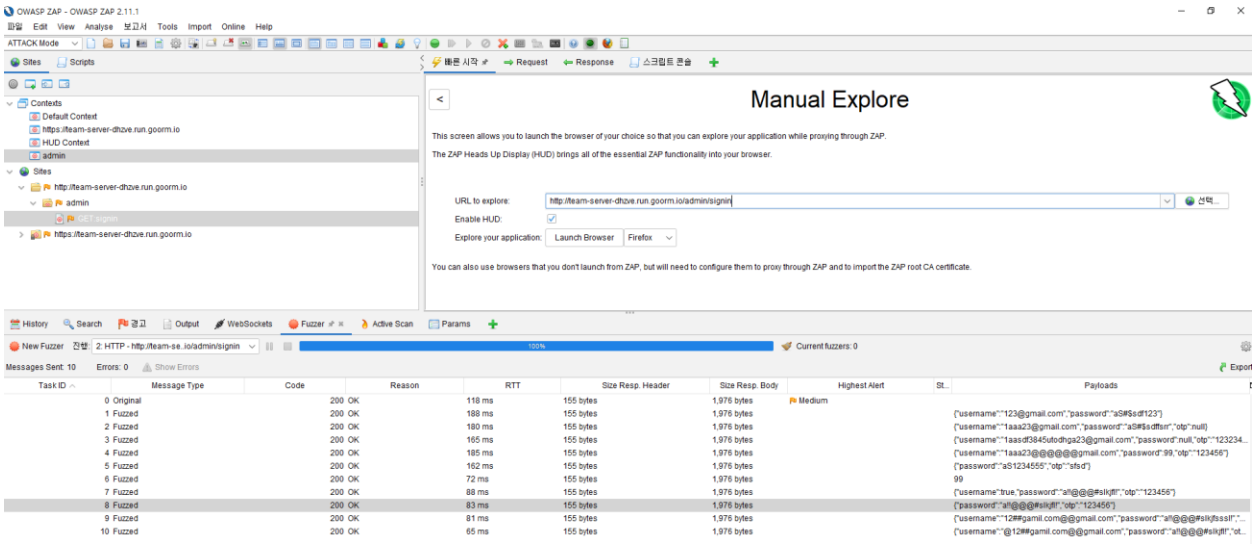


Figure 26 Result of Fuzz - admin login API

4 Vulnerabililty Analysis

This section describes

4.1 Analysis Criteria

4.1.1 Security Properties

[CONFIDENTIALITY] - compromises confidentiality
[INTEGRITY] - compromises integrity
[AVAILABILITY] - compromises availability

4.1.2 Approach (What we did to find the vulneravility)

manual
| [REVIEW/CODE] - done by reviewing code itself
| [REVIEW/DESIGN] - done by reviewing documents or code
| [FUZZING] - done by running fuzzing tools
v [STATIC] - done by running static analysis tools
toolly

4.1.3 Attack vector (pathway, what attacker obtained)

high opportunity
| [SERVERINFO] - can attack only with server ip, port
| [NETWORK] - can attack over the network communicating
| [CLIENT/ACCESS] - can attack with client accessible
| [CLIENT/SOURCE] - can attack with client source exposed
| [CLIENT/PRIVILEGED] - can attack with client accessible with privileged
| [SERVER/ACCESS] - can attack with server accessible
v [SERVER/SOURCE] - can attack with server source exposed
low opportunity

4.1.4 Exploit Techniques

[SQLINJECTION] - for by-passing authentication
[BUFFEROVERFLOW] - reading or writing beyond legitimate area
[WRAPAROUND] - making use of unsinged type wraparound
[FORMATSTRING] - mainly used for leaking data on the stack
[REVEALEDKEY] - decrypting secure data using revealed keys
[SNIFFING] - sniffing packets over the network
[SPOOFING] - so-called, man in the middle attack
[BRUTEFORCE] - trying all possible input until success
[CRAFTPACKET] - crafting and sending a customized packet
[TAMPERING] - modifying system components for a purpose
[WEBDEBBER] - sniffing credentials using web debugger.
[NOSPECIFIED] - no special techniques specified

4.2 Vulnerability Analysis

4.2.1 Vulnerability List

Table 60 Vulnerability List

STRIDE	VID	vulnerabilities	
		Client side	Server side
Spoofing	V08, V11, V13, V15, V16, V18, V22, V23, V24, V25, V26	• Server connection and plate information inquiry are possible with an spoofing client app (ex. jwt token) • using phishing of fake client web site, get a user credential • Available to change https to http by tempering the client's ser ver.js file.	• An attacker can retrieve plate information with a newly added user account by the stolen administrator credentials. • Remote server spoofing available - The external server provided by goorm does not supp ort mutual authentication using tls based on pki.
Tampering	V12, V13, V25, V26, V27	• A TOCTOU attack is possible because there is a time gap between file selection and file playback	• Illegal user account can be added by obtaining administrator privileges(ex. Admin cookie). - team-server-dhzve.run.goorm.io/admin/signup
Repudiation	V14, V21, V25, V26		• no log records for login or logout with an administrator or general account
Information Di sclosure	V01, V02, V03, V05, V06, V07, V09, V10, V15, V17, V18, V19, V20, V22, V23, V24, V25, V26	▪ User Credentials are disclosure by wireshark sniffing or web debugger tool, User Interface - User Name, Password, OTP, Token, Plate Information	• User names and OTP seeds are disclosure by anonymously accessible API - "team-server-dhzve.run.goorm.io/admin/log" - "team-server-dhzve.run.goorm.io/auth/<user>" • Admin cookie i s disclosure by wireshark sniffing or web debugger tool
DoS	V04	• App crash due to non input validation	
Elevation of P rivilege	V11, V12, V13, V16,	• User privileges can be obtained by using token	• Administrator privileges can be obtained by below API - team-server-dhzve.run.goorm.io/admin/signin

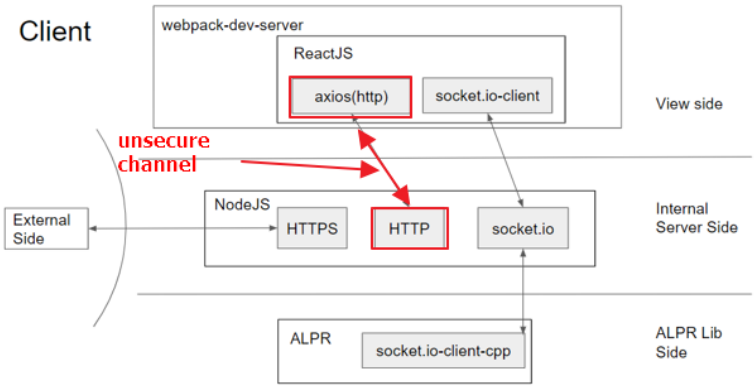
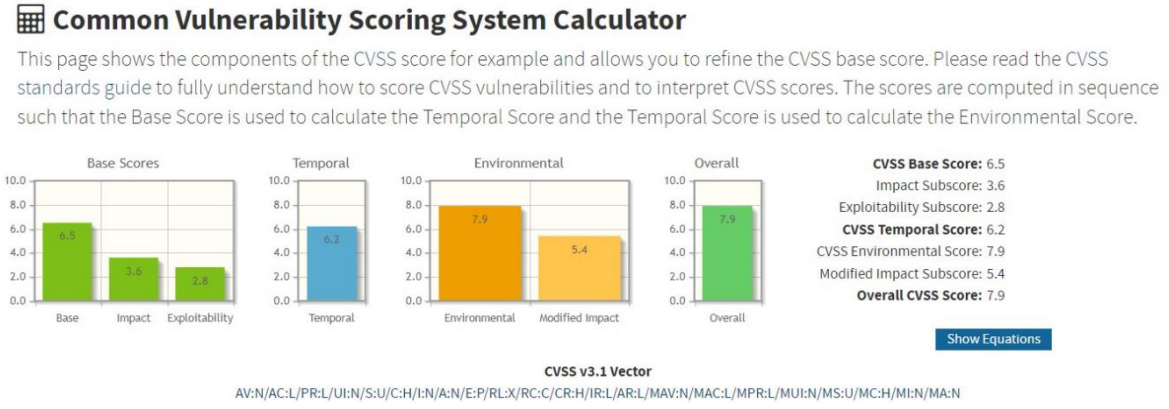
4.2.2 Recommended Mitigations

Table 61 Recommended Mitigations

STRIDE	VID	vulnerabilities	
		Consequences / Impact	Recommendations
Spoofing	V08, V11, V13, V15, V16, V18, V22, V23, V24, V25, V26	• An attacker can acquire the plate information • An Attacker can add new account arbitrarily • An attacker can steal admin or general account information by f ake client app • An attacker can steal admin or general account information by f ake remote server using DNS spoofing • An attacker can sniff all information by changing HTTPS to HTTP by tampering with the internal server of the client app	• Using the PKI with SSL/TLS • Using JWT token revoke mechanism • No multiple access with one account
Tampering	V12, V13, V25, V26, V27	• An attacker can pollute the user db by adding illegal users • An attacker can be possible to disturb normal plate inquiry by re placing the plate image	• Using the PKI with SSL/TLS • Using ACLs/permissions
Repudiation	V14, V21, V25, V26	• The attacker accesses the server with the stolen account, but the server does not record. So, it is impossible to track the attacker	• Logging
Information Di sclosure	V01, V02, V03, V05, V06, V07, V09, V10, V15, V17, V18, V19, V20, V22, V23, V24, V25, V26	▪ An attacker can steal general user credentials such as user name, password, otp seed, token and use this credentials to acquire the plate information ▪ An attacker can steal Administrator`s credentials such as ad min name, password, otp seed, cookie and use this admin credentials to add new illegal account	• Minimize interfaces that can expose information disclosu re • Provide the client interface in the form of WebApp. - Eliminate exposed IPC channels by operating in one proc ess - Block external web debug tools at the WebApp level • Add access permissions to the API provided to the adminis trator • Using Javascript Obfuscation technology
DoS	V04	• An attacker can disable the use of the client app	• Check to input validation on the client side • Using file input filters
Elevation of Pr ivilege	V11, V12, V13, V16,	• An attacker can acquire the plate information • An Attacker can add new account arbitrarily	• Eliminate account information disclosure by using role based access controls for server API. • Minimize attackable URLs such as "team-server-dhzve.run. goorm.io/auth/<user>" • When the number of failed logins is exceeded, the accoun t is locked

4.2.3 V01 – UI Webapp in client side send user credential as raw string, not encrypted

Table 62 Vulnerability - V01

Description	[V01] UI Webapp in client side send user credential as raw string, not encrypted		
CIA	[CONFIDENTIALITY]	Attack vector	[CLIENT/ACCESS]
Approach	[REVIEW/DESIGN]	Exploit technique	[SNIFFING]
Vulnerabilities			
1. In client side, there are three parts; webapp for UI, local nodejs server and ALPR 2. To communicate between webapp UI and local server, HTTP which is not secured channel is used 3. It causes information disclosure, because data is not encrypted  			
CVSS Score	7.9	Severity	High
 			
Consequence / Impact analysis			
The officer's credential is not only for ALPR service. It may be SSO(Single Sign On) account for police office. So, this officer's credential can be used for following attack like penetrate police office server and connect to police office portal and then steal target data and so on. In addition, store this user credential and use this for another site's dictionary attack. Because people sometimes use same ID/PW on various site. Also, attacker can sell user credential to illegal group and make money			
Recommended mitigation			
Use secure communication channel protocol like HTTPS(above TLS version 1.2) between module, instead of HTTP			

Tools needed	Wireshark(Packet capture)

Relative component / source code

Local Server - UI webapp

client/web/src/images/server.js

```
13
14  // const server = https.createServer(options, app);
15  const server = http.createServer(app);
16  const cors = require('cors');
17 const io = require('socket.io')(server,{
18      cors : {
214 server.listen(4000, function () {
215     var host = server.address().address
216     var port = server.address().port
217 })
```

client/web/src/pages/login.js

```
38     onSubmit: (values) => {
39         const formData = new URLSearchParams();
40         formData.append("email", values.email);
41         formData.append("password", values.password);
42         formData.append("otp", values.otp);
43
44         axios.post('http://localhost:4000/login', null, {
45             params: formData
46         }).then((response => {
47             if(response.data.result.status !== "fail"){
48               sessionStorage.setItem("key",response.data.result.token);
49               onCancel();
50             } else {
51                alert(response.data.result.message);
52             }
53         }));
54     }
55   });
```

Proof of concept / How to attack

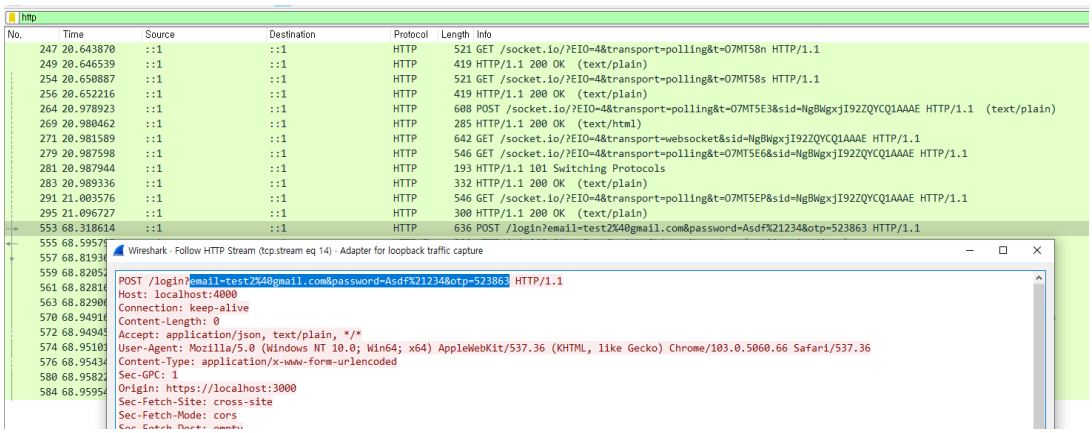
Constraints:

- Attacker can access client device and run Wireshark for packet capturing

Procedure:

1. Run Wireshark for capturing packets and bind to localhost
2. Connect to officer UI webpage(https://localhost:3000)
3. Attempt to login with ID/PW/OTP which are provided

4. Check that user credential is not encrypted(HTTP Protocol) in Wireshark

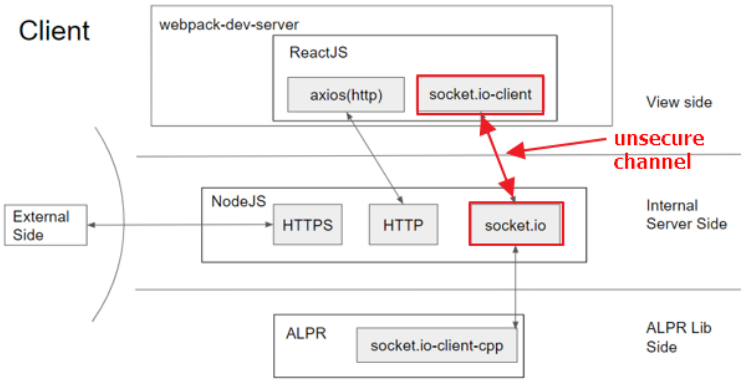


4.2.4 V02 - Privacy is exposed in vehicle registration information

Table 63 Vulnerability - V02

Description	[V02] Privacy is exposed in vehicle registration information		
CIA	[CONFIDENTIALITY]	Attack vector	[CLIENT/ACCESS]
Approach	[REVIEW/DESIGN]	Exploit technique	[SNIFFING]
Vulnerabilities			

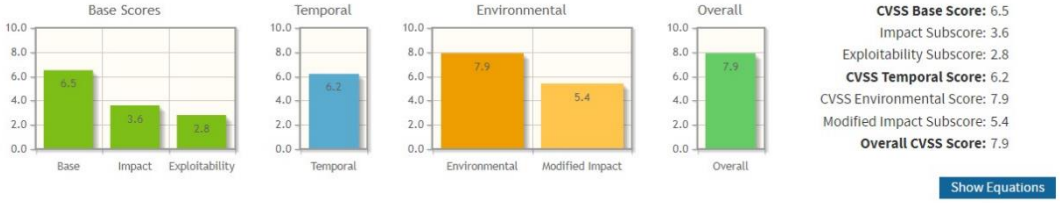
1. In client side, there are three parts; webapp for UI, local nodejs server and ALPR
2. To communicate between webapp UI and local server, websocket which is not secured channel is used
3. It causes information disclosure, because data is not encrypted



CVSS Score	7.9	Severity	High

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



CVSS v3.1 Vector

AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N/E:P/RL:X/RC:C/CR:H/IR:L/AR:L/MAV:N/MAC:L/MPR:L/MUI:N/MS:U/MC:H/MI:N/MA:N

Consequence / Impact analysis	
From vehicle registration, privacy is exposed and stolen by attacker. Attack can know vehicle user's home address and abuse for criminal. Also, attacker can sell privacy to illegal group and make money	
Recommended mitigation	
Use WSS(WebSockets over SSL/TLS) for secure data communication channel instead of websocket	
Tools needed	Wireshark(Packet capture)
Relative component / source code	

Local Server / UI webapp

client/web/src/images/server.js

```
115                 if(plateArr.length === 0 || !plateArr.includes(_data[1])){
116                     plateArr.push(_data[1]);
117                     const request = https.request(host+"/plate/get?plate_number="+_data[1], options, (response) => {
118                         let data = '';
119                         response.on('data', (chunk) => {
120                             data = data + chunk.toString();
121 
122                         });
123 
124                         response.on('end', () => {
125                             try{
126                                 const body = JSON.parse(data);
127                                 io.to(socketId).emit('queryTimeout', '');
128                                 io.to(socketId).emit('plateInfo', body);
129                             } catch(e){
130                                 console.log(e);
131                             }
132                         });
133                     });
```

client/web/src/pages/index.js

```
72     socket.on('plateInfo', (message) => {
73       let result = message;
74       let data = result.result;
75 
76       if(data.status !== "fail"){
77         setPlateInfo(oldArr => [data, ...oldArr]);
78       } else {
79         if(data.message === "Invalid token"){
80           alert(data.message + ". Please signin again.");
81           sessionStorage.clear();
82           setJwt("");
83           window.location.reload();
84         }
85       }
86     });
```

Proof of concept / How to attack

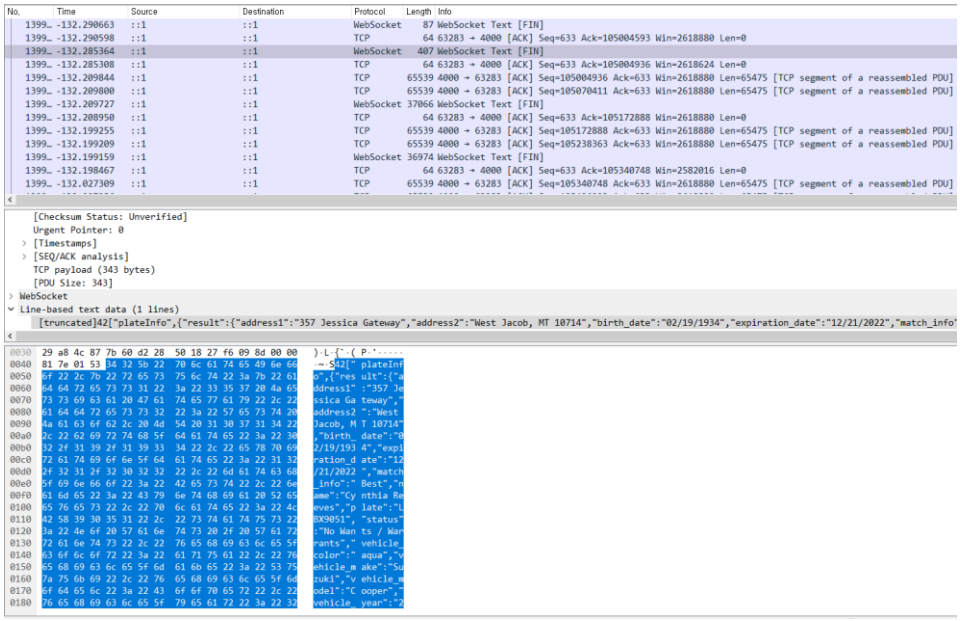
Constraints:

- Attacker can access client device and run Wireshark for packet capturing

Procedure:

1. Run Wireshark for capturing packets and bind to localhost
2. Connect to officer UI webpage(https://localhost:3000)
3. Attempt to login with ID/PW/OTP which are provided
4. Start ALPR service by opening video file "beaver1.avi" and clicking "START ALPR"

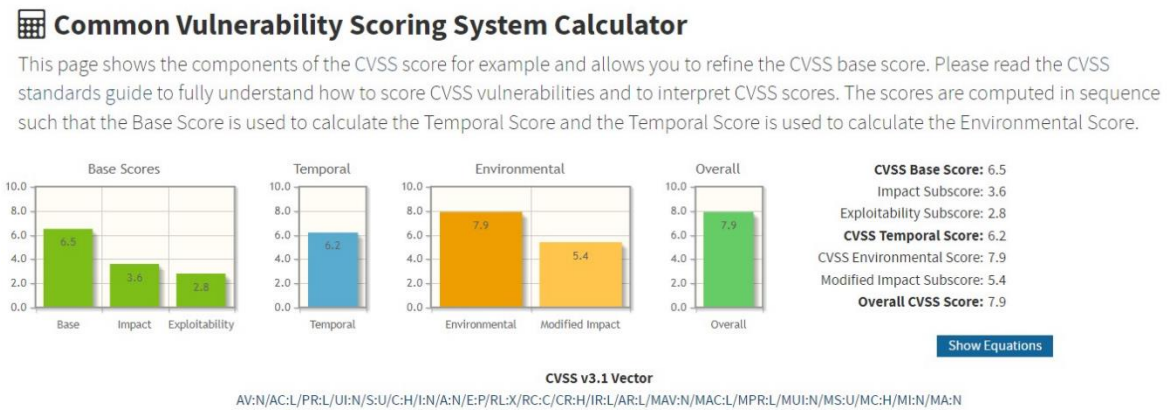
5. Check that vehicle registration info is not encrypted(WebSocket Protocol) in Wireshark



4.2.5 V03 - User credentials are exposed to URL params due to using the 'GET' method.

Table 64 Vulnerability - V03

Description	[V03] User credentials are exposed to URL params due to using the 'GET' method.		
CIA	[CONFIDENTIALITY]	Attack vector	[NETWORK]
Approach	[REVIEW/CODE]	Exploit technique	[SNIFFING]
Vulnerabilities			
Using 'Get' method for login API. "https://team-server- dhzve.run.goorm.io/signin?username=test2@gmail.com&password=Asdf!234&otp=932468"			
Relative component / source code			
CVSS Score	7.9	Severity	High



Consequence / Impact analysis

an attacker can steal user credentials such as user name, password, otp seed. using this information, an attacker can acquire the plate information.	
Recommended mitigation	
Use “POST” method instead of “GET” method in login process.	
Tools needed	Postman
Proof of concept / How to attack	

1. server.js

```
app.post('/login', function(req, res){
    let email = req.query.email;
    let password = req.query.password;
    let otp = req.query.otp;

    const url = host +"/signin?username="+email+"&password="+password+"&otp="+otp;
    const request = https.request(url, (response) => {
        let data = '';
        response.on('data', (chunk) => {
            data = data + chunk.toString();
        });
```

2. server.py

```
user_management = Blueprint('usermanagement', __name__, url_prefix='/')

@user_management.route("/signin", methods=['GET','POST'])
def signin():

  if request.method == 'GET':
    form_user = request.args.get("username")
    form_pass = request.args.get("password")
    form_otp = request.args.get("otp")
  else:
    form_user = request.form.get("username")
    form_pass = request.form.get("password")
    form_otp = request.form.get("otp")
```

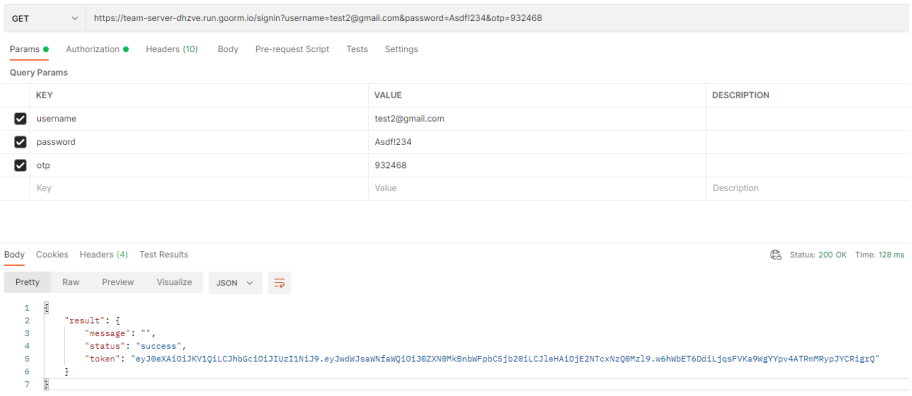
3. Run Postman

4. make Headers & Params

5. send packet to server

https://team-server-dhzve.run.goorm.io/signin?username=test2@gmail.com&password=Asdf!234&otp=932468

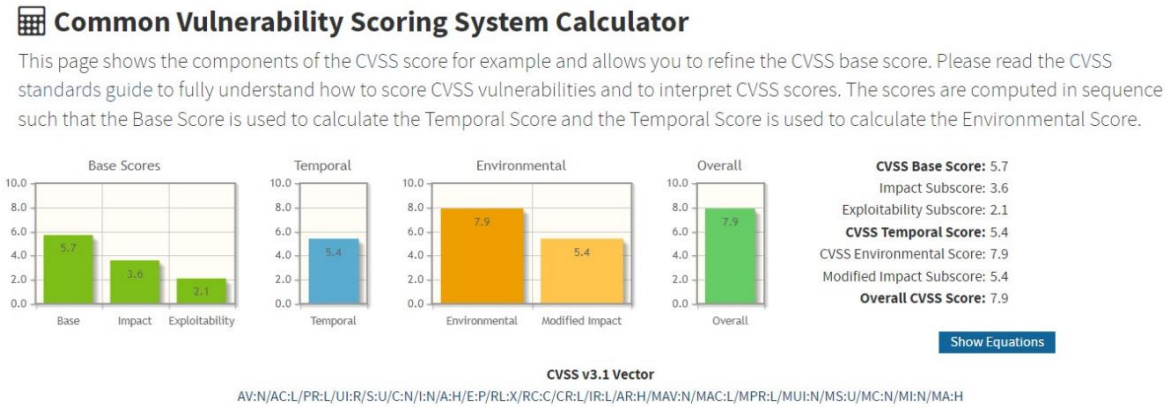
6. result (both http and https protocol possible)



4.2.6 V04 - Attacker can disable playback input operation of client’s Server.js

Table 65 Vulnerability - V04

Description	[V04] Attacker can disable playback input operation of client's Server.js		
CIA	AVAILABILITY	Attack vector	CLIENT/ACCESS
Approach	MANUAL	Exploit technique	NOSPECIFIED
Vulnerabilities			
Abnormal selection of input playback file can compromise the Server.js as client broker. Once Server.js doesn't work, it will be never recovered until restarting client system.			
Relative component / source code			
Local Server			
CVSS Score	7.9	Severity	High



Consequence / Impact analysis
Attacker can disable ALPR system
Recommended mitigation
Input validation can be a choice to prevent selecting abnormal playback file.

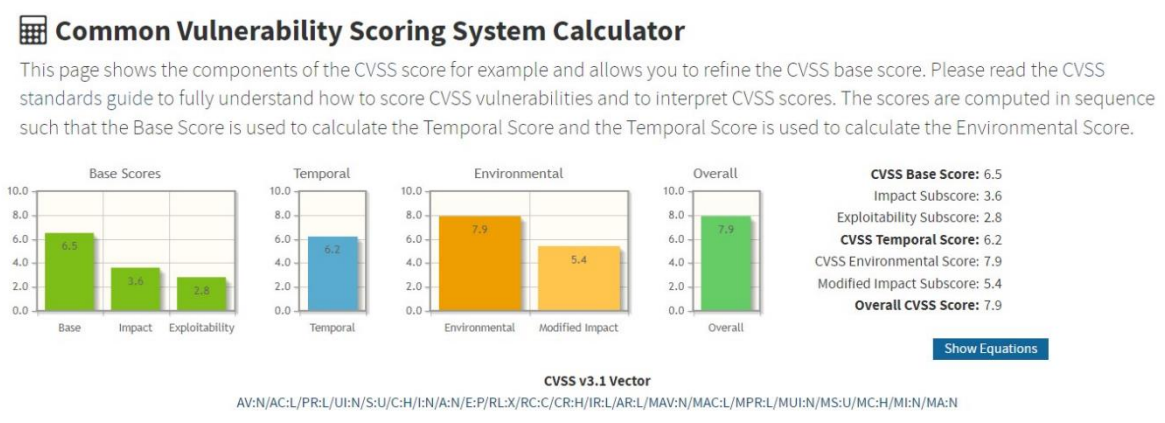
Tools needed	NA

Proof of concept / How to attack
1. Click "파일 선택" and select the normal playback file 2. "START ALPR" click 3. Check displaying playback : normal operation 4. Click "파일 선택" without "STOP ALPR" 5. Select "liblept-DLL.dll" 6. Click "START ALPR" after click "STOP ALPR" 7. Check displaying nothing : normal operation 8. Click "파일 선택" and select the normal playback file 9. Click "START ALPR" 10. Check display nothing : abnormal operation - From now on, Playback doesn't work even though selecting normal playback file. - Server.js seems to be malfunctioning and never recovered until killing and restarting.

4.2.7 V05 - User credentials(username, password, otp) are exposed when using the debug tool such as chrome inspector.

Table 66 Vulnerability - V05

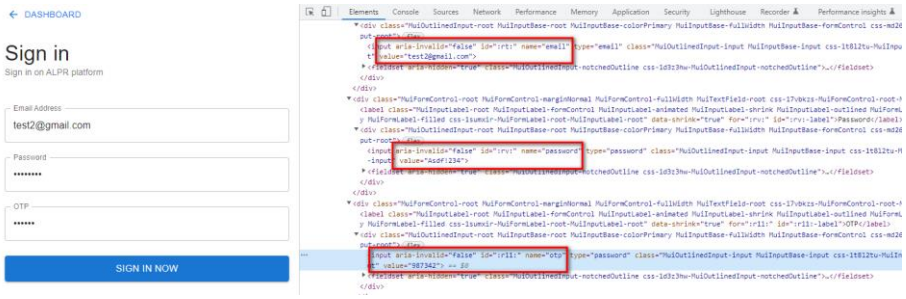
Description	[V05] User credentials(username, password, otp) are exposed when using the debug tool such as chrome inspector.		
CIA	[CONFIDENTIALITY]	Attack vector	[CLIENT/SOURCE]
Approach	[REVIEW/CODE]	Exploit technique	[WEBDEBBER]
Vulnerabilities			
User credentials(username, password, otp) are exposed when using the debug tool such as chrome inspector. 1. login.js source file `function Login(props) { const navigate = useNavigate(); const onCancel = () =>{ navigate(-1); } const formik = useFormik({ initialValues: { email: 'test@gmail.com', password: 'test123', otp: '123456' }, validationSchema: Yup.object({ email: Yup .string() .email('Must be a valid email') .max(80) .required('Email is required'), password: Yup .string() .max(255) .required('Password is required'), otp: Yup .number() .required('OTP is required') }), onSubmit: (values) => { const formData = new URLSearchParams(); formData.append("email", values.email); formData.append("password", values.password); formData.append("otp", values.otp); axios.post('http://localhost:4000/login', null, { params: formData }).then((response => { if(response.data.result.status !== "fail"){ sessionStorage.setItem("key",response.data.result.token); onCancel(); } else { alert(response.data.result.message); } })); } });`			
Relative component / source code			
UI webapp			
CVSS Score	7.9	Severity	High



Consequence / Impact analysis
an attacker can steal user credentials such as user name, password, otp seed. using this information, an attacker can acquire the plate information.
Recommended mitigation
1. Provide the client interface in the form of WebApp because it is possible to block external web debug tools at the WebApp level. 2. Using Javascript Obfuscation technology

Tools needed	Chrome Debug Mode

Proof of concept / How to attack
1. Chrome > More Tool > Developer Tools 2. Select an element in the page to inspect it 3. User credentials(username, password, otp) are exposed



4.2.8 V06 - Token is exposed when using the debug tool such as chrome inspector or client console log

Table 67 Vulnerability - V06

Description	[V06] Token is exposed when using the debug tool such as chrome inspector or client console log		
CIA	[CONFIDENTIALITY]	Attack vector	[CLIENT/SOURCE] [CLIENT/ACCESS]

Approach	[REVIEW/CODE]	Exploit technique	[WEBDEBBER]

Vulnerabilities

Token is exposed when using the debug tool such as chrome inspector or client console log

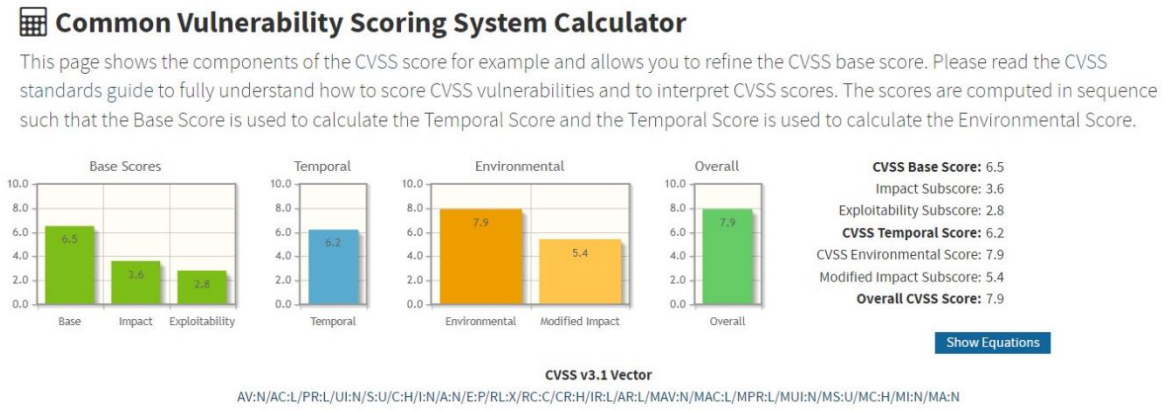
see Usermanagement.py

```
@user_management.route("/signin", methods=['GET','POST'])
def signin():

  if request.method == 'GET':
    form_user = request.args.get("username")
    form_pass = request.args.get("password")
    form_otp = request.args.get("otp")
  else:
    form_user = request.form.get("username")
    form_pass = request.form.get("password")
    form_otp = request.form.get("otp")

  if form_user is None or form_pass is None or form_otp is None:
    return {"result":{
              "status" : "fail",
              "message": "Invalid signin info."
            }}

  result, value = valid_username(form_user)
  if not result:
    return jsonify(value)

  data = User.query.filter_by(username=form_user).first()   # ID 조회Query 실행

  if data is not None:
    if data.password == form_pass:
      totp_instance = pyotp.TOTP(data.otp)
      valid = totp_instance.verify(form_otp)
      if valid:
        data.otpfcnt = 0
        data.passwordfcnt = 0
        db.session.commit()
        token = jwt.encode({'public_id': data.username, 'exp': datetime.datetime.utcnow() + datetime.timedelta(minutes=45)}, config._JWT_SECRET_KEY_, "HS256")
        value = {"result":{
          "status" : "success",
          "token" : token,
          "message": ""
        }}
```

Relative component / source code

UI webapp

CVSS Score	7.9	Severity	High



Consequence / Impact analysis

an attacker can steal user credentials such as jwt token value.

using this information, an attacker can acquire the plate information.

Recommended mitigation

1. Provide the client interface in the form of WebApp because it is possible to block external web debug tools at the WebApp level.

2. Using Javascript Obfuscation technology

Tools needed	Chrome Debug Mode

Proof of concept / How to attack

1. Chrome > More Tool > Developer Tools

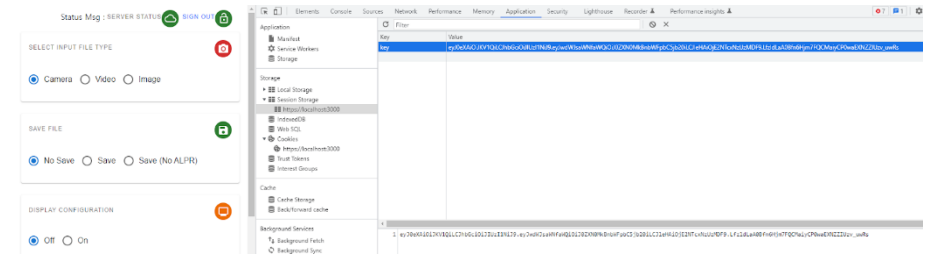
2. Select an Application Tab

3. Storage > Session Storage

4. find key

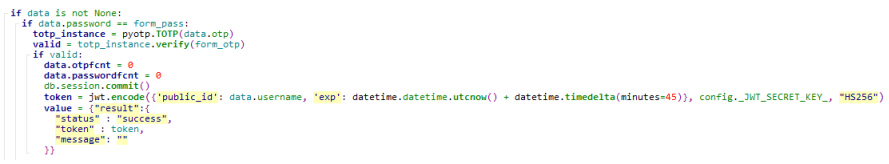
5. or check to client console

6. result

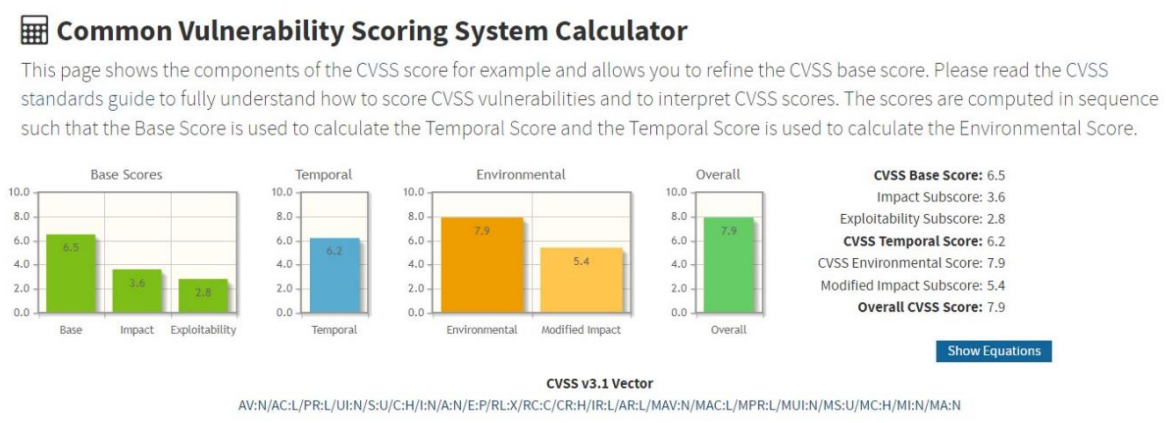


4.2.9 V07 - Username is exposed in the decoded token information.

Table 68 Vulnerability - V07

Description	[V07] Username is exposed in the decoded token information.		
CIA	[CONFIDENTIALITY]	Attack vector	[CLIENT/ACCESS]
Approach	[REVIEW/CODE]	Exploit technique	[SNIFFING]
Vulnerabilities			
Username is exposed in the decoded token information. See Usermanagement.py 			
Relative component / source code			
UI webapp			
CVSS Score	7.9	Severity	High

```
if data is not None:
  if data.password == form_pass:
    totp_instance = pyotp.TOTP(data.otp)
    valid = totp_instance.verify(form_otp)
    if valid:
      data.otpfcnt = 0
      data.passwordfcnt = 0
      db.session.commit()
      token = jwt.encode({'public_id': data.username, 'exp': datetime.datetime.utcnow() + datetime.timedelta(minutes=45)}, config._JWT_SECRET_KEY_, "HS256")
      value = {"result":{
        "status" : "success",
        "token" : token,
        "message": ""
      }}
```



Consequence / Impact analysis
an attacker can steal user credentials such as user name. using this information, an attacker can try to access to login server.
Recommended mitigation
Avoid exposing user accounts to jwt token payload. Instead, use a user id number that can replace public_id.

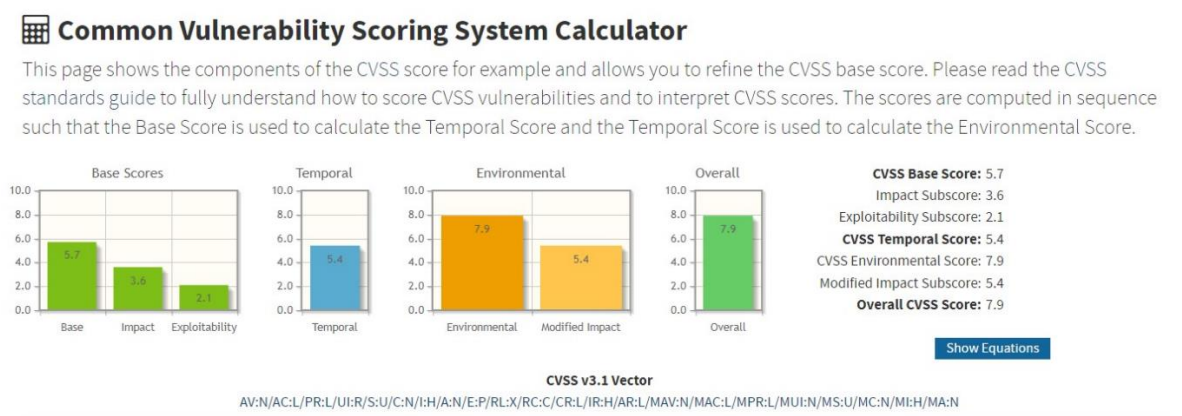
Tools needed	JWT Decoder online

Proof of concept / How to attack
1. get token using V06 Vulnerabilities 2. decode token using Base64

4.2.10 V08 - Attacker can inject a fake play-back to officer's browser

Table 69 Vulnerability - V08

Description	[V08] Attacker can inject a fake play-back to officer's browser		
CIA	INTEGRITY	Attack vector	CLIENT/ACCESS
Approach	MANUAL	Exploit technique	SPOOFING
Vulnerabilities			
Attacker can inject malicious playback file instead of normal playback file.			
Relative component / source code			
Local Server, ALPR.exe, UI webapp			
CVSS Score	7.9	Severity	High



Consequence / Impact analysis
An attacker can use the corrupted file to deceive the officer.
Recommended mitigation
Change design of the interface between local server and alpr.exe (ALPR.exe instance should be corresponded to UI webapp with 1:1 manner.)
Tools needed: NA
Proof of concept / How to attack

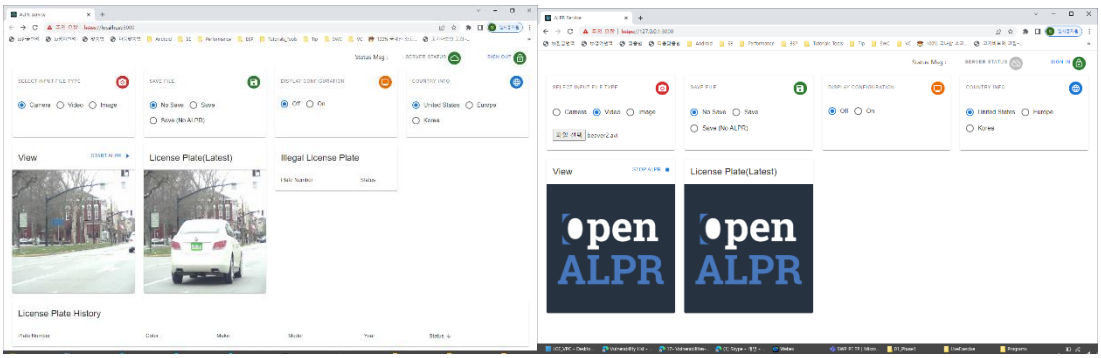
Pre-Condition

1. Officer did log-in normally.

Procedure

1. Attacker executes another browser.
2. Attacker accesses local client's server : "https://127.0.0.1:3000"
3. Attacker selects playback file with skipping log-in.
4. Attacker clicks "START ALPR."
5. Playback file selected by attacker is played on officer's browser and corresponding vehicle information are displayed also.

- Officer's browser - Attacker's browser

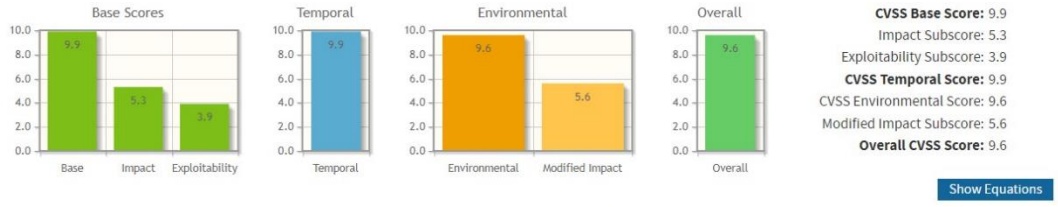


4.2.11 V09 - All Username were exposed when entering a specific page

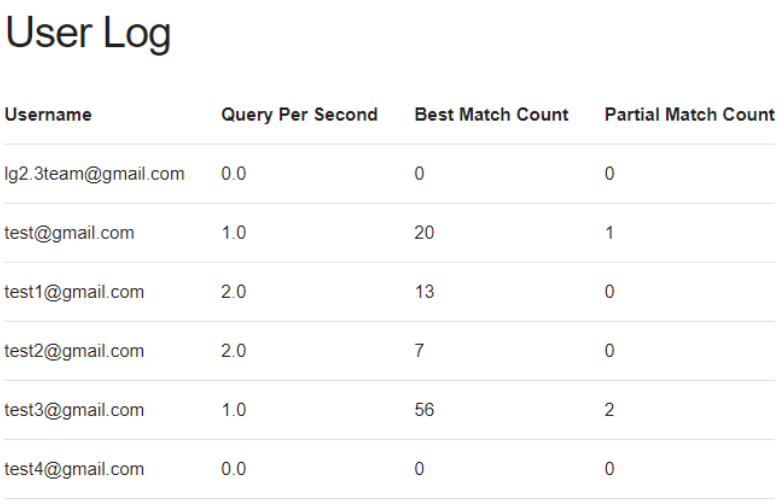
Table 70 Vulnerability - V09

Description	[V09] All Username were exposed when entering a specific page		
CIA	[CONFIDENTIALITY]	Attack vector	[SERVER/ACCESS]
Approach	[REVIEW/DESIGN] [REVIEW/CODE]	Exploit technique	[NOSPECIFIED]
Vulnerabilities			

All Username were exposed when entering a specific page.
- no restrictions on API access.
- GET team-server-dhzve.run.goorm.io/admin/log
see usermanagement.py : no api access restrictions.
`@user_management.route('/admin/log')`
`def adminlog():`
`  logs = Log.query.all()`
`  #sum_best = db.session.query(db.func.sum(Log.bestcnt)).first()[0]`
`  #sum_partial = db.session.query(db.func.sum(Log.partialcnt)).first()[0]`
`  #print(sum_best)`
`  #print(sum_partial)`
`  with  open("log.txt", "w+") as file:`
`  for log in logs:`
`  line = f"{log.username} | {log.qps} | {log.bestcnt} | {log.partialcnt} \n"`
`  file.write(line)`
`  file.close()`
`  return render_template('log.html', title='User Log', logs=logs)`			

Relative component / source code			
CVSS Score	9.6	Severity	Critical
Common Vulnerability Scoring System Calculator This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.  **CVSS v3.1 Vector** AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:L/E:H/RL:X/RC:C/CR:H/IR:L/AR:M/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:N/MA:L			
Consequence / Impact analysis			
An attacker can steal user credentials such as user name in e-mail format. using this information, an attacker can try to access to login server.			
Recommended mitigation			
1. Add access permissions to the API provided to the administrator. 2. Minimize attackable URLs such as "team-server-dhzve.run.goorm.io/admin/log”			
Tools needed	NA		
Proof of concept / How to attack			

1. Enter https://team-server-dhzve.run.goorm.io/admin/log



User Log

Username	Query Per Second	Best Match Count	Partial Match Count
lg2.3team@gmail.com	0.0	0	0
test@gmail.com	1.0	20	1
test1@gmail.com	2.0	13	0
test2@gmail.com	2.0	7	0
test3@gmail.com	1.0	56	2
test4@gmail.com	0.0	0	0

4.2.12 V10 - All OTP seeds were exposed when entering a specific page.

Table 71 Vulnerability - V10

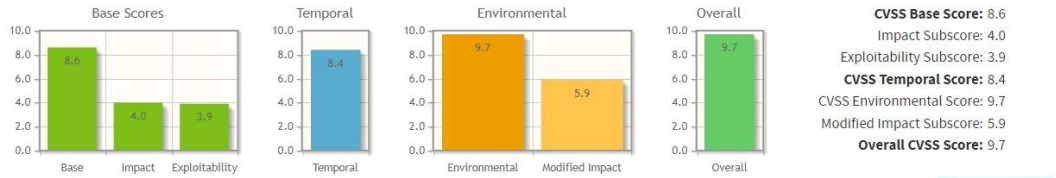
Description	[V10] All OTP seeds were exposed when entering a specific page. As a result, 2 factor authentication becomes useless.		
CIA	[CONFIDENTIALITY]	Attack vector	[SERVER/ACCESS]
Approach	[REVIEW/DESIGN] [REVIEW/CODE]	Exploit technique	[NOSPECIFIED]
Vulnerabilities			

All OTP seeds were exposed when entering a specific page.
As a result, 2 factor authentication becomes useless.
- no restrictions on API access.
- GET team-server-dhzve.run.goorm.io/auth/<user>
see Usermanagement.py : no api access restrictions.
`@user_management.route("/auth/<user>", methods=['GET'])`
`def OTP_auth(user):`
`  data = User.query.filter_by(username=user).first()`
`  if data is not None:`
`  return render_template('auth.html', title='User OTP Info.', secret_key=data.otp,`
`  prov_uri=pyotp.TOTP(data.otp).provisioning_uri(name=user, issuer_name='3team Studio Project'))`
`  else:`
`  flash("Invalid user. Please try again.")`
`  return redirect(url_for("usermanagement.index"))`			

Relative component / source code			
CVSS Score	9.7	Severity	Critical

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



CVSS Base Score: 8.6
Impact Subscore: 4.0
Exploitability Subscore: 3.9
CVSS Temporal Score: 8.4
CVSS Environmental Score: 9.7
Modified Impact Subscore: 5.9
Overall CVSS Score: 9.7

Show Equations

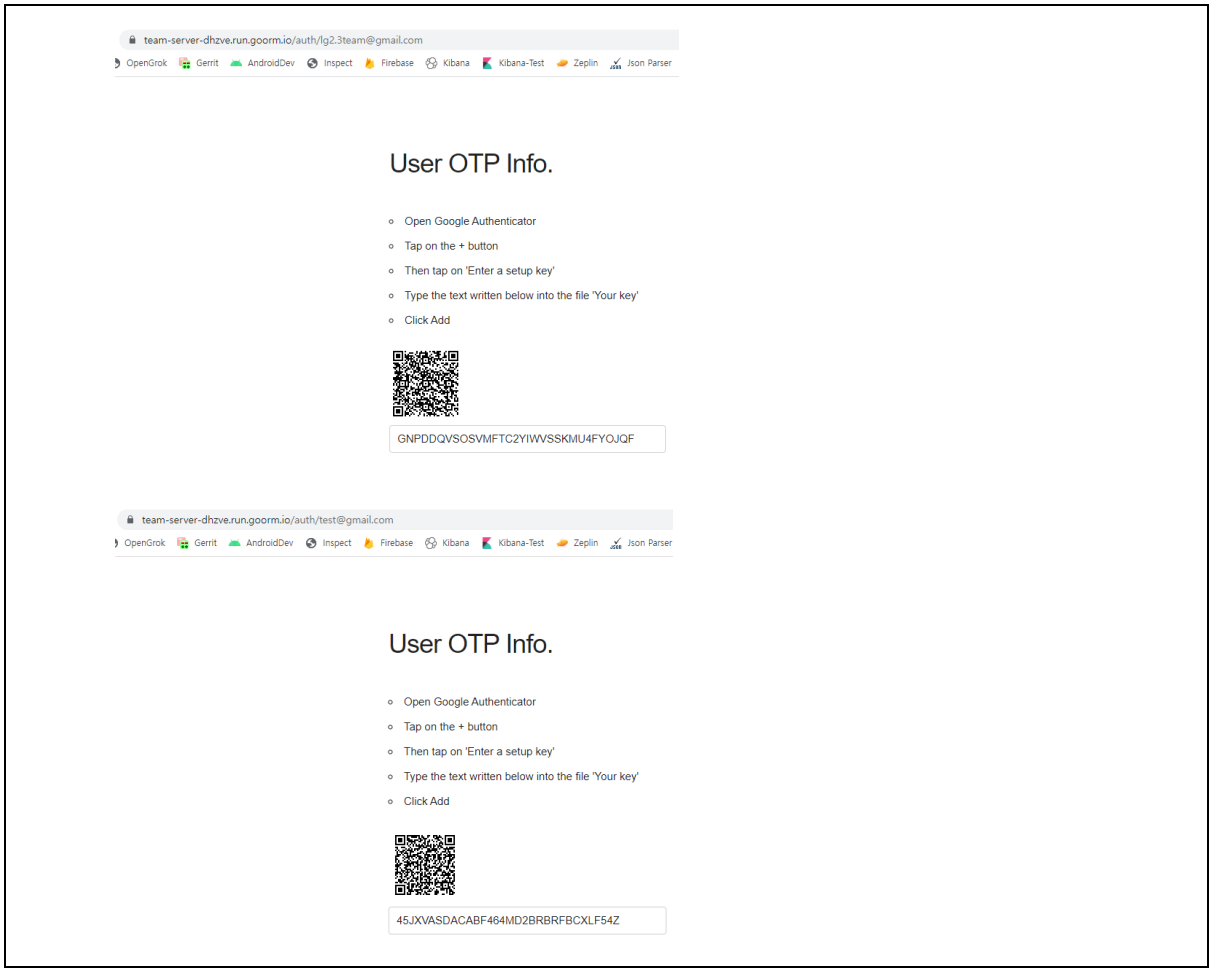
CVSS v3.1 Vector

AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N/E:F/RL:X/RC:C/CR:H/IR:L/AR:L/MAV:N/MAC:L/MPR:N/MUI:N/MS:C/MC:H/MI:N/MA:N

Consequence / Impact analysis
An attacker can steal user credentials such as user otp seed. using this information, an attacker can try to access to login server.
Recommended mitigation
1. Add access permissions to the API provided to the administrator. 2. Minimize attackable URLs such as "team-server-dhzve.run.goorm.io/auth/<user>"

Tools needed	NA

Proof of concept / How to attack
1. Get username from “https://team-server-dhzve.run.goorm.io/admin/log” page without authentication. 2. Enter “https://team-server-dhzve.run.goorm.io/auth/lg2.3team@gmail.com” “https://team-server-dhzve.run.goorm.io/auth/test@gmail.com” “https://team-server-dhzve.run.goorm.io/auth/test1@gmail.com” “https://team-server-dhzve.run.goorm.io/auth/test2@gmail.com” “https://team-server-dhzve.run.goorm.io/auth/test3@gmail.com” “https://team-server-dhzve.run.goorm.io/auth/test4@gmail.com”



4.2.13 V11 - The administrator account has been hijacked and can login with the admin`s credentials(username, password, otp).

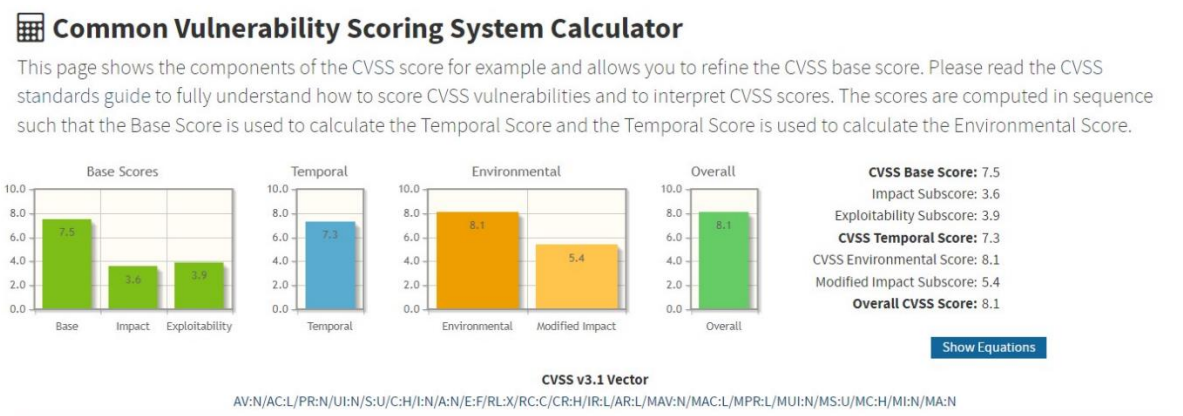
Table 72 Vulnerability - V11

Description	[V11] The administrator account has been hijacked and can login with the admin`s credentials(username, password, otp).		
CIA	[CONFIDENTIALITY]	Attack vector	[SERVER/ACCESS]
Approach	[REVIEW/DESIGN] [REVIEW/CODE]	Exploit technique	[CRAFTPACKET] [SPOOFING]
Vulnerabilities			
The administrator account has been hijacked and can login with the admin`s credentials(username, password, otp). - “team-server-dhzve.run.goorm.io/admin/signin” see Usermamagement.py : use POST method			

```python
@user_management.route("/admin/signin", methods=['GET', 'POST'])
def adminsignin():
  form_user = request.form.get("username")
  form_pass = request.form.get("password")
  form_otp = request.form.get("otp")

  if request.method == 'POST':
    data = User.query.filter_by(username=form_user).first()

    if data is not None:
      if data.password == form_pass :
        totp_instance = pyotp.TOTP(data.otp)
        valid = totp_instance.verify(form_otp)
        if valid:
          if data.admin :
            session['username'] = "admin"
            data.otpfcnt = 0
            data.passwordfcnt = 0
            db.session.commit()
            return redirect(url_for("usermanagement.page"))
          else :
            flash("You don't have admin authorizetion")
```

Relative component / source code

CVSS Score	8.1	Severity	High

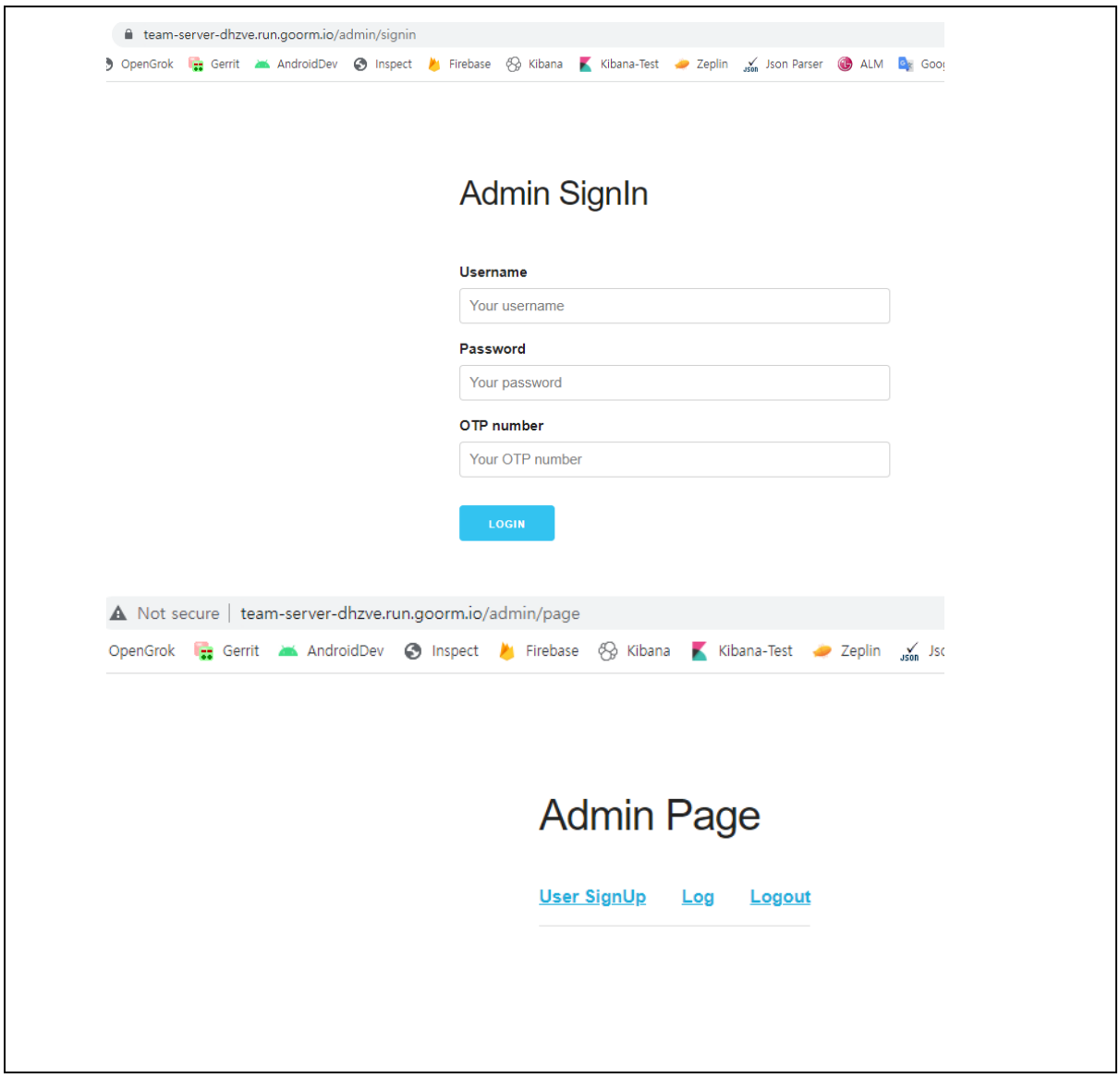


Consequence / Impact analysis
1. an attacker accesses the server with the stolen admin`s account information and gets a admin`s cookie from server. 2. Using this information, an attacker can add new illegal account.

Recommended mitigation
1. Add access permissions to the API provided to the administrator. 2. Minimize attackable URLs such as "team-server-dhzve.run.goorm.io/auth/<user>” 3. Block repeated failed login attempts for a certain period of time 4. When the number of failed logins is exceeded, the account is locked.

Tools needed	Brutus Password Cracker

Proof of concept / How to attack
1. to get admin`s username and password from login`s URL params or chrome inspector or sniff the client. 2.toget otp seeds form “https://team-server-dhzve.run.goorm.io/auth/lg2.3team@gmail.com” 3. enter “https://team-server-dhzve.run.goorm.io/admin/signin” - otp seed : “GNPDDQVSOSVMFTC2YIWVSSKMU4FYOJQF” 4. input Username : “lg2.3team@gmail.com” and Password: “qwer!234”, OTP number : “xxxxxx”



4.2.14 V12 - Any user account can be added by obtaining administrator privileges

Table 73 Vulnerability - V12

Description	[V12] Any user account can be added by obtaining administrator privileges (admin login or stolen cookie)		
CIA	[INTEGRITY]	Attack vector	[SERVER/ACCESS]
Approach	[REVIEW/DESIGN] [REVIEW/CODE]	Exploit technique	[CRAFTPACKET] [SPOOFING]
Vulnerabilities			
Any user account can be added by obtaining administrator privileges. (admin login or stolen cookie) Usermanagement.py			

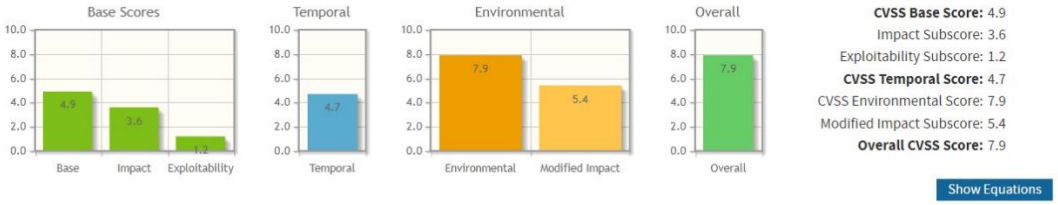
```python
@user_management.route("/admin/signup", methods=['GET', 'POST'])
def siginup():
  if not "username" in session:
    flash("You are not authorized to signup, Please sign in admin account")
    return render_template('index.html', title='Admin SignIn')

  if session['username'] != "admin":
    flash("You are not authorized to signup, Please sign in admin account")
    return render_template('index.html', title='Admin SignIn')

  form_user = request.form.get("username")
  form_pass = request.form.get("password")
  form_confirm_pass = request.form.get("confirm_password")

  if request.method == 'POST':
    if form_user is None or form_pass is None or form_confirm_pass is None:
      return {"result":{
              "status" : "fail",
              "message": "Invalid signup info."
             }}

    result, value = valid_username(form_user)
```

Relative component / source code

CVSS Score	7.9	Severity	High

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



CVSS Base Score: 4.9
Impact Subscore: 3.6
Exploitability Subscore: 1.2
CVSS Temporal Score: 4.7
CVSS Environmental Score: 7.9
Modified Impact Subscore: 5.4
Overall CVSS Score: 7.9

Show Equations

CVSS v3.1 Vector

AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:H/A:N/E:P/RL:X/RC:C/CR:L/IR:H/AR:L/MAV:N/MAC:L/MPR:L/MUI:N/MS:U/MC:N/MI:H/MA:N

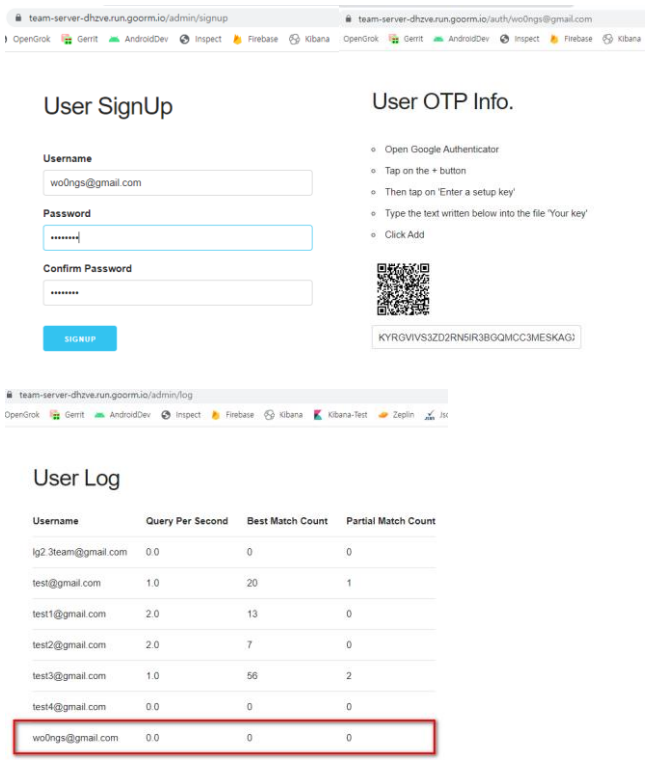
Consequence / Impact analysis
An attacker can pollute the user db by adding illegal users. And An attacker can acquire the plate information using illegal users information

Recommended mitigation
Use trusted device and user authentication using the PKI with SSL/TLS

Tools needed	Postman

Proof of concept / How to attack
1. Log in through the page below “https://team-server-dhzve.run.goorm.io/admin/signin” - Administrator credentials can be acquired through ID12. 2. enter “https://team-server-dhzve.run.goorm.io/admin/signup”

3. add user



4.2.15 V13 - Direct access to "https://team-server-dhzve.run.goorm.io/admin/signup" page is possible through the exposed admin cookie value.

Table 74 Vulnerability - V13

Description	[V13] Direct access to "https://team-server-dhzve.run.goorm.io/admin/signup" page is possible through the exposed admin cookie value. and any account can be added possibly		
CIA	[CONFIDENTIALITY] [INTEGRITY]	Attack vector	[SERVER/ACCESS]
Approach	[REVIEW/DESIGN] [REVIEW/CODE]	Exploit technique	[CRAFTPACKET] [WEBDEBBER]
Vulnerabilities			

usermanagement.py

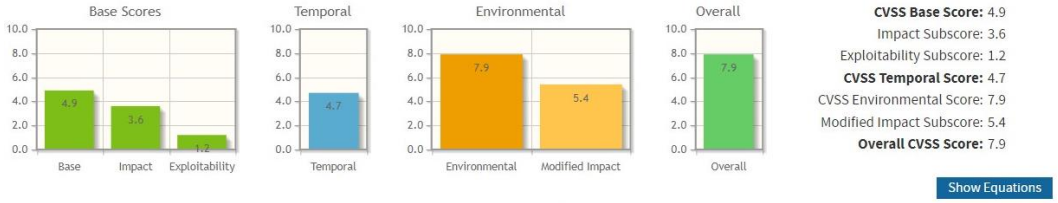
```
@user_management.route("/admin/signup", methods=['GET', 'POST'])
def signinup():
  if not "username" in session:
    flash("You are not authorized to signup, Please sign in admin account")
    return render_template('index.html', title='Admin SignIn')

  if session['username'] != "admin":
    flash("You are not authorized to signup, Please sign in admin account")
    return render_template('index.html', title='Admin SignIn')
  form_user = request.form.get("username")
  form_pass = request.form.get("password")
  form_confirm_pass = request.form.get("confirm_password")

  if request.method == 'POST':
    if form_user is None or form_pass is None or form_confirm_pass is None:
      return {"result":{
                "status" : "fail",
                "message": "Invalid signup info."
              }}

    result, value = valid_username(form_user)
    if not result:
      flash(value)
      return render_template('signup.html', title='User Signup')

    result, value = valid_password(form_pass)
    if not result:
      flash(value)
      return render_template('signup.html', title='User Signup')

    data = User.query.filter_by(username=form_user).first()

    if data is not None:
      flash("User Name already exists. Please try with another one")
      return render_template('signup.html', title='User Signup')

    else:
      if form_pass == form_confirm_pass:
        otptoken=pyotp.random_base32()
        user = User(username=form_user, password=form_pass, otp=otptoken, admin=False)
        log = Log(username=form_user)
        db.session.add(user)
        db.session.add(log)
        db.session.commit()

        return render_template('auth.html', title="User OTP Info.", secret_key=otptoken,
                               prov_uri=pyotp.TOTP(otptoken).provisioning_uri(name=form_user, issuer_name='3team Studio Project'))
```

Relative component / source code

CVSS Score	7.9	Severity	High

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



CVSS Base Score: 4.9
Impact Subscore: 3.6
Exploitability Subscore: 1.2
CVSS Temporal Score: 4.7
CVSS Environmental Score: 7.9
Modified Impact Subscore: 5.4
Overall CVSS Score: 7.9

Show Equations

CVSS v3.1 Vector

AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:H/A:N/E:P/RL:X/RC:C/CR:L/IR:H/AR:L/MAV:N/MAC:L/MPR:L/MUI:N/MS:U/MC:N/MI:H/MA:N

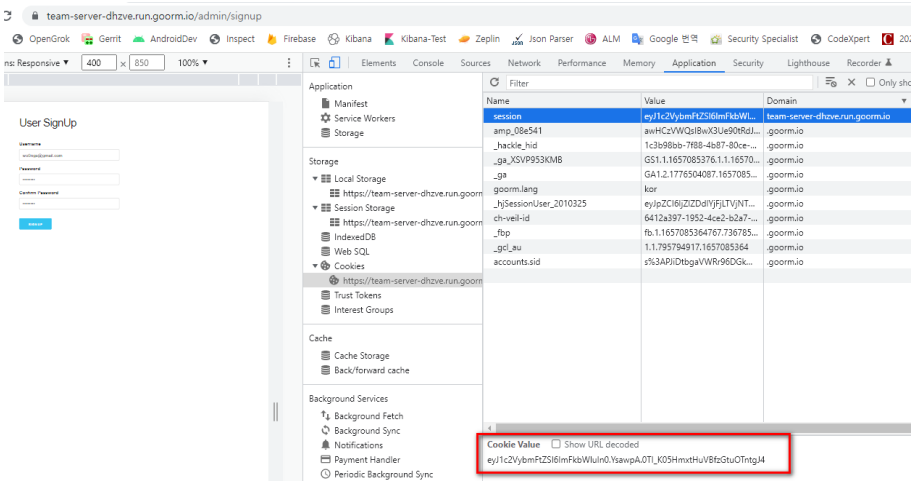
Consequence / Impact analysis
An attacker can steal Administrator`s credentials such as cookie and use this admin credentials to add new illegal account.
Recommended mitigation
1. Use trusted device and user authentication using the PKI with SSL/TLS 2. Using Javascript Obfuscation technology

Tools needed	Postman or Chrome inspector,

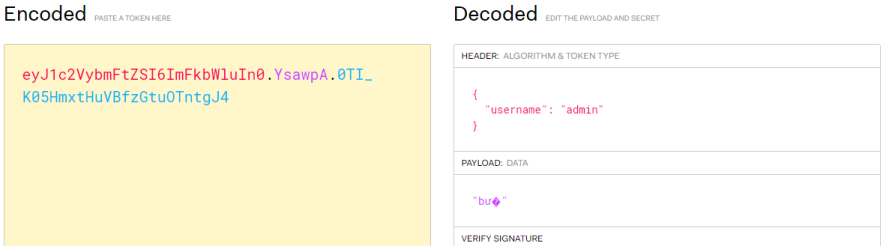
Proof of concept / How to attack

1. exposed admin cookies

- domain : team-server-dhzve.run.goorm.io



2. decoded cookies



3. insert cookie to another pc browser

4.2.16 V14 - no log records for login or logout with an administrator account and looking up log on the server side

Table 75 Vulnerability - V14

Description	[V14] no log records for login or logout with an administrator account and looking up log on the server side		
CIA	[CONFIDENTIALITY]	Attack vector	[SERVER/ACCESS]
Approach	[REVIEW/CODE]	Exploit technique	[NOSPECIFIED]
Vulnerabilities			
usermanagement.py - no log			

```
@user_management.route("/admin/signin", methods=['GET', 'POST'])
def adminsignin():
  form_user = request.form.get("username")
  form_pass = request.form.get("password")
  form_otp = request.form.get("otp")

  if request.method == 'POST':
    data = User.query.filter_by(username=form_user).first()

    if data is not None:
      if data.password == form_pass :
        totp_instance = pyotp.TOTP(data.otp)
        valid = totp_instance.verify(form_otp)
        if valid:
          if data.admin :
            session['username'] = "admin"
            data.otpfcnt = 0
            data.passwordfcnt = 0
            db.session.commit()
            return redirect(url_for("usermanagement.page"))
          else :
            flash("You don't have admin authorizetion")
```

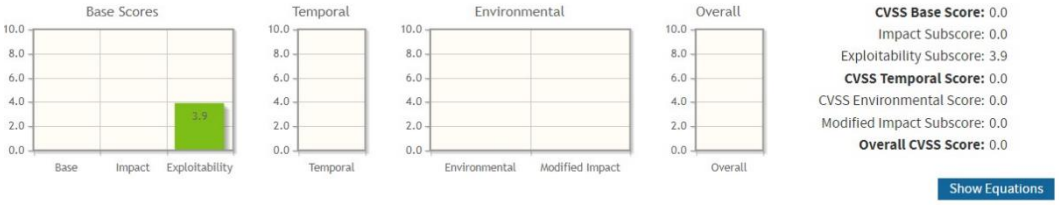
```
@user_management.route("/admin/signin", methods=['GET', 'POST'])
def adminsignin():
  form_user = request.form.get("username")
  form_pass = request.form.get("password")
  form_otp = request.form.get("otp")

  if request.method == 'POST':
    data = User.query.filter_by(username=form_user).first()

    if data is not None:
      if data.password == form_pass :
        totp_instance = pyotp.TOTP(data.otp)
        valid = totp_instance.verify(form_otp)
        if valid:
          if data.admin :
            session['username'] = "admin"
            data.otpfcnt = 0
            data.passwordfcnt = 0
            db.session.commit()
            return redirect(url_for("usermanagement.page"))
          else :
            flash("You don't have admin authorizetion")
```

Relative component / source code

CVSS Score	0.0	Severity	None

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



CVSS v3.1 Vector

AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N/E:P/RL:X/RC:C/CR:L/IR:L/AR:L/MAV:N/MAC:X/MPR:X/MUI:N/MS:U/MC:N/MI:N/MA:N

※ In the CVSS Scoring system, impact is judged only from a CIA perspective.

※ Non-repudiation vulnerabilities have a limit that impact cannot be calculated.

Consequence / Impact analysis
An attacker accesses the server with the stolen account, but the server does not record. So, it is impossible to track the attacker.
Recommended mitigation
Logging

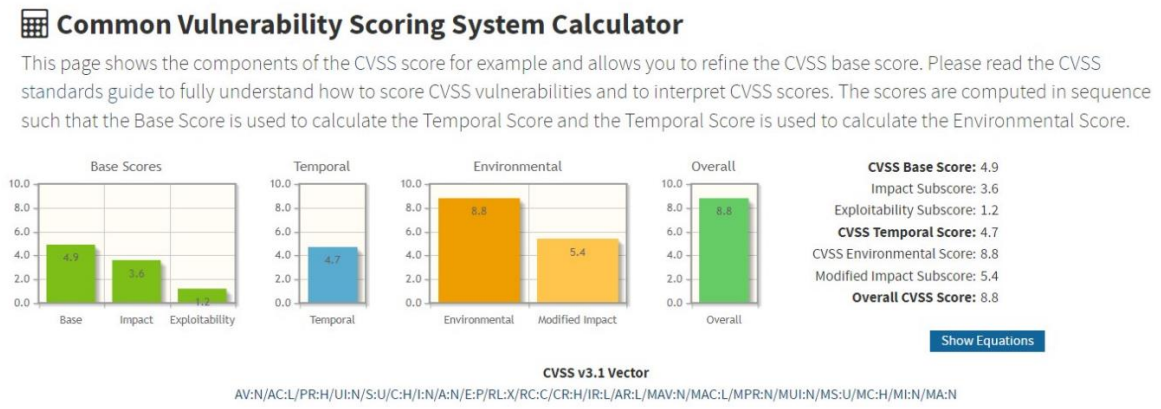
Tools needed	NA
Proof of concept / How to attack	

4.2.17 V15 - An attacker can retrieve plate information with a newly added user account by the stolen administrator account

Table 76 Vulnerability - V15

Description	[V15] An attacker can retrieve plate information with a newly added user account by the stolen administrator account.

CIA	[CONFIDENTIALITY] [INTEGRITY]	Attack vector	[SERVER/ACCESS]
Approach	[REVIEW/CODE]	Exploit technique	[CRAFTPACKET] [SPOOFING]
Vulnerabilities			
Attacker can add new user account.			
Relative component / source code			
CVSS Score	8.8	Severity	High

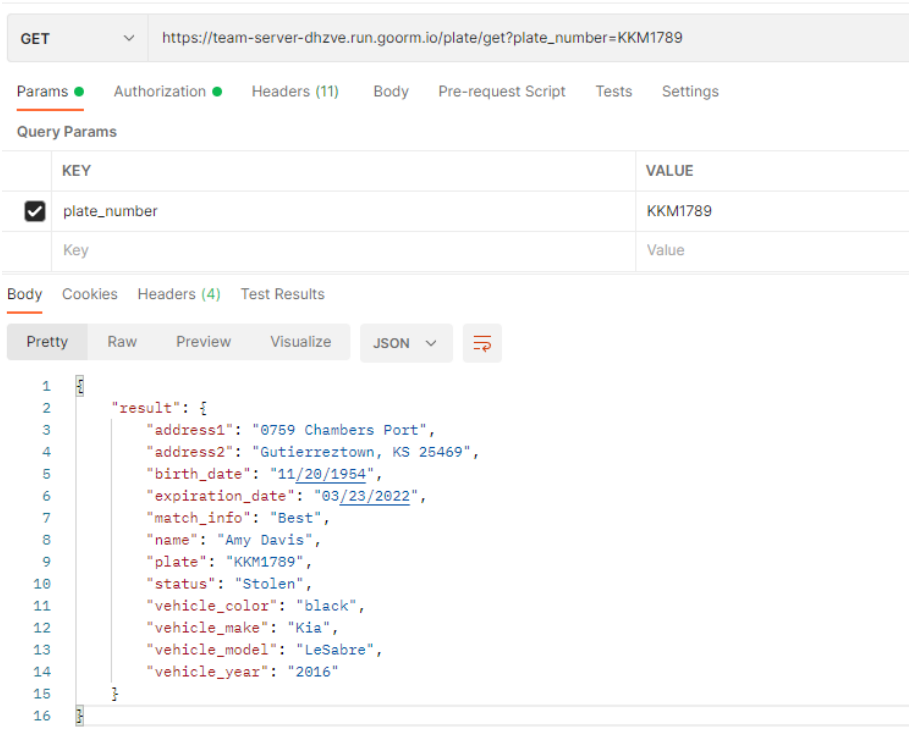


Consequence / Impact analysis
An attacker can acquire the plate information An Attacker can add new account arbitrarily
Recommended mitigation
1. Use trusted device and user authentication using the PKI with SSL/TLS 2. Using Javascript Obfuscation technology 3. Strengthen the security of the OTP issuance process

Tools needed	Postman

Proof of concept / How to attack
1. The attacker adds user accounts according to V12. - wo0ngs@gmail.com is added 2. Login with account “wo0ngs@gmail.com”

3. We can obtain plate and license information by inquiring about any plate number.



4.2.18 V16 - jwt token spoofed by an attacker

Table 77 Vulnerability - V16

Description	[V16] jwt token spoofed by an attacker		
CIA	[CONFIDENTIALITY]	Attack vector	[SERVER/ACCESS]
Approach	[REVIEW/CODE]	Exploit technique	[SPOOFING]
Vulnerabilities			

For the jwt token issued by the server, it is not stored in association with user information.

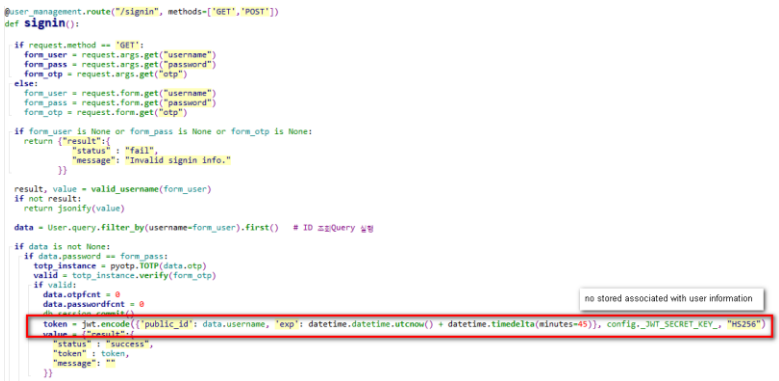
Therefore, multiple logins with the same username are possible.

and there is no logic to handle expired tokens by user logout.

Also if the private key is exposed, it is impossible to validate the jwt token spoofed by an attacker on the server side because the issued jwt token format is simple.

it is defending on the config._JWT_SECRET_KEY_ only.

1. usermanagement.py



.utils.py

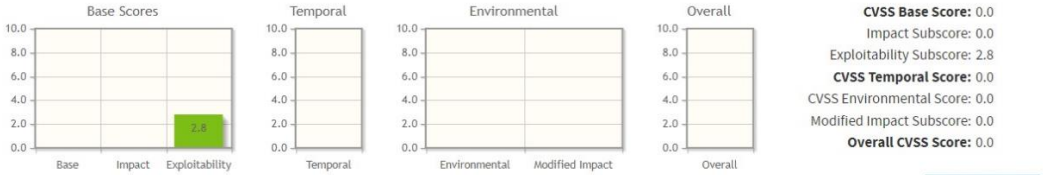
- the token is checked only for tampering without comparing the token stored in the server.

```
def token_required(f):
  @wraps(f)
  def decorator(*args, **kwargs):
    auth = request.headers.get("Authorization")
    if not auth:
      value = {"result":{
        "status" : "fail",
        "message": "Valid token is Missing"
      }}
      return jsonify(value)

    token = None
    if auth.startswith("Bearer "):
      split = auth.split("Bearer")
      token = split[1].strip()
    if not token:
      value = {"result":{
        "status" : "fail",
        "message": "Valid token is Missing"
      }}
      return jsonify(value)

    try:
      data = jwt.decode(token, config._JWT_SECRET_KEY_, algorithms=["HS256"])
      valid_user = User.query.filter_by(username=data['public_id']).first()
    except:
      value = {"result":{
        "status" : "fail",
        "message": "Invalid token"
      }}
      return jsonify(value)

    return f(valid_user, *args, **kwargs)

  return decorator
```

Relative component / source code			
CVSS Score	0.0	Severity	None

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



CVSS v3.1 Vector

AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N/E:P/RL:X/RC:C/CR:L/IR:L/AR:L/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:N/MI:N/MA:N

※ In the CVSS Scoring system, impact is judged only from a CIA perspective.

※ Non-repudiation vulnerabilities have a limit that impact cannot be calculated.

Consequence / Impact analysis
An attacker can acquire the plate information

Recommended mitigation
Using JWT token revoke mechanism No multiple access with one account

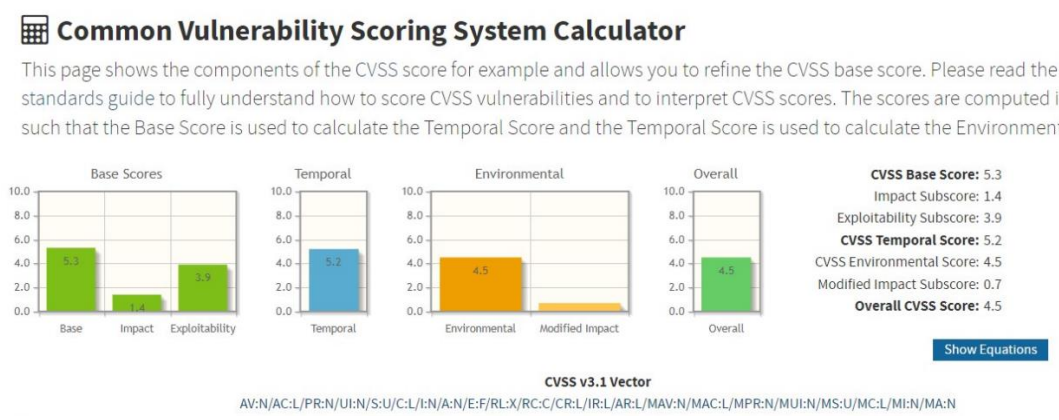
Tools needed	Postman
Proof of concept / How to attack	

4.2.19 V17 – Vulnerability of recovering user password

Table 78 Vulnerability - V17

Description	[V17] Vulnerability of recovering user password		
CIA	[CONFIDENTIALITY]	Attack vector	[SERVER/ACCESS]
Approach	[REVIEW/CODE]	Exploit technique	[BRUTEFORCE]
Vulnerabilities			
If the password and OTP information are wrong 3 times or more, a new password is automatically sent to the account email without authentication. In addition, the new password is the value in which the old passwords are listed in reverse order. So, the original password and the new password appear repeatedly. Therefore, if the user`s account email is exposed, the attacker could easily obtain a password and may be vulnerable to brute force attacks. - usermanagement.js 			
Relative component / source code			
CVSS Score	4.5	Severity	Medium

```
if data is not None:
  if data.password == form_pass:
    totp_instance = pyotp.TOTP(data.otp)
    valid = totp_instance.verify(form_otp)
    if valid:
      data.otpfcnt = 0
      data.passwordfcnt = 0
      db.session.commit()
      token = jwt.encode({'public_id': data.username, 'exp': datetime.datetime.utcnow() + datetime.
      value = {"result":{
        "status" : "success",
        "token" : token,
        "message": ""
      }}

    else:
      data.otpfcnt = data.otpfcnt + 1
      db.session.commit()

      if data.passwordfcnt + data.otpfcnt > 2 :
        password = data.password
        data.password = password[::-1]
        data.passwordfcnt = 0
        data.otpfcnt = 0
        db.session.commit()
        snedmail(data.username, f"new password : {data.password}")
        value = {"result":{
          "status" : "fail",
          "token" : "",
          "message": f"A new password has been sent to you by backup e-mail({data.username})"
        }}
      else:
        value = {"result":{
          "status" : "fail",
          "token" : "",
          "message": "Invalid OTP or Password Number"
        }}
```

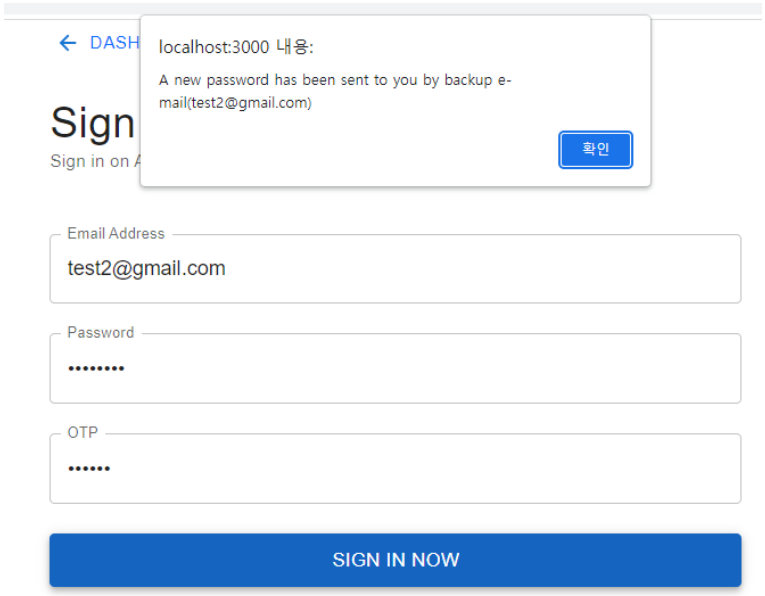


Consequence / Impact analysis
An attacker can try to steal admin or user account using brute force attacks.
Recommended mitigation
When the number of failed logins is exceeded, the account is locked.

Tools needed	brute force attacks

Proof of concept / How to attack

1. password invalid two times.



2. repeat password invalid two times.
3. input same password and valid otp
4. login successful

4.2.20 V18 - An attacker can retrieve plate information with a stolen jwt token.

Table 79 Vulnerability - V18

Description	[V18] An attacker can retrieve plate information with a stolen jwt token.		
CIA	[CONFIDENTIALITY]	Attack vector	[SERVER/ACCESS]

Approach	[REVIEW/CODE]	Exploit technique	[CRAFTPACKET] [SPOOFING]

Vulnerabilities

An attacker can retrieve plate information with a stolen jwt token.

If a jwt token is stolen, it cannot be dealt with until it expires.

1.server.js

```
function getFrameData(query){
    var jwt = query.jwt;
    var executablePath = path.resolve('alpr.exe');
    var parameters = ["-c", query.ct, "-s", query.save, "-f", query.res];

    if(query.type === "camera"){
        parameters.push(query.name);
    } else if(query.type === "video" || query.type === "image"){
        parameters.push(path.resolve(query.name));
    }

    if(pid === 0){
        var cmd = executablePath + " " + parameters.join(' ');
        const child = exec(cmd, (error, stdout, stderr) => {
            if (error) {
            }
        });

        pid = child.pid;
        child.stdout.on('data', (data) => {
            if(data.includes("plate")){
                let _data = data.split(",");
                try{
                    if(_data[1] !== null && _data[2] !== null && _data[2] >= 90){
                        if(jwt !== null){
                            var options = {
                                timeout: 10000,
                                time: true,
                                headers: {
                                    'Content-Type': 'application/json',
                                    'Authorization': `Bearer ${jwt}`
                                }
                            };
```

2. platequery.py

```
@plate_query.route("/get", methods=['GET'])
@token_required
def getPlate(valid_user):
    if valid_user is not None:

      plate_number = request.args.get('plate_number')

      if plate_number is None:
        value = {"result":{"status" : "fail",
                           "message":"Invailed plase number"}}
        return jsonify(value)

      result, value = valid_platenumber(plate_number)
      if not result:
        return jsonify(value)

      data = Plate.query.filter_by(plate=plate_number).first()
```

3. the token is the only access right to the API.

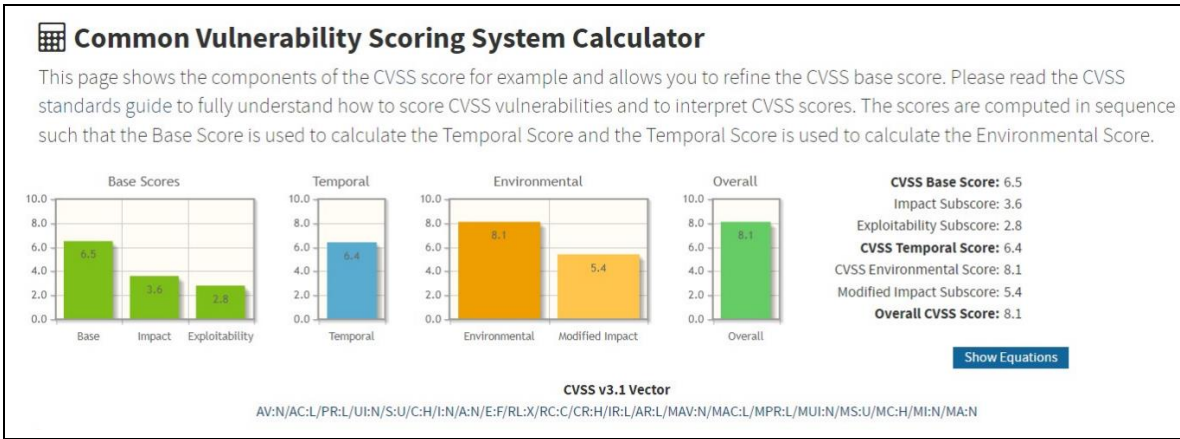
4. After logout of the client app, the token is still valid on the server side.

- index.js : only clear client side`s token

```
const signOut = () => {
  sessionStorage.clear();
  setJwt(sessionStorage.getItem("key"));
}
```

Relative component / source code

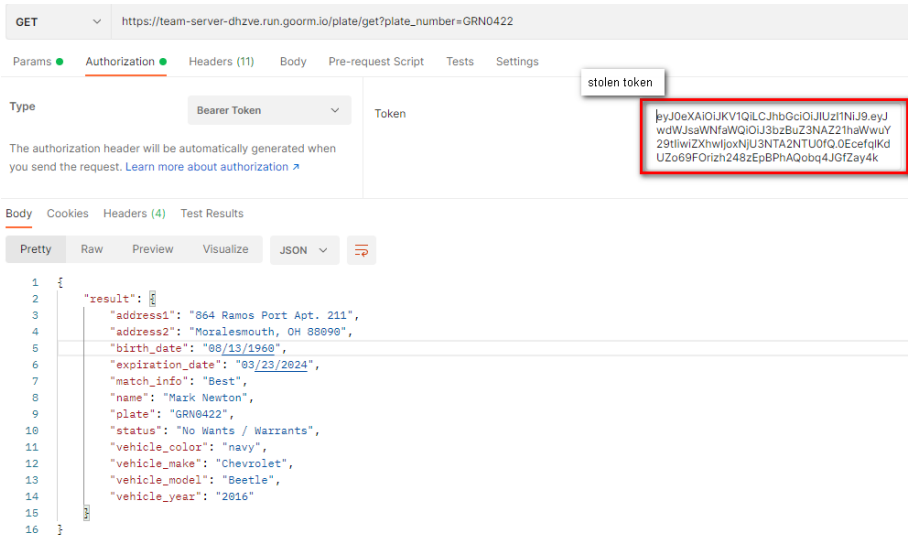
CVSS Score	8.1	Severity	High



Consequence / Impact analysis
An attacker can acquire the plate information
Recommended mitigation
Using the PKI with SSL/TLS Using JWT token revoke mechanism
Tools needed: Postman
Proof of concept / How to attack

1. Run Postman
2. make Headers with token & Params
3. send packet to server

   GET "http://team-server-dhzve.run.goorm.io/plate/get?plate_number=GRN0422"



4.2.21 V19 - There is no JWT token revoke mechanism

Table 80 Vulnerability - V19

Description	[V19] There is no JWT token revoke mechanism		
CIA	[CONFIDENTIALITY]	Attack vector	[SERVER/ACCESS]

Approach	[REVIEW/CODE] [REVIEW/DESIGN]	Exploit technique	[SNIFFING]

Vulnerabilities

The expired time of the jwt token is as short as 45 minutes, but refresh token and logic procedure required to maintain a session do not exist.

therefore the user should repeatedly call the login GET method API, there is a risk of account information exposure.

“https://team-server-dhzve.run.goorm.io/signin?username=test2@gmail.com&password=Asdf!234&otp=932468

1.usermanagement.js

- only jwt token available

```
@user_management.route("/signin", methods=['GET','POST'])
def signin():

  if request.method == 'GET':
    form_user = request.args.get("username")
    form_pass = request.args.get("password")
    form_otp = request.args.get("otp")
  else:
    form_user = request.form.get("username")
    form_pass = request.form.get("password")
    form_otp = request.form.get("otp")

  if form_user is None or form_pass is None or form_otp is None:
    return {"result":{
              "status" : "fail",
              "message": "Invalid signin info."
            }}

  result, value = valid_username(form_user)
  if not result:
    return jsonify(value)

  data = User.query.filter_by(username=form_user).first()   # ID 조회Query 실행

  if data is not None:
    if data.password == form_pass:
      totp_instance = pyotp.TOTP(data.otp)
      valid = totp_instance.verify(form_otp)
      if valid:
        data.otpfcnt = 0
        data.passwordfcnt = 0
        db.session.commit()
        token = jwt.encode({'public_id': data.username, 'exp': datetime.datetime.utcnow() + datetime.timedelta(minutes=45)}, config._JWT_SECRET_KEY_, "HS256")
        value = {"result":{
          "status" : "success",
          "token" : token,
          "message": ""
        }}
```

2. platequery.py

```
@user_management.route("/signin", methods=['GET','POST'])
def signin():

  if request.method == 'GET':
    form_user = request.args.get("username")
    form_pass = request.args.get("password")
    form_otp = request.args.get("otp")
  else:
    form_user = request.form.get("username")
    form_pass = request.form.get("password")
    form_otp = request.form.get("otp")

  if form_user is None or form_pass is None or form_otp is None:
    return {"result":{
              "status" : "fail",
              "message": "Invalid signin info."
            }}

  result, value = valid_username(form_user)
  if not result:
    return jsonify(value)

  data = User.query.filter_by(username=form_user).first()   # ID 조회Query 실행

  if data is not None:
    if data.password == form_pass:
      totp_instance = pyotp.TOTP(data.otp)
      valid = totp_instance.verify(form_otp)
      if valid:
        data.otpfcnt = 0
        data.passwordfcnt = 0
        db.session.commit()
        token = jwt.encode({'public_id': data.username, 'exp': datetime.datetime.utcnow() + datetime.timedelta(minutes=45)}, config._JWT_SECRET_KEY_, "HS256")
        value = {"result":{
          "status" : "success",
          "token" : token,
          "message": ""
        }}
```

3.utils.py - only jwt token valid check.

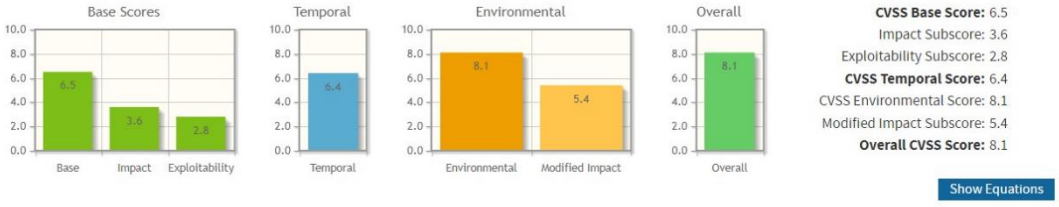
```
def token_required(f):
  @wraps(f)
  def decorator(*args, **kwargs):
    auth = request.headers.get("Authorization")
    if not auth:
      value = {"result":{
        "status" : "fail",
        "message": "Valid token is Missing"
      }}
      return jsonify(value)

    token = None
    if auth.startswith("Bearer "):
      split = auth.split("Bearer")
      token = split[1].strip()
    if not token:
      value = {"result":{
        "status" : "fail",
        "message": "Valid token is Missing"
      }}
      return jsonify(value)

    try:
      data = jwt.decode(token, config._JWT_SECRET_KEY_, algorithms=["HS256"])
      valid_user = User.query.filter_by(username=data['public_id']).first()
    except:
      value = {"result":{
        "status" : "fail",
        "message": "Invalid token"
      }}
      return jsonify(value)

    return f(valid_user, *args, **kwargs)

  return decorator
```

Relative component / source code

CVSS Score	8.1	Severity	High

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



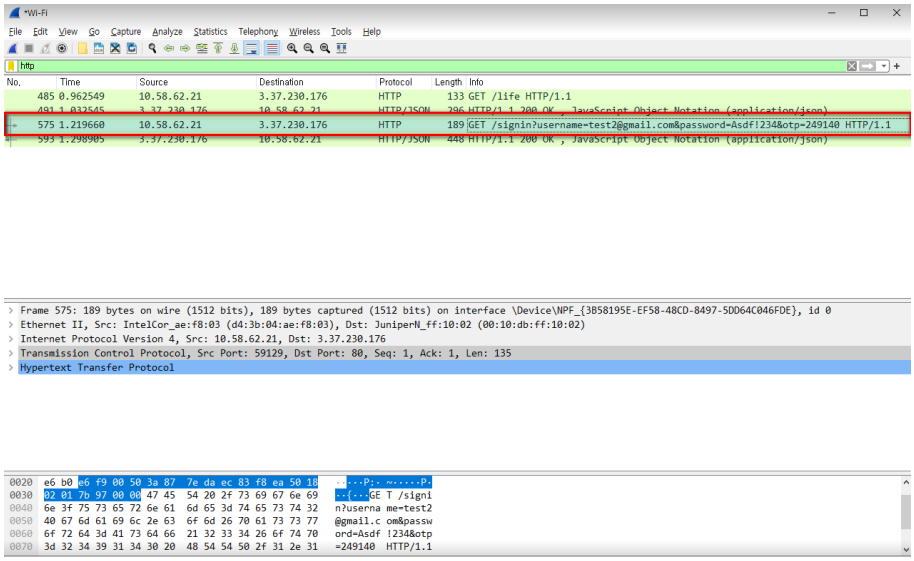
CVSS v3.1 Vector

AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N/E:F/RL:X/RC:C/CR:H/IR:L/AR:L/MAV:N/MAC:L/MPR:L/MUI:N/MS:U/MC:H/MI:N/MA:N

Consequence / Impact analysis
An attacker can steal general user credentials such as user name, password, otp, token and use its credentials to acquire the plate information
Recommended mitigation
1. Using JWT token revoke mechanism 2. Use “POST” method instead of “GET” method in login process.

Tools needed	Wireshark
Proof of concept / How to attack	

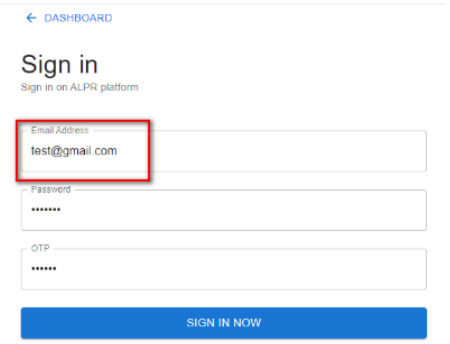
sniffing



4.2.22 V20 - Actual account(test@gmail.com) is entered as an example in the Email Address field.

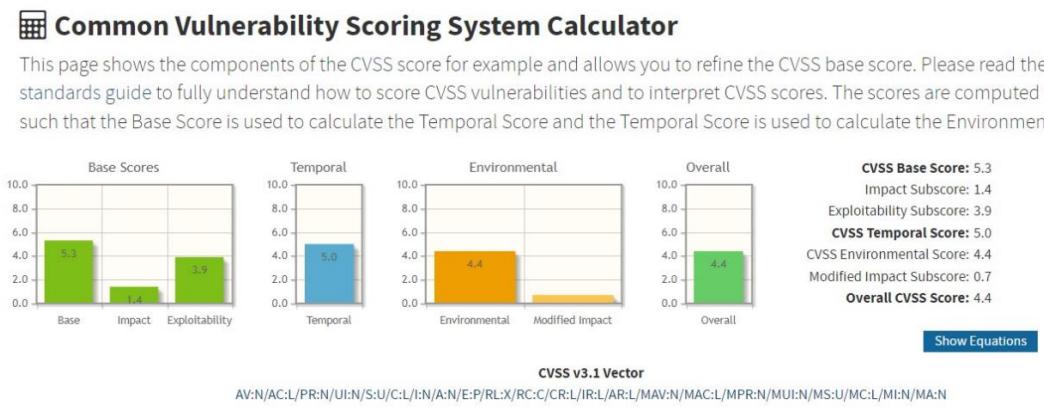
Table 81 Vulnerability - V20

Description	[V20] actual account(test@gmail.com) is entered as an example in the Email Address field.		
CIA	[CONFIDENTIALITY]	Attack vector	[CLIENT/ACCESS]
Approach	[REVIEW/CODE]	Exploit technique	[NOSPECIFIED]
Vulnerabilities			
In the "Sign in" page of the client, an actual account(test@gmail.com) is entered as an example in the Email Address field. There is a risk that the account may be hijacked. 1.login.js  2. client`s Sign in page			



Relative component / source code

CVSS Score	4.4	Severity	Medium



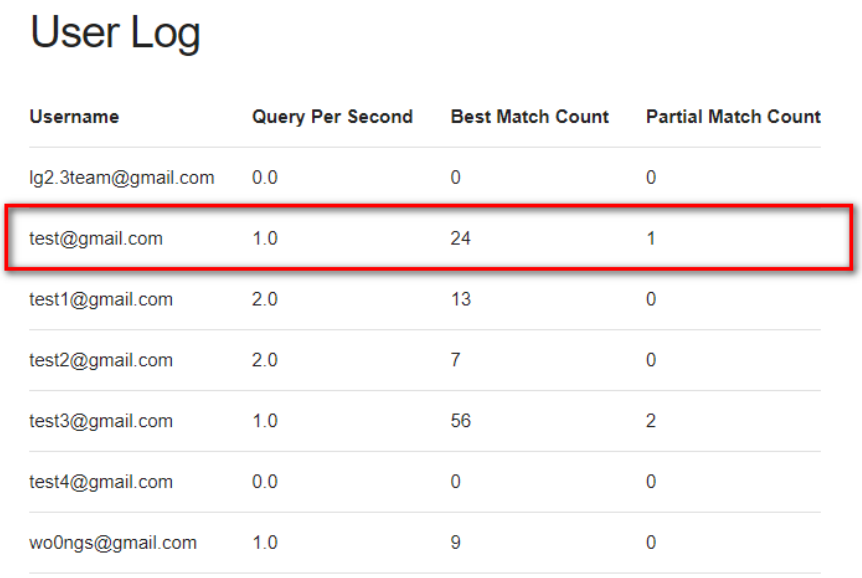
Consequence / Impact analysis
An attacker can try to steal “test@gamil.com” account using brute force attacks.

Recommended mitigation
Don't show real accounts as examples in the user interface.

Tools needed	

Proof of concept / How to attack

we retrieve the account information and to confirm that "test@gmail.com" actually exists.



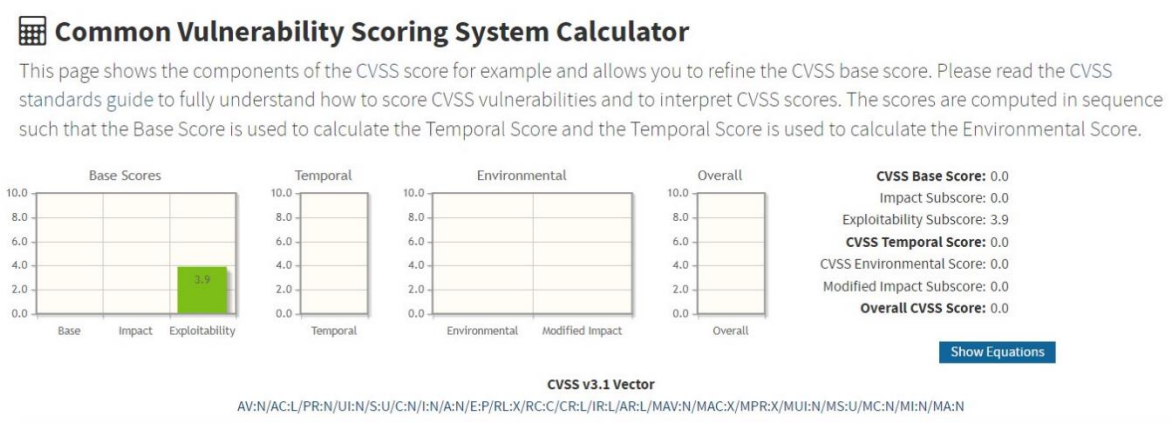
User Log

Username	Query Per Second	Best Match Count	Partial Match Count
lg2.3team@gmail.com	0.0	0	0
test@gmail.com	1.0	24	1
test1@gmail.com	2.0	13	0
test2@gmail.com	2.0	7	0
test3@gmail.com	1.0	56	2
test4@gmail.com	0.0	0	0
wo0ngs@gmail.com	1.0	9	0

4.2.23 V21 - no log records for login with a general account on the server side.

Table 82 Vulnerability - V21

Description	[V21] no log records for login with a general account on the server side.		
CIA	[CONFIDENTIALITY]	Attack vector	[SERVER/ACCESS]
Approach	[REVIEW/CODE]	Exploit technique	[NOSPECIFIED]
Vulnerabilities			
no log records for login with a general account on the server side. only a token is generated and delivered to the client.			
Relative component / source code			
CVSS Score	0.0	Severity	None



Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.

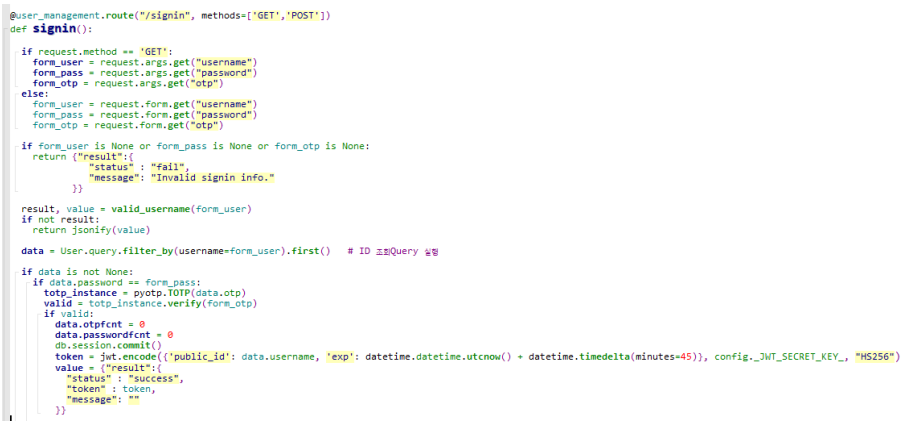
Base Scores | Temporal | Environmental | Overall

CVSS Base Score: 0.0
Impact Subscore: 0.0
Exploitability Subscore: 3.9
CVSS Temporal Score: 0.0
CVSS Environmental Score: 0.0
Modified Impact Subscore: 0.0
Overall CVSS Score: 0.0

Show Equations

CVSS v3.1 Vector

AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N/E:P/RL:X/RC:C/CR:L/IR:L/AR:L/MAV:N/MAC:X/MPR:X/MUI:N/MS:U/MC:N/MI:N/MA:N

※ In the CVSS Scoring system, impact is judged only from a CIA perspective. **※ Non-repudiation vulnerabilities have a limit that impact cannot be calculated.**
Consequence / Impact analysis
An attacker accesses the server with the stolen account, but the server does not record. So, it is impossible to track the attacker.
Recommended mitigation
Logging
Tools needed: NA
Proof of concept / How to attack
usermanagement.py - no log 

```
@user_management.route("/signin", methods=['GET','POST'])
def signin():

  if request.method == 'GET':
    form_user = request.args.get("username")
    form_pass = request.args.get("password")
    form_otp = request.args.get("otp")
  else:
    form_user = request.form.get("username")
    form_pass = request.form.get("password")
    form_otp = request.form.get("otp")

  if form_user is None or form_pass is None or form_otp is None:
    return {"result":{
              "status" : "fail",
              "message": "Invalid signin info."
            }}

  result, value = valid_username(form_user)
  if not result:
    return jsonify(value)

  data = User.query.filter_by(username=form_user).first()   # ID 조회Query 실행

  if data is not None:
    if data.password == form_pass:
      totp_instance = pyotp.TOTP(data.otp)
      valid = totp_instance.verify(form_otp)
      if valid:
        data.otpfcnt = 0
        data.passwordfcnt = 0
        db.session.commit()
        token = jwt.encode({'public_id': data.username, 'exp': datetime.datetime.utcnow() + datetime.timedelta(minutes=45)}, config._JWT_SECRET_KEY_, "HS256")
        value = {"result":{
          "status" : "success",
          "token" : token,
          "message": ""
        }}
```

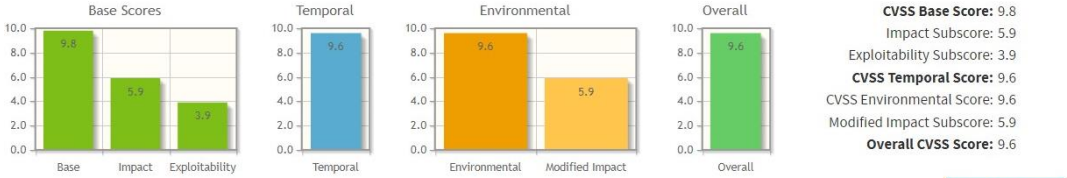
4.2.24 V22 - Remote server spoofing

Table 83 Vulnerability - V22

Description	[V22] Remote server spoofing		
CIA	[Availability] [Confidentiality]	Attack vector	[NETWORK]
Approach	[REVIEW/DESIGN]	Exploit technique	[SPOOFING]
Vulnerabilities			
1. Local server in client side and remote server do not proceed with mutual authentication. 2. The remote server is implemented with http, but Goorm hosting service(COTS) supports https(TLS) when packet is came into 443 port. And convert to http packet because remote server in container is bounded at 80 port(HTTP). It means that client authentication is omitted 3. Local server uses HTTPS/TLS1.2, but dose not authenticate server's certificate, just verify root CA of server certificate if this certificate is issued from authorized agency. It means that there is no authentication of server identification and encrypt packet on external network only. 4. If attacker can redirect remote server URL to attacker's URL/IP, server spoofing could be done without any additional procedure			
CVSS Score	9.6	Severity	Critical

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



CVSS Base Score: 9.8
Impact Subscore: 5.9
Exploitability Subscore: 3.9
CVSS Temporal Score: 9.6
CVSS Environmental Score: 9.6
Modified Impact Subscore: 5.9
Overall CVSS Score: 9.6

Show Equations

CVSS v3.1 Vector

AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RL:X/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:H/MA:H

Consequence / Impact analysis
Spoofing server can occur stealing sensitive data (e.g. user credential) and it is sold to illegal group for making money or used for following attack. Secondly, officer cannot execute law enforcement. It means that ALPR service is completely down (all stop) as long as DNS spoofing is worked. It damages to police office(government) reputation critically.
Recommended mitigation
Apply mutual authentication procedure between server and client like TLS(above 1.2)

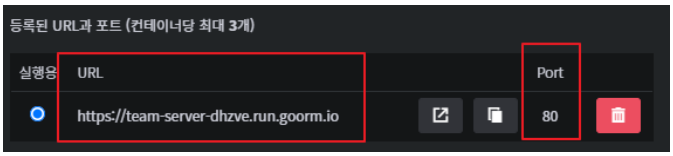
Tools needed	Goorm hosting service

Relative component / source code
Local Server in client side (there is no additional authentication procedure or option, just request with https)

client/web/src/images/server.js

```
173 app.post('/login', function(req, res){
174     let email = req.query.email;
175     let password = req.query.password;
176     let otp = req.query.otp;
177
178     const url = host +"/signin?username="+email+"&password="+password+"&otp="+otp;
179     const request = https.request(url, (response) => {
180         let data = '';
181         response.on('data', (chunk) => {
182             data = data + chunk.toString();
183         });
```

HTTPS converting service from Goorm hosting service



HTTP bind in remote server(server.py)

```python
app = Flask(__name__)

if __name__ == '__main__':
    app = Flask(__name__)
    app.config['DEBUG'] = True
    app.config['SQLALCHEMY_DATABASE_URI'] = config._DB_PATH_
    app.config["SECRET_KEY"] = config._SESSION_SECRET_KEY_

    db.init_app(app)
    if not os.path.isfile('./server.db'):
        with app.app_context():
            db.create_all()
            createdb()

    app.register_blueprint(usermanagement.user_management)
    app.register_blueprint(platequery.plate_query)

    app.run(host='0.0.0.0', port=80)
```

Proof of concept / How to attack

Constraints:

1. Attacker has capability for DNS spoofing in external network (DNS cache poisoning)
2. Attacker should have cert which is not self-signed

   → we use goorm hosting service for certificate which issued from root CA

   (real attackers prepare their site with root CA and IP)

   → But It works with container, cannot access directly remote server in ours container

   with URL or IP

   → So verifying server spoofing, we test with URL modification in code like below,

```
//const host = "https://team-server-dhzve.run.goorm.io";
const host = "https://con-t-ythyc.run.goorm.io";
```

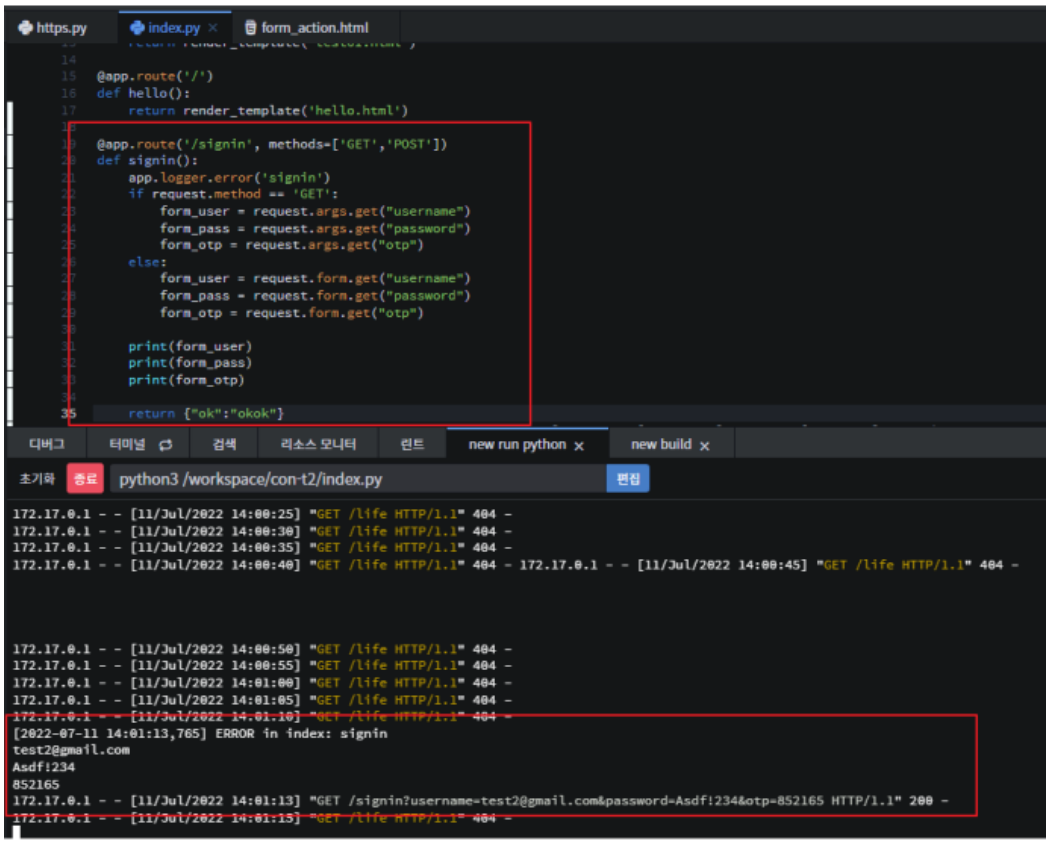
(below is attacker's)

⇨ Assume that DNS spoofing is applied for team3's server URL by modifying URL in source code

Procedure:

1. make fake server with python flask and run in goorm hosting service
2. connect localhost:3000 and try to login
3. server is spoofed
4. check if user credential is received or not in attacker's server
5. check if ALPR service doses not worked well

faker server's impl - signin method : sniffing user credential



without processing signin, we just sniffing
http get method and user credential is passed

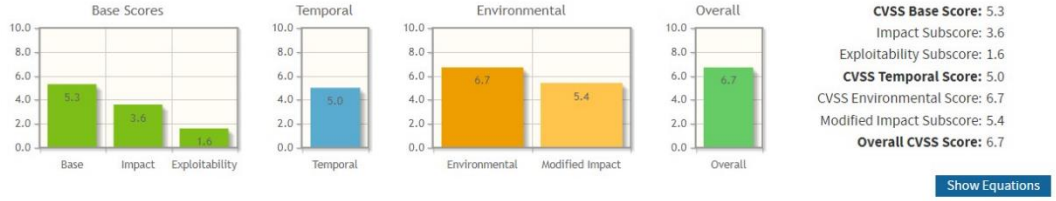
4.2.25 V23 - phishing client web site

Table 84 Vulnerability - V23

Description	[V23] phishing client web site		
CIA	[Availability] [Confidentiality]	Attack vector	[CLIENT/PRIVILEGED]
Approach	[REVIEW/DESIGN]	Exploit technique	[SPOOFING]
Vulnerabilities			
1. S/W structurally, on client side, there are UI app and local host server and there is no any authentication of client webapp. It means that there isn't any protection on client. 2. Attacker can spoof client UI web for phishing and steal sensitive data or prevent officer executing law enforcement.			
CVSS Score	6.7	Severity	Medium

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



CVSS Base Score: 5.3
Impact Subscore: 3.6
Exploitability Subscore: 1.6
CVSS Temporal Score: 5.0
CVSS Environmental Score: 6.7
Modified Impact Subscore: 5.4
Overall CVSS Score: 6.7

Show Equations

CVSS v3.1 Vector

AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:N/A:N/E:P/RL:X/RC:C/CR:H/IR:L/AR:L/MAV:N/MAC:H/MPR:L/MUI:N/MS:U/MC:H/MI:N/MA:N

Consequence / Impact analysis
Phishing client can occur stealing sensitive data (e.g. user credential) and it is sold to illegal group for making money or used for following attack. Secondly, officer cannot execute law enforcement. In other words, officer connect to fake client, try to login, start APLR service. Everything is worked well visually. But fake client does not send any license plate data to server. So, ALPR service is not worked normally.
Recommended mitigation
Run client webapp in server side or apply authentication procedure with web server

Tools needed	socat(for port forwarding), yarn(running webapp)

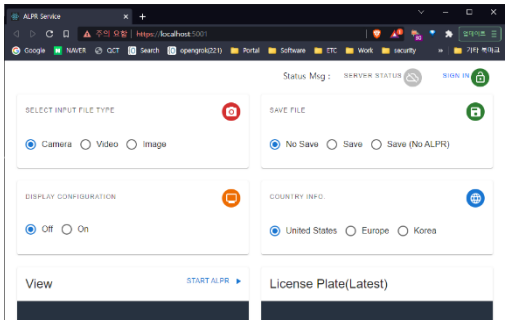
Relative component / source code
Client UI Webapp (design problem)
Proof of concept / How to attack

Constraints:

- client PC has received any malware which attacker injects for forwarding port and running attacker's fake webapp

Procedure:

1. Make fake client webapp and run with port 5001 like below



2. modify login button function

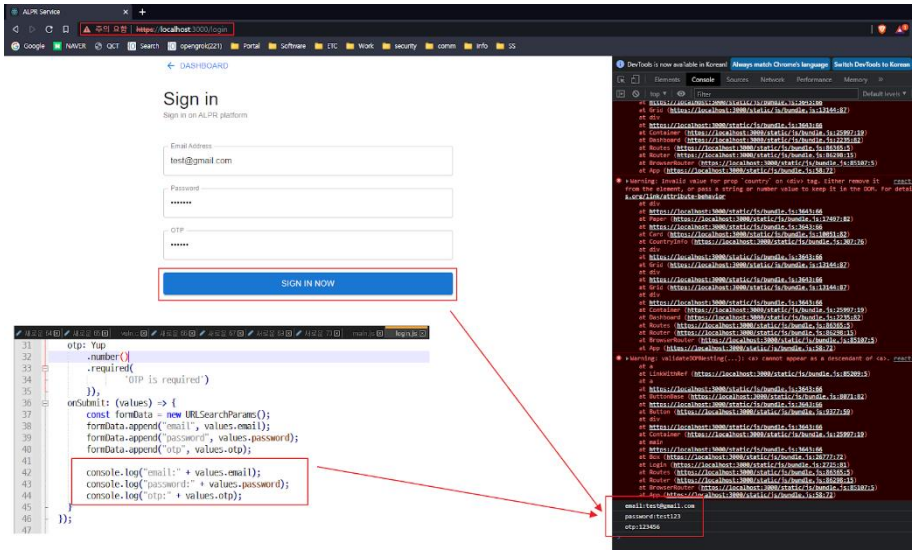
   (print console log → It can be changed to other method to transfer user credential)

3. apply port forwarding 3000 → 5001 with socat

   insert command like below, it forward 3000 port to 5001

   > socat tcp-listen:3000,reuseaddr,fork tcp:127.0.0.1:5001

4. officer connect to localhost:3000(ALPR web service UI URI)
5. It redirects the site to localhost:5001, officer sees the same graphical web page, but it is attacker's phishing site!
6. User attempts to login, then attacker could get user credential



Although the officer connects to the localhost:3000, he connects to the attacker's site (localhost:5001). and his credential is stolen by attacker. And finally, he does not execute law enforcement because client is fake site

4.2.26 V24 - Stolen token, spoofing as valid user, sensitive information could be sniffed

Table 85 Vulnerability - V24

Description	[V24] Stolen token, spoofing as valid user, sensitive information could be sniffed		
CIA	[CONFIDENTIALITY]	Attack vector	[NETWORK]

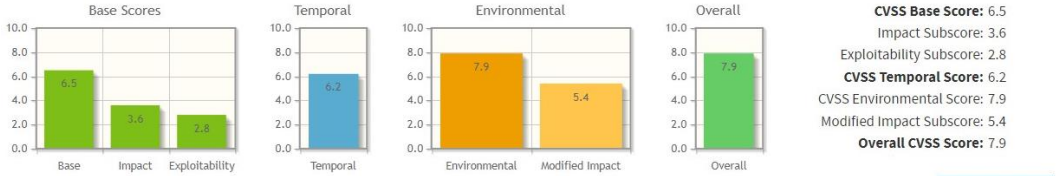
Approach	[REVIEW/CODE] [REVIEW/DESIGN]	Exploit technique	[SNIFFING]

Vulnerabilities
Stolen token, spoofing as valid user, sensitive information could be sniffed
Relative component / source code

CVSS Score	7.9	Severity	High

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



CVSS Base Score: 6.5
Impact Subscore: 3.6
Exploitability Subscore: 2.8
CVSS Temporal Score: 6.2
CVSS Environmental Score: 7.9
Modified Impact Subscore: 5.4
Overall CVSS Score: 7.9

Show Equations

CVSS v3.1 Vector

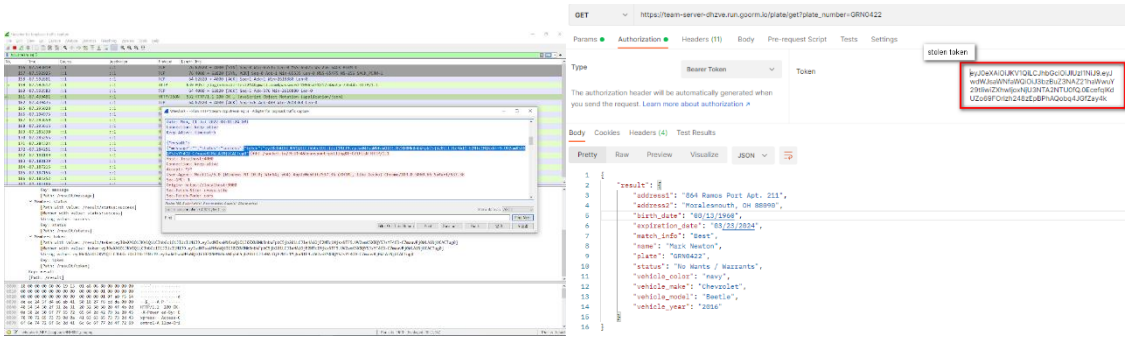
AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N/E:P/RL:X/RC:C/CR:H/IR:L/AR:L/MAV:N/MAC:L/MPR:N/MUI:R/MS:U/MC:H/MI:N/MA:N

Consequence / Impact analysis
An attacker can steal general user credentials such as user name, password, otp seed, token and use this credentials to acquire the plate information
Recommended mitigation
Provide the client interface in the form of WebApp. So, Eliminate exposed IPC channels by operating in one process.

Tools needed	Wireshark

Proof of concept / How to attack

1. packet capture start with localhost
2. connect webpage to https://localhost:3000/
3. proceed login
4. check token in wireshark packet
5. Insert token to header, could be impersonate as validated



4.2.27 V25 - The external server provided by goorm does not support mutual authentication using tls based on pki.

Table 86 Vulnerability - V25

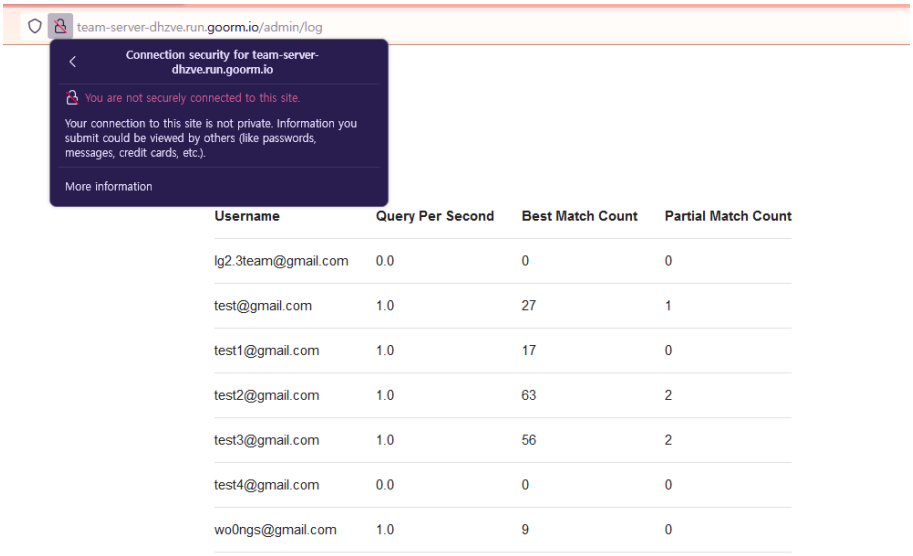
Description	[V25] The external server provided by goorm does not support mutual authentication using tls based on pki.		
CIA	[CONFIDENTIALITY] [INTEGRITY]	Attack vector	[NETWORK]
Approach	[REVIEW/CODE]	Exploit technique	[SNIFFING]
Vulnerabilities			

server.py
- does not implement mutual authentication using tls based on pki.
`app = Flask(__name__)`
`if __name__ == '__main__':`
`  app = Flask(__name__)`
`  app.config['DEBUG'] = True`
`  app.config['SQLALCHEMY_DATABASE_URI'] = config._DB_PATH_`
`  app.config["SECRET_KEY"] = config._SESSION_SECRET_KEY_`
`  db.init_app(app)`
`  if not os.path.isfile('./server.db'):`
`  with app.app_context():`
`  db.create_all()`
`  createdb()`
`  app.register_blueprint(usermanagement.user_management)`
`  app.register_blueprint(platequery.plate_query)`
`  app.run(host='0.0.0.0', port=80)`			

Relative component / source code			
CVSS Score	9.3	Severity	Critical
Common Vulnerability Scoring System Calculator This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score. Base Scores: 9.1 (Base), 5.2 (Impact), 3.9 (Exploitability) Temporal: 8.6 Environmental: 9.3, 5.9 (Modified Impact) Overall: 9.3 **CVSS Base Score:** 9.1 Impact Subscore: 5.2 Exploitability Subscore: 3.9 **CVSS Temporal Score:** 8.6 CVSS Environmental Score: 9.3 Modified Impact Subscore: 5.9 **Overall CVSS Score:** 9.3 Show Equations **CVSS v3.1 Vector** AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N/E:P/RL:X/RC:C/CR:H/IR:H/AR:L/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:H/MA:N			
Consequence / Impact analysis			
An attacker can steal admin or general account information by fake remote server using DNS spoofing An attacker can sniff all information by changing HTTPS to HTTP through the network			
Recommended mitigation			
Using the PKI with SSL/TLS			
Tools needed	Wireshark, Unprivileged new browser		

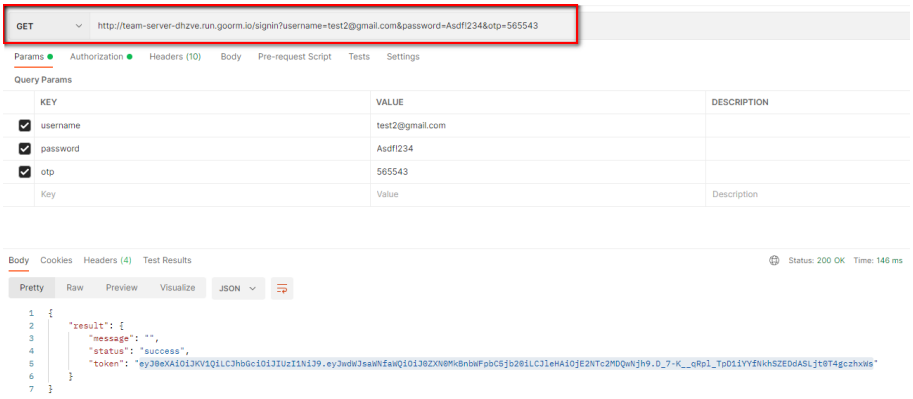
Proof of concept / How to attack

- Allow access to the server using http protocol.



Username	Query Per Second	Best Match Count	Partial Match Count
lg2.3team@gmail.com	0.0	0	0
test@gmail.com	1.0	27	1
test1@gmail.com	1.0	17	0
test2@gmail.com	1.0	63	2
test3@gmail.com	1.0	56	2
test4@gmail.com	0.0	0	0
wo0ngs@gmail.com	1.0	9	0

- available access the API using http



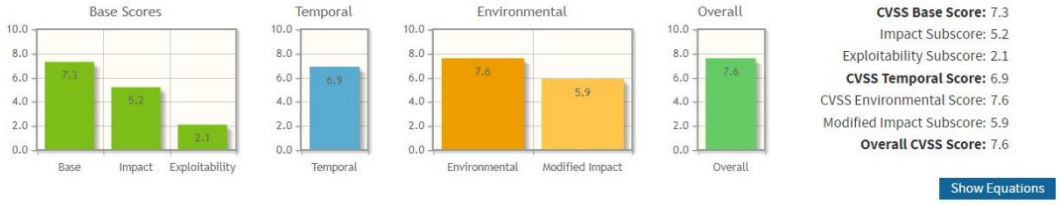
4.2.28 V26 - Available to change https to http by tempering the client's server.js file.

Table 87 Vulnerability - V26

Description	[V26] Available to change https to http by tempering the client's server.js file. because mutual authentication with an external server is not performed.		
CIA	[CONFIDENTIALITY] [INTEGRITY]	Attack vector	[CLIENT/SOURCE]
Approach	[REVIEW/CODE]	Exploit technique	[TAMPERING]
Vulnerabilities			
Available to change https to http by tempering the client's server.js file. because mutual authentication with an external server is not performed.			
Relative component / source code			
CVSS Score	7.6	Severity	High

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



CVSS v3.1 Vector

AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:N/E:P/RL:X/RC:C/CR:H/IR:H/AR:L/MAV:N/MAC:L/MPR:L/MUI:R/MS:U/MC:H/MI:H/MA:N

Consequence / Impact analysis
An attacker can sniff all information by changing HTTPS to HTTP by tampering with the internal server of the client app. And using its information, an attacker can acquire the plate information.
Recommended mitigation
Using the PKI with SSL/TLS
Tools needed: Wireshark
Proof of concept / How to attack

server.js

: tempering code

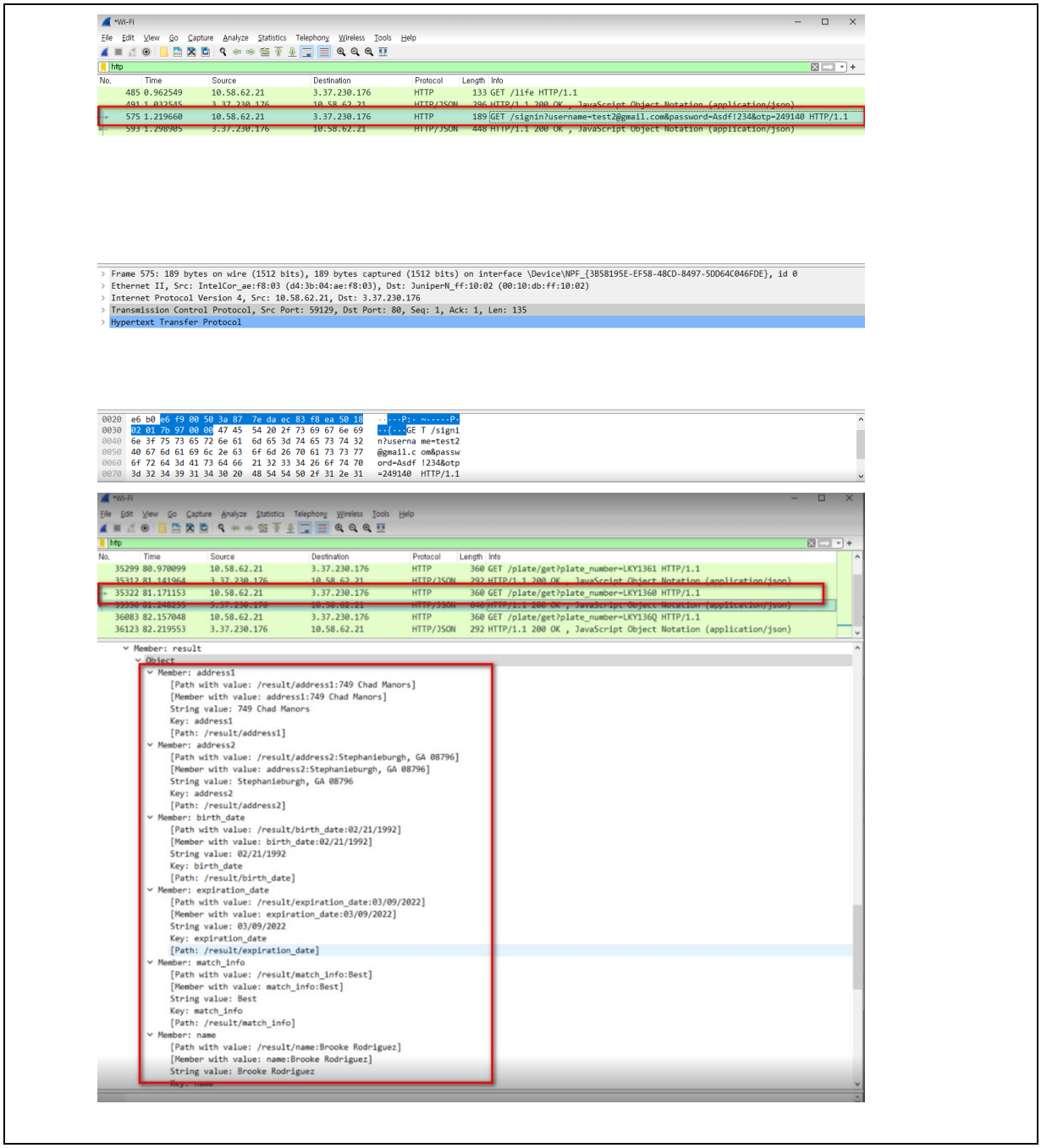
- const https = require('http');

- const host = http://team-server-dhzve.run.goorm.io;

```
const https = require('http');
const ssl_config = require('./ssl-config');

const options = {
    key: ssl_config.privateKey,
    cert: ssl_config.certificate
}

// const server = https.createServer(options, app);
const server = http.createServer(app);
const cors = require('cors');
const io = require('socket.io')(server,{
    cors : {
        origin :"*",
        credentials :true
    }
});

app.use(cors());
app.use(express.urlencoded({
    extended: true
}))

const host = "http://team-server-dhzve.run.goorm.io";

var socketId = "";
io.on('connection', socket => {
```

Communication with an external server has been changed to http.

Requests and responses are made to plain text, and all information is exposed.



4.2.29 V27 - A TOCTOU attack is possible because there is a time gap between file selection and file playback.

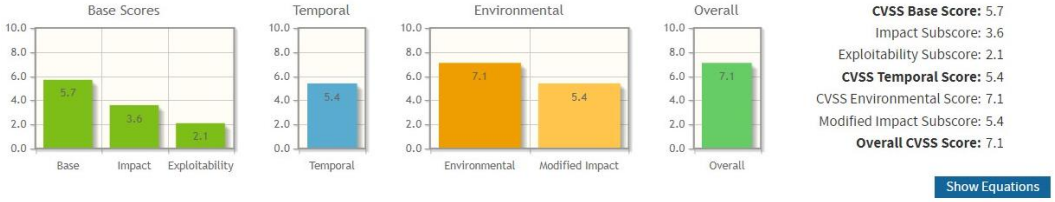
Table 88 Vulnerability - V27

Description	[V27] A TOCTOU attack is possible because there is a time gap between file selection and file playback.		
CIA	Integrity	Attack vector	[CLIENT/SOURCE]
Approach	[REVIEW/DESIGN]	Exploit technique	[TAMPERING]
Vulnerabilities			
A TOCTOU attack is possible because there is a time gap between file selection and file playback.			
Relative component / source code			

Source/client/web/src/images/playback files			
CVSS Score	7.1	Severity	High

Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



CVSS Base Score: 5.7
Impact Subscore: 3.6
Exploitability Subscore: 2.1
CVSS Temporal Score: 5.4
CVSS Environmental Score: 7.1
Modified Impact Subscore: 5.4
Overall CVSS Score: 7.1

Show Equations

CVSS v3.1 Vector
AV:N/AC:L/PR:L/UI:R/S:U/C:N/I:H/A:N/E:P/RL:X/RC:C/CR:L/IR:H/AR:L/MAV:N/MAC:L/MPR:L/MUI:R/MS:U/MC:N/MI:H/MA:N

Consequence / Impact analysis

1. Attacker can change playback file, so a criminal driver can evade to be looked up

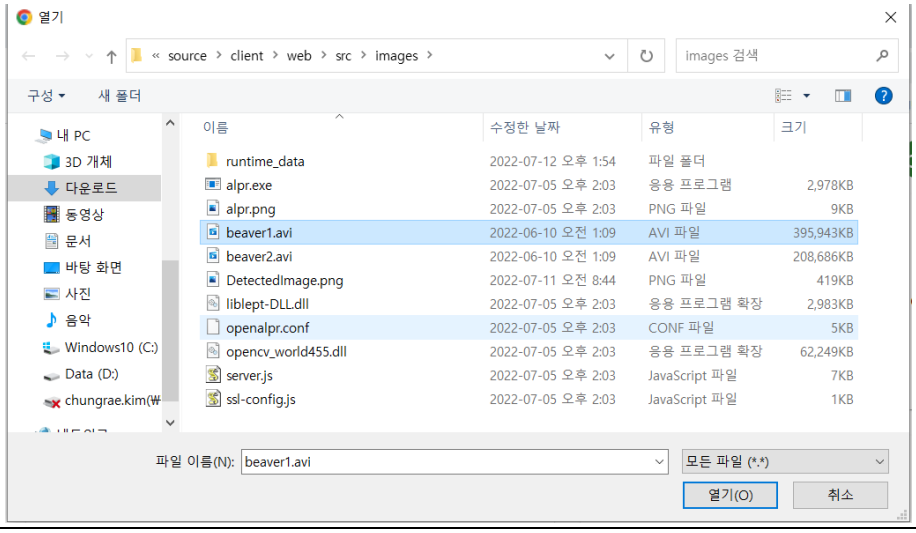
Recommended mitigation

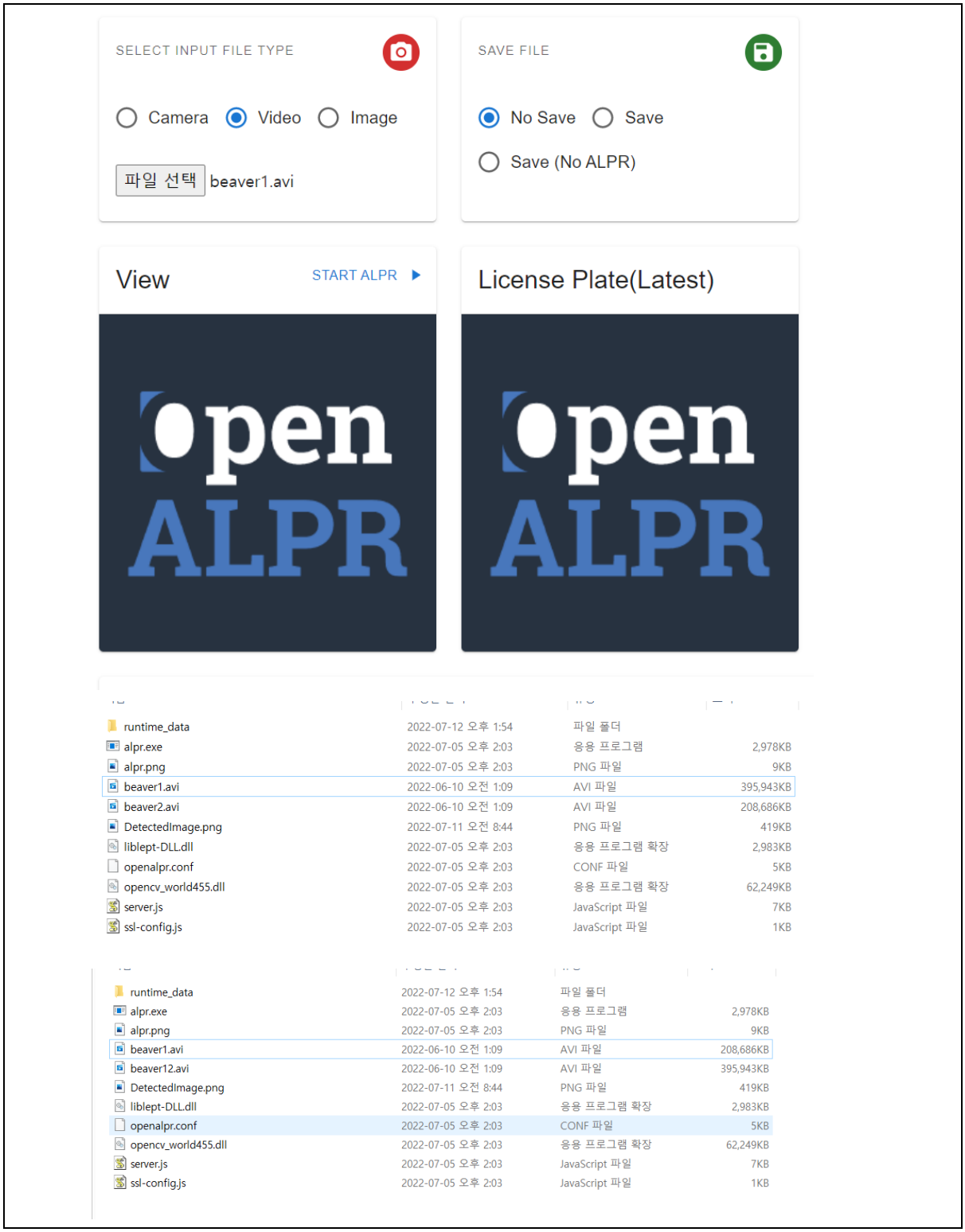
1. Do not use an additional button to start ALPR
2. If choosing the file is done, run ALPR right now

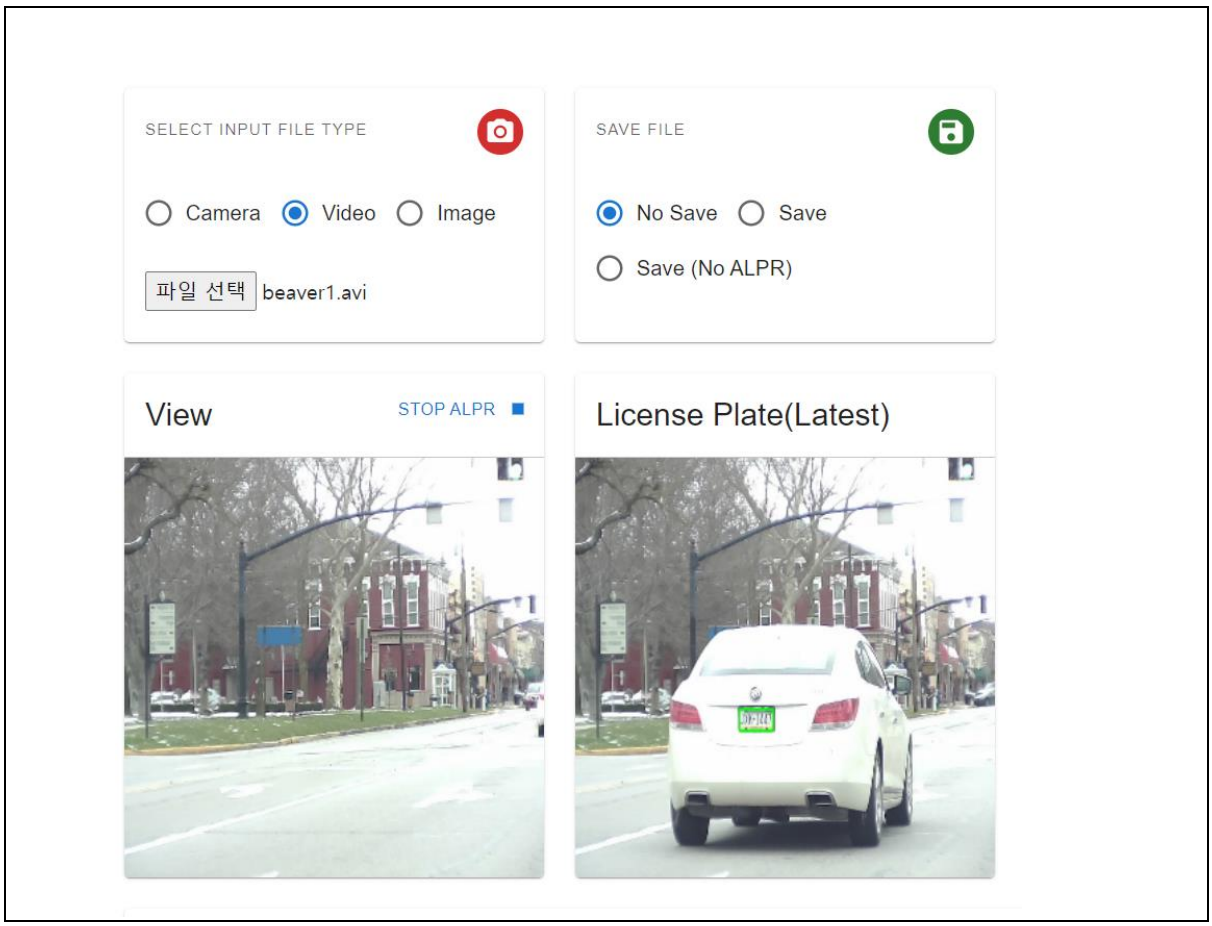
Tools needed	NA

Proof of concept / How to attack

1. SELECT INPUT FILE TYPE to Video
2. Select a playback file (beaver1.avi) via “파일선택” button
3. if attacker change file beaver2.avi to beaver1.avi before pushing “START ALPR”, the officer will run "beaver2.avi" named "beaver1.avi" instead of original "beaver1.avi"




SELECT INPUT FILE TYPE
Camera
Video
Image
파일 선택
beaver1.avi
SAVE FILE
No Save
Save
Save (No ALPR)
View
START ALPR
License Plate(Latest)
runtime_data 2022-07-12 오후 1:54 파일 폴더
alpr.exe 2022-07-05 오후 2:03 응용 프로그램 2,978KB
alpr.png 2022-07-05 오후 2:03 PNG 파일 9KB
beaver1.avi 2022-06-10 오전 1:09 AVI 파일 395,943KB
beaver2.avi 2022-06-10 오전 1:09 AVI 파일 208,686KB
DetectedImage.png 2022-07-11 오전 8:44 PNG 파일 419KB
liblept-DLL.dll 2022-07-05 오후 2:03 응용 프로그램 확장 2,983KB
openalpr.conf 2022-07-05 오후 2:03 CONF 파일 5KB
opencv_world455.dll 2022-07-05 오후 2:03 응용 프로그램 확장 62,249KB
server.js 2022-07-05 오후 2:03 JavaScript 파일 7KB
ssl-config.js 2022-07-05 오후 2:03 JavaScript 파일 1KB
runtime_data 2022-07-12 오후 1:54 파일 폴더
alpr.exe 2022-07-05 오후 2:03 응용 프로그램 2,978KB
alpr.png 2022-07-05 오후 2:03 PNG 파일 9KB
beaver1.avi 2022-06-10 오전 1:09 AVI 파일 208,686KB
beaver12.avi 2022-06-10 오전 1:09 AVI 파일 395,943KB
DetectedImage.png 2022-07-11 오전 8:44 PNG 파일 419KB
liblept-DLL.dll 2022-07-05 오후 2:03 응용 프로그램 확장 2,983KB
openalpr.conf 2022-07-05 오후 2:03 CONF 파일 5KB
opencv_world455.dll 2022-07-05 오후 2:03 응용 프로그램 확장 62,249KB
server.js 2022-07-05 오후 2:03 JavaScript 파일 7KB
ssl-config.js 2022-07-05 오후 2:03 JavaScript 파일 1KB



5 Lessons Learned

5.1 Lesson #1

- ✓ **Thinking like an attacker!**

- ✓ **Not all vulnerabilities can be eliminated. However, it should be possible to remove as much as possible at each stage of the SDL cycle.**

- ✓ **Security issues also arise where there is no direct relationship with the security requirements. It seems to occur mainly in weak design, implementation, and the interface between modules.**

- ✓ **Software requirement is easily implemented with COTS, opensource and so on. But without deep consideration about threat analysis, security requirement and design, It can occur tremendous vulnerabilities**

- ✓ **By finding vulnerabilities in target software, we could find hidden security issue and mitigation reversely**

5.2 Learned methodologies, techniques and tools

- ✓ **Threat modeling using STRIDE → To identify and eliminate potential vulnerabilities**

- ✓ **Understanding into the most common security risks so that we could use the findings of the report as part of our class-mates security practices Using OWASP**

- ✓ **How much using code-review and utilizing some tools is important**

- ✓ **Utilize wireshark, Chrome inspector and a few types of static analysis tools**