Automatic License Plate Recognition (ALPR) System

Team3 PURPLE

- Kibong Song
- Brian Moon
- Seungkyu Lee
- Haenggi Lee
- Seongsik Kim
- Youngmi Choi

Table of Contents

Version	Date	Status	Author
1.0	2022.07.05	1st release	team3
1.1	2022.07.12	organization, schedule is added.	team3

0. Organization and Schedule

Role and Responsibilities

- Developer Artifacts and Documentation
  - Youngmi Choi (최영미), Seungkyu Lee (이승규)
- Client Design and Implementation
  - Seongsik Kim (김성식), Haenggi Lee (이행기)
- Server Design and Implementation
  - Kibong Song (송기봉), Gyunggui Moon (문경귀)
- Mentor
  - David Belasco (데이비드)

Schedule

Task Name	Week.1					Week.2					Week.3				
1.Define Roles & Responsibilities															
2.Requirement Analysis															
2.1. Define Terminology															
2.2. Analysis Customer Requirement															
3. Security Analysis															
3.1. Define Security Goals															
3.2. Define Asset Identification															
3.3. Threat Analysis															
3.4. Security Risk Assessment															
3.5. Mitigating Threats															
3.6. Define Product Specification															
4. Architecture Design															
5. SW implementation															
5.1. SW Vulnerability Evaluation															
5.2. Check Secure Coding Rule															
6. Test															

1. Terminology and Definitions

This section provides a description of the terms and definition that will be used throughout this document.

Term	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].
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].
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.
Use Case	Describes the functional interaction between an application and its users.
Utility	The system and associated processes shall be designed, implemented, operated and maintained so that the utility of their assets is maintained throughout their life cycle
Security Use Case	Specifies the security requirements that the application shall successfully protect itself from its relevant security threats.
Vulnerability	Part of the information security infrastructure that could represent a weakness to attack in the absence of a control.

Sensitive Data	User Credentials Social Security number or other identifying information Credit card numbers or other financial information Health information Private keys or other data that could be used to decrypt encrypted information System or application information that can be used to more effectively attack the application
ALPR system	Automatic License Plate Recognition System

2. Customer Requirement

2.1 Client

2.1.1 Functional Requirements

1. 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.
2. The system should allow a law enforcement officer to select and save retrieved information locally.
3. 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.
4. The system should allow officers to configure computed camera / playback frames per second, average time per frame, jitter and frame number.
5. The system should allow the officer to choose between using a live camera and playback file in the UI.

2.1.2 Non-Functional Requirements

1. The system shall allow an officer to access the ALPR system through a secure web interface.
2. Lost or compromised credentials must be handled in a reasonable way.
3. The system should provide secure communication between the client application and to the backend license plate database lookup system.
4. The system should read images from the vehicle camera or a playback file and identify license plates for evaluation.
5. The system should perform the ALPR function in real-time while maintaining a frame rate of at least 25fps.
6. 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.
7. 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.
8. The system should provide an area in the user interface that always contains the current camera /playback view.
9. The ability to detect network connectivity issues with the backend server within 5 seconds and automatically resolve the communication issue if possible.
10. The system should alert officers of any communication errors or failures.
11. The system must fetch vehicle information in no more than 10 seconds as officers are often making queries in real time.

2.2 Server

2.2.1 Functional Requirements

1. Support license plate queries.
2. Authenticate remote laptop users.
3. Support configurable values via a configuration file

2.2.2 Non-Functional Requirements

1. Ensure secure communication with the client applications.
2. Support multiple users.
3. 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.
4. Track the average number of queries per second for each user and overall queries per second, for all users.
5. Track the number partial matches and no matches for each user and all users

2.3 Common Information

1. Ensuring that all software in both applications are architected and coded to be secure and free of vulnerabilities.
2. Conduct proper fault/error detection, recovery and reporting.
3. Ensure the developed software adheres to the company coding standard and quality standards.
4. Ensure the developed software is adequately tested.
5. The application should allow a law enforcement officer to use a video feed/picture to identify a license plate and then query that license plate number to determine if there are any outstanding fines or warrants against the vehicle's owner.
6. The proposed system should be a client server system where the client is an on-board computer in a police vehicle or mobile device carried by an officer.

Initial customer requirements have some parts that are ambiguous or unclear, and have been detailed through discussion with the customer (David Belasco).

In the next step, the security requirements will be elicited using the SQUARE methodology.
The SQUARE methodology to help organizations build security into the early stages of the production life cycle.

2.4 Assumptions

The following assumptions were made according to "Keep It Short and Simple" principle to remove ambiguity of given requirements.
1. All recognition results from the server are maintained on the client view during execution, allowing to see information with or without the alert.
2. The last recognized plate image of certain vehicles should be viewed during continuous recognition.
3. Jitter is not considered in this project. It is related to a performance issue.
4. Allowing multiple user connections is supported by the public cloud hosting service we use.
5. Partial match in this project means that the system can retrieve vehicle information from the server using a plate number without the first or last letter missing.
6. Log is not encrypted.
7. Frame rate limitation of video or live input is not considered in this project.
8. Mobile client environment is not considered in this project.
9. If a user has not logged in, only plate recognition is available without querying to the server.

3. Security goals

3.1 Business goals

The system allows authorized users to make decisions based on the information provided by the image recognition web-based system.

3.2 Security goals

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.

4. System definition

4.1 Server

1. Hardware : Cloud hosting platform
2. Environment : WSGI server - it will run the web app based on python web framework
3. OS : Linux
4. SW module : plate query module, user management module, config module
5. data : access control list, vehicle plate information

4.2 Client for laptop

1. Hardware : laptop computer
2. Environment : ReactJS, NodeJS, SocketIO
3. OS : Windows
4. SW module : GUI, Connection Module
5. data : Video file, License Plate Image

5. Software inventory

5.1 Reuse

ALPR source code from CMU.
The source code provided by the customer does not proceed with the analysis to see if it meets the requirements.

5.2 Open Source SW

1. Server
    a. security goal
        i. The server system shall be available for use when needed.
        ii. The confidentiality and integrity of the system's DB shall be maintained.
        iii. The server system should support basic security features.
    b. preliminary security requirements.
        i. All sensitive data should be encrypted.
        ii. Whole user operations should be monitored.
    c. tradeoff analysis
        i. Additional time is required for encryption/decryption when serving data from the server.
        ii. Need a database space for logging all user actions and an interface for monitoring.
    d. open source name (python package)
        i. click
        ii. itsdangerous
        iii. jinja2
        iv. werkzeug
        v. flask
        vi. flask-sqlalchemy
        vii. sqlalchemy-utils
        viii. pyotp
        ix. pyjwt
        x. cryptography
        xi. faker
        xii. faker-vehicle
2. Client
    a. security goal
        i. The system shall allow an officer to access the ALPR system through a secure web interface.
        ii. The system must use two factor authentication for sign on and user credentials must be protected.
        iii. The system should provide secure communication between the client application and to the backend license plate database lookup system.
    b. preliminary security requirements
        i. The web interface uses HTTPS.
        ii. Enter the google OTP number in the sign-in step.
        iii. The communication method is HTTPS.
    c. tradeoff analysis
        i. n/a
    d. open source name (npm package)
        i. react
        ii. react-bootstrap
        iii. react-dom

iv. react-perfect-scrollbar
v. react-router-dom
vi. react-scripts
vii. socket.io-client
viii. web-vitals
ix. yup
x. axios
xi. formik
xii. http-proxy-middleware
xiii. @mui/icon/ns-material
xiv. @mui/material
xv. npm
xvi. express
xvii. path
xviii. https
xix. http
xx. cors
xxi. socket.io
xxii. child_process
xxiii. socket.io-client-cpp
xxiv. openalpr(provided by CMU)

5.3 COTS

Not used

6. Asset Identification

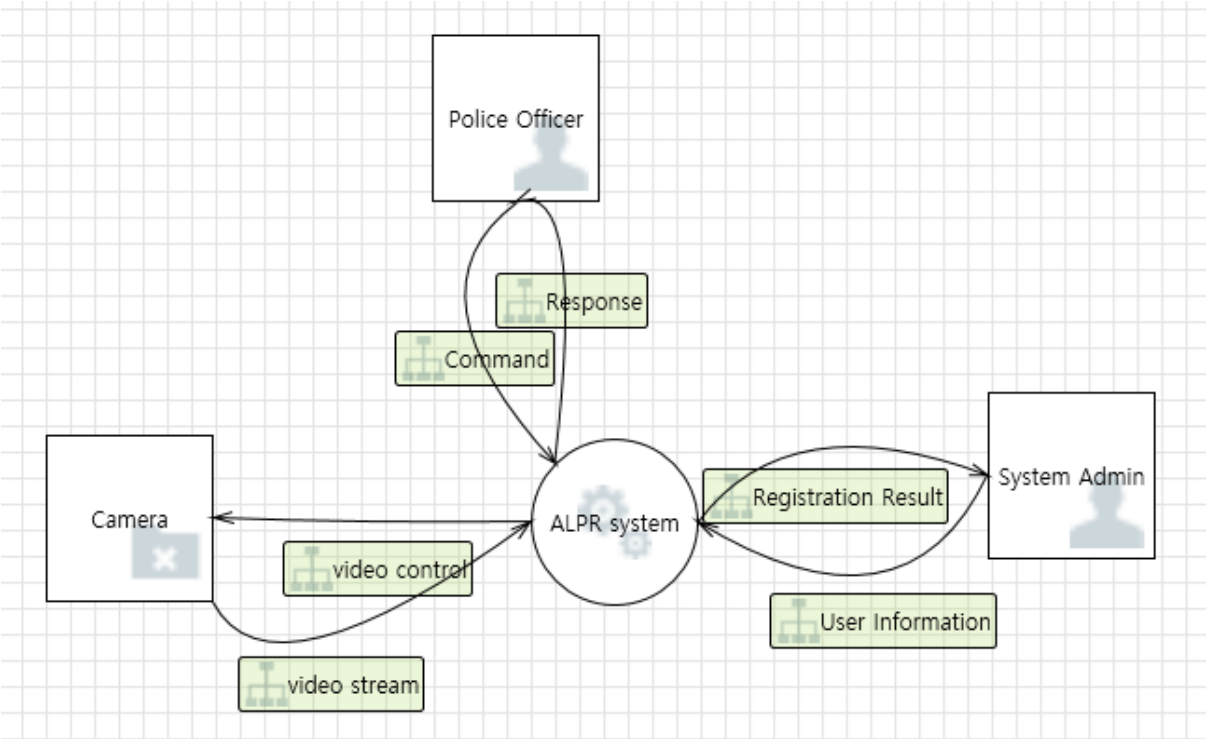
It was considered an important asset that influenced users to make informed decisions and derived assets.

System	Assets	Security Characteristics	Damage Scenario
External Entity	police officer	Authentication	Attackers may steal the user's ID, password, and OTP information, access the laptop client app, and assume the user's ID and password.
External Entity	admin	Integrity, Authorization	Attackers can acquire administrator rights through an escalation of privilege attack to compromise the DB or leak information.
Server	user DB	Integrity	Attackers can access the user DB and change the ID and password information.
Server	user DB	Confidentiality	Attackers can access user DB and obtain personal information.
Server	Drive/Car DB	Integrity	Attackers may access the Drive/Car DB to change vehicle information and owner information.
Server	Drive/Car DB	Confidentiality	Attackers can access the Drive/Car DB to obtain vehicle information and owner information.
Server	Drive/Car DB	Availability	Attackers can paralyze the ALPR system by performing excessive queries on the Drive/Car DB.
Client	Drive/Car Info. Storage	Integrity	Attackers may access Drive/Car Info. Storage to change vehicle information and owner information.
Client	Drive/Car Info. Storage	Confidentiality	Attackers can access Drive/Car Info. Storage to obtain vehicle information and owner information.
Network Interface	laptop client ↔ server	Integrity, Confidentiality	Attackers may cause a malfunction of the system by stealing and modulating data transferred between the client and the server.
Network Interface	laptop client ↔ server	Availability	Attackers can cause a network overload.

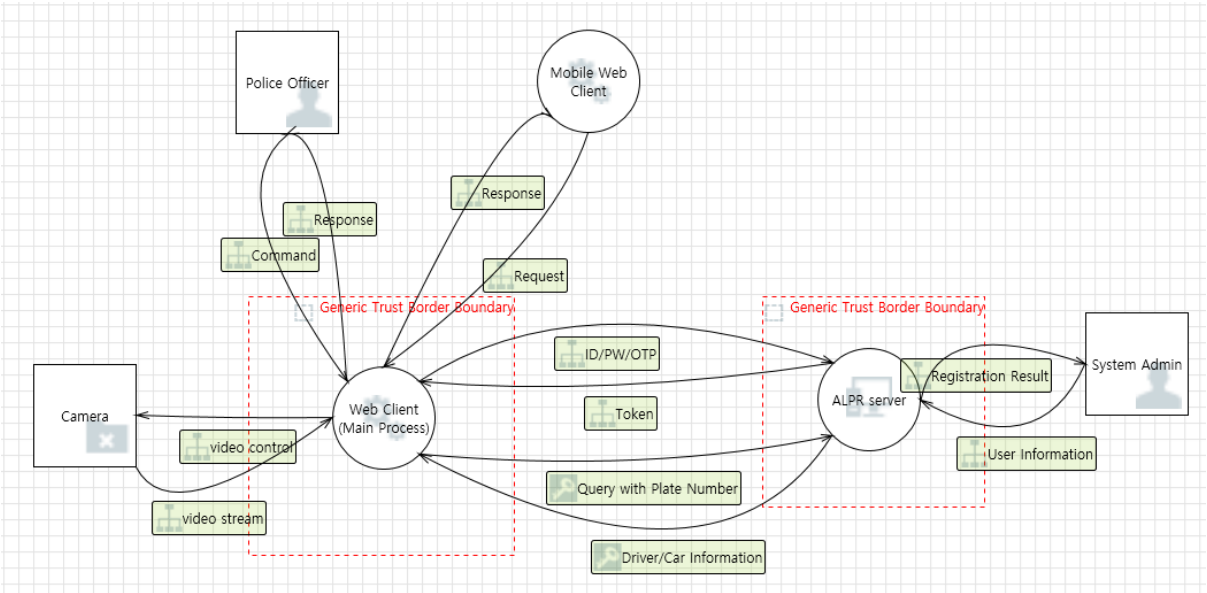
7. Threat Analysis

7.1 STRIDE

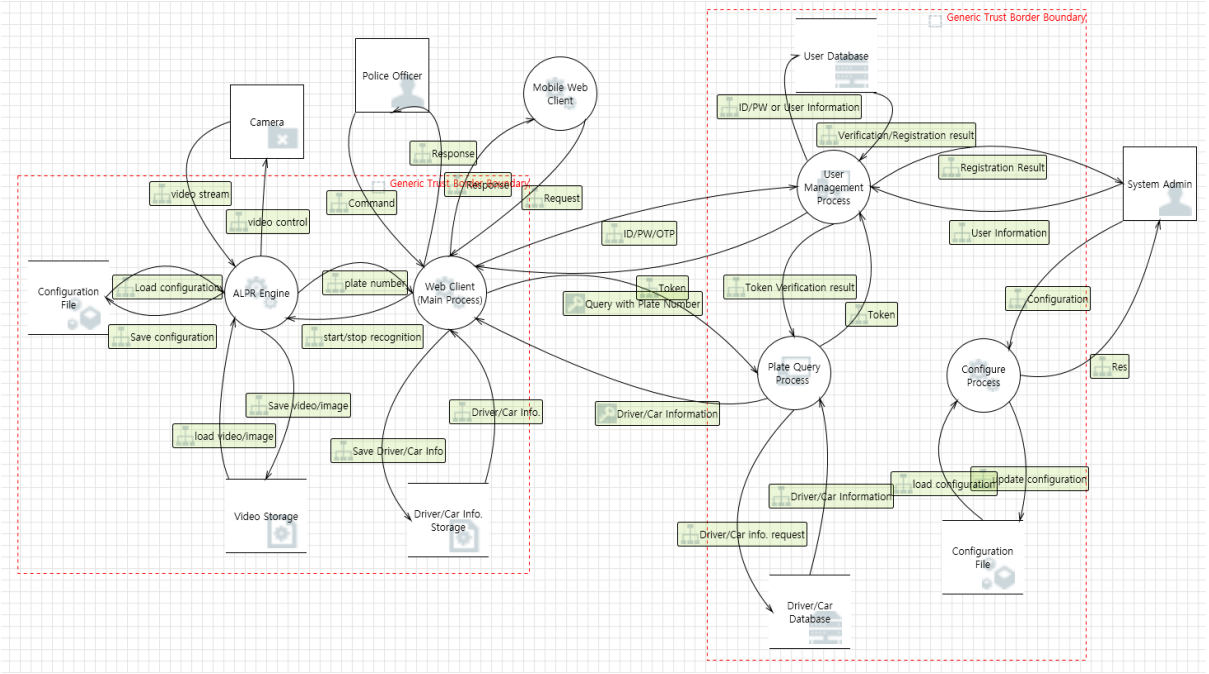
1. Development Artifacts
    a. DFD 0



b. DFD 1



c. DFD 2

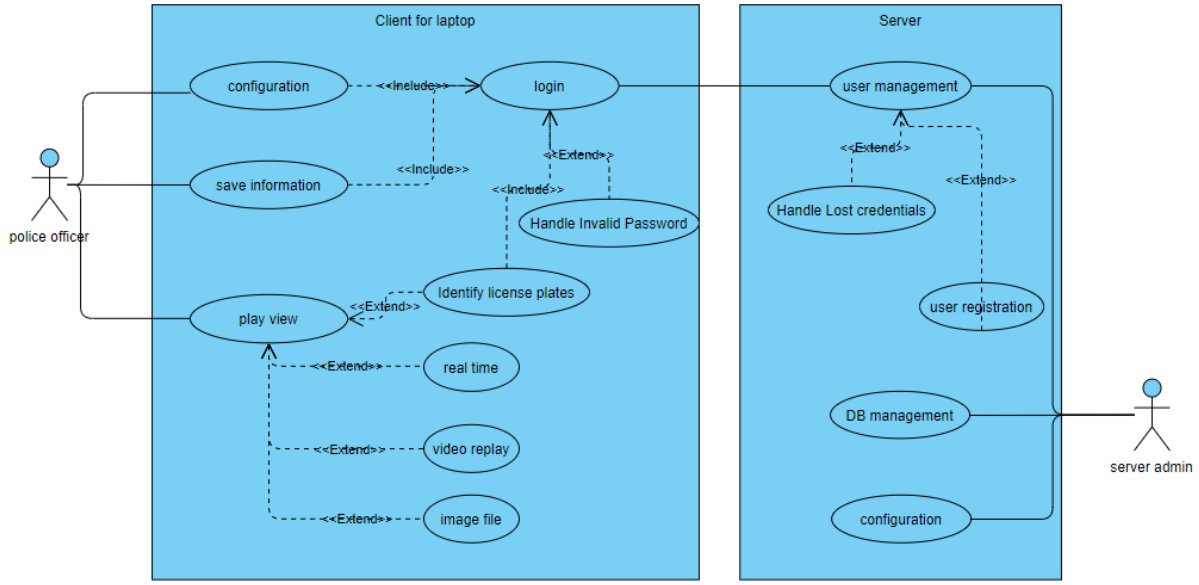


2. Threats list
    a. 145 threats were derived using Microsoft threat modeling tool.
    b. You can find the “All threats” in the “Team3_Purple_ALPR_Phase.1_ThreatModeling.pdf" document in the deliverables.

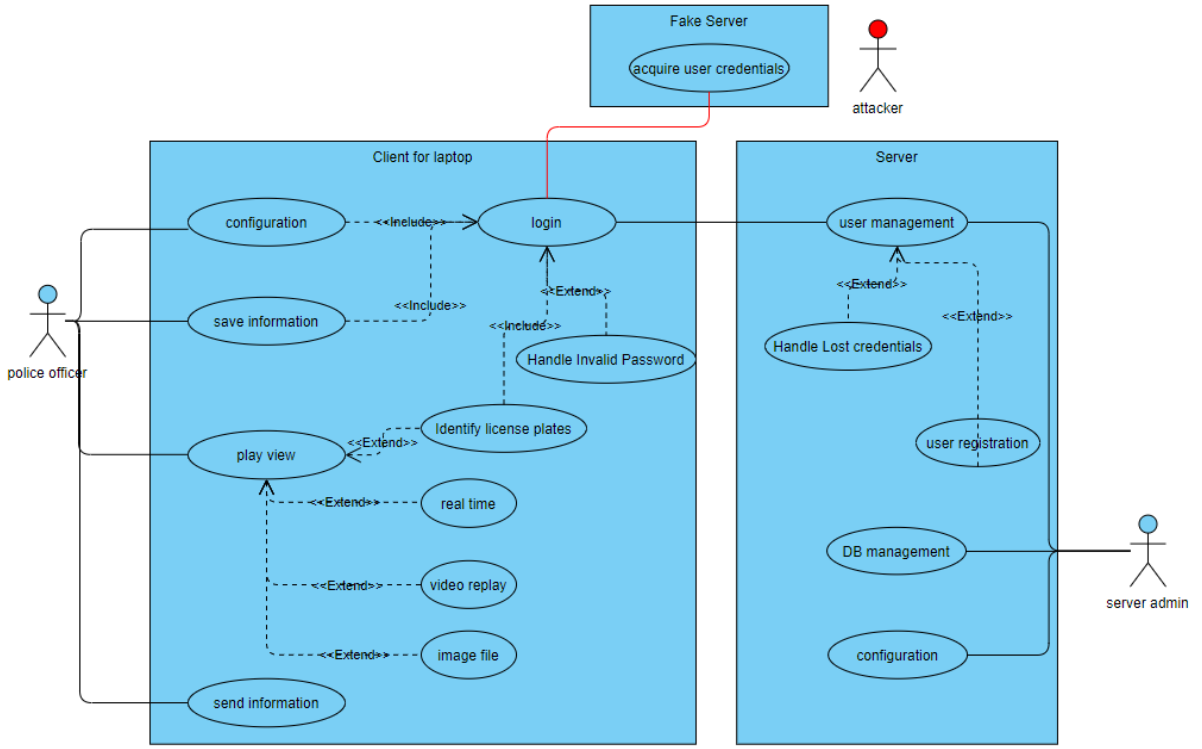
7.2 PnG

PnG was briefly used to derive threats from the attacker's point of view.

1. Development Artifacts
    a. use case diagram



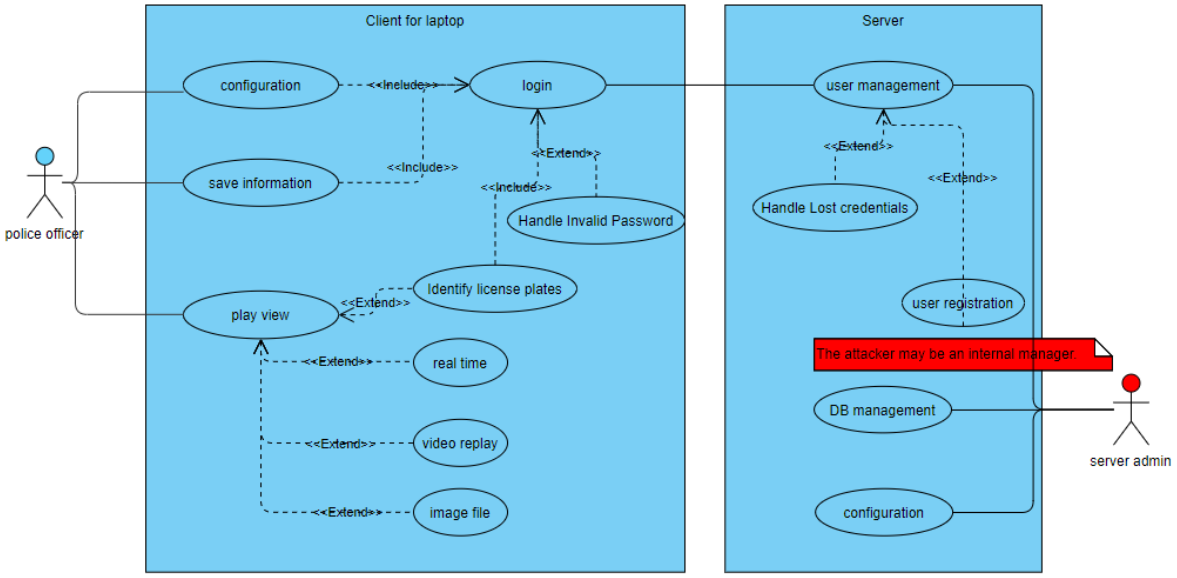
b. misuse case diagram



2. PnG Type #1 : boring police officer
   a. motivations :
      i. I'm so free and bored these days. I want to create an incident or an accident.
   b. goals :
      i. I want to create a chaotic situation by malfunctioning the ALPR system used by road patrol officers.
   c. skills :
      i. I know how to distribute the client application.
      ii. I have a friend who has experience in hacking the client/server system.
   d. misuse cases :

i. boring police officer tells fellow police officers they need a client update and give them a fake server link
ii. All user account information accessed on the fake server is hijacked.
iii. The attacker accesses the client application with the acquired user account information to freely use and manipulate the desired information.

e. threats :
i. login spoofing



3. PnG Type #2 : Backend license plate server administrator without a sense of duty
a. motivations :
i. I want to get revenge on my ex-girlfriend after receiving a breakup notice from her.
ii. After the successful revenge of his girlfriend, there was a client who wanted to take small private revenge, so he earned extra income.
b. goals :
i. I want to make a fine by leaving a record of non-payment of the fine on the DB status of my ex-girlfriend's vehicle.
ii. I leave a false record of non-payment of fines in the requested vehicle DB.
c. skills :
i. Good knowledge of license plate server DB schema and how to change it
ii. Ability to delete DB access logs
d. misuse cases :
i. Falsely change the non-payment of fine information in the vehicle status information of the ex-girlfriend or client and remove change logs
e. threats
i. repudiation of authorized user

7.3 Critical threats selection

Since the threat affecting the asset identified in Section 6 is important, the critical threat was selected based on this.
You can find the “Critical treats” in the “Team3_Purple_ALPR_Phase.1_ThreatModeling.pdf" document in the deliverables.

In the case of an actual project, the entire project must be analyzed, but since it is a project for assignment with time constraints, some critical threats were selected and subsequent activities were carried out.

8. Security Risk Assessment

8.1 Risk Rating

We evaluated the risk of the asset inside the system among the **identified assets** using the OWASP method.

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

1. Drive/Car DB in server

Likelihood								
Threat agent factors					Vulnerability factors			
Skill level	Motive	Opportunity	Size		Ease of discovery	Ease of exploit	Awareness	Intrusion detection
6 - Network and programming skills	9 - High reward	4 - Special access or resources required	9 - Anonymous Internet users		7 - Easy	7 -	9 - Public knowledge	3 - Logged and reviewed
			Overall likelihood:	6.750	HIGH			

Technical Impact					Business Impact			
Loss of confidentiality	Loss of integrity	Loss of availability	Loss of accountability		Financial damage	Reputation damage	Non-compliance	Privacy violation
9 - All data disclosed	9 - All data totally corrupt	3 -	7 - Possibly traceable		8 -	8 -	9 -	9 - Millions of people
Overall technical impact:		7.000	HIGH		Overall business impact:		8.500	HIGH
			Overall impact:	7.750	HIGH			

Overall Risk Severity : CRITICAL

2. User DB in server

Likelihood								
Threat agent factors					Vulnerability factors			
Skill level	Motive	Opportunity	Size		Ease of discovery	Ease of exploit	Awareness	Intrusion detection
6 - Network and programming skills	9 - High reward	4 - Special access or resources required	9 - Anonymous Internet users		7 - Easy	7 -	9 - Public knowledge	3 - Logged and reviewed
			Overall likelihood:	6.750	HIGH			

Technical Impact					Business Impact			
Loss of confidentiality	Loss of integrity	Loss of availability	Loss of accountability		Financial damage	Reputation damage	Non-compliance	Privacy violation
5 - Extensive critical data disclosed	5 - Extensive slightly corrupt data	3 -	7 - Possibly traceable		5 -	4 - Loss of major accounts	4 -	4 -
Overall technical impact:		5.000	MEDIUM		Overall business impact:		4.250	MEDIUM
			Overall impact:	4.625	MEDIUM			

Overall Risk Severity : HIGH

3. client ↔ server

Likelihood								
Threat agent factors					Vulnerability factors			
Skill level	Motive	Opportunity	Size		Ease of discovery	Ease of exploit	Awareness	Intrusion detection
5 - Advanced computer user	7 -	4 - Special access or resources required	9 - Anonymous Internet users		9 - Automated tools available	9 - Automated tools available	9 - Public knowledge	3 - Logged and reviewed
			Overall likelihood:	6.875	HIGH			

Technical Impact					Business Impact			
Loss of confidentiality	Loss of integrity	Loss of availability	Loss of accountability		Financial damage	Reputation damage	Non-compliance	Privacy violation
9 - All data disclosed	9 - All data totally corrupt	9 - All services completely lost	7 - Possibly traceable		9 - Bankruptcy	9 - Brand damage	9 -	9 - Millions of people
Overall technical impact:		8.500	HIGH		Overall business impact:		9.000	HIGH
			Overall impact:	8.750	HIGH			

Overall Risk Severity : CRITICAL

8.2 Threat Priority

1. The priority of selected threats is critical. This is because all of the critical threats selected above are DB managed by the server and query parts between the server and the client.
2. You can find the “Critical treats” in the “Team3_Purple_ALPR_Phase.1_ThreatModeling.pdf" document in the deliverables.

9. Mitigating Threats

The mitigation measures did not devise new ones, but were used referring to existing ones.
In addition, they tried to use mitigation measures that could prevent multiple threats at once.
References)
https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-authentication
https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

You can find the “Critical threats” including mitigation in the
“Team3_Purple_ALPR_Phase.1_ThreatModeling.pdf" document in the deliverables.

10 security requirements (SR1 to SR10) were derived as threat modeling, and 10 requirement specifications (RS37 to RS47) applied with mitigation measures were prepared.

You can find the details in the “ALPR REQ. tracking” in the
“Team3_purple_ALPR_Phase.1_Requirements.pdf” document in the deliverables.

10. Final product specification

10.1 Categorize requirements as to level

The requirement specifications are classified according to the following criteria.

Category	Meanings
System level	information on the requirements for a system.
Software level	in-depth descriptions of the software that will be developed.
Technical constraints	the Non-Functional aspects of a system or component, such as restrictions on technology, resources or techniques to be used.
Business constraints	the Non-Functional aspects of a system or component, such as restrictions on business.

10.2 Prioritize requirements

The priority of the requirements was classified as Essential, conditional, and optional, and was determined by discussing it with the stakeholder from a security perspective.
Requirements set as Optional were excluded from the scope of this development.

Priority	Meanings
Essential	the product is not acceptable unless these requirements are satisfied
Conditional	would enhance the product, but the product is not unacceptable if absent
Optional	functions that may or may not be worthwhile

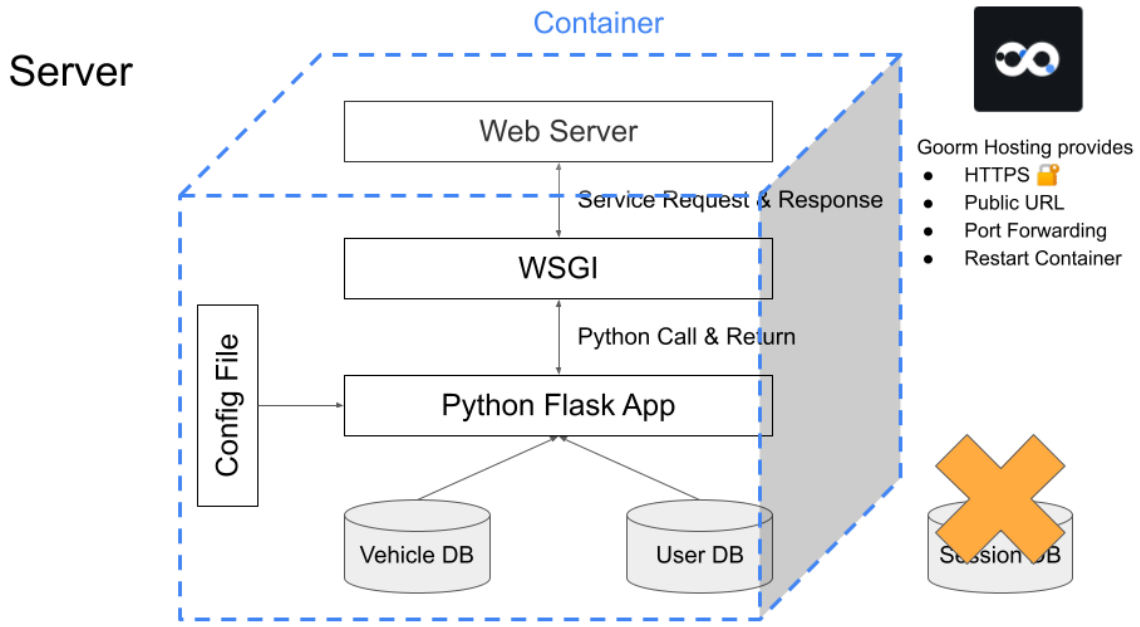
10.3 Requirements inspection

For requirement inspection, stakeholders gathered to conduct requirements walkthrough.
The checklist used for inspection can be found in the deliverables.

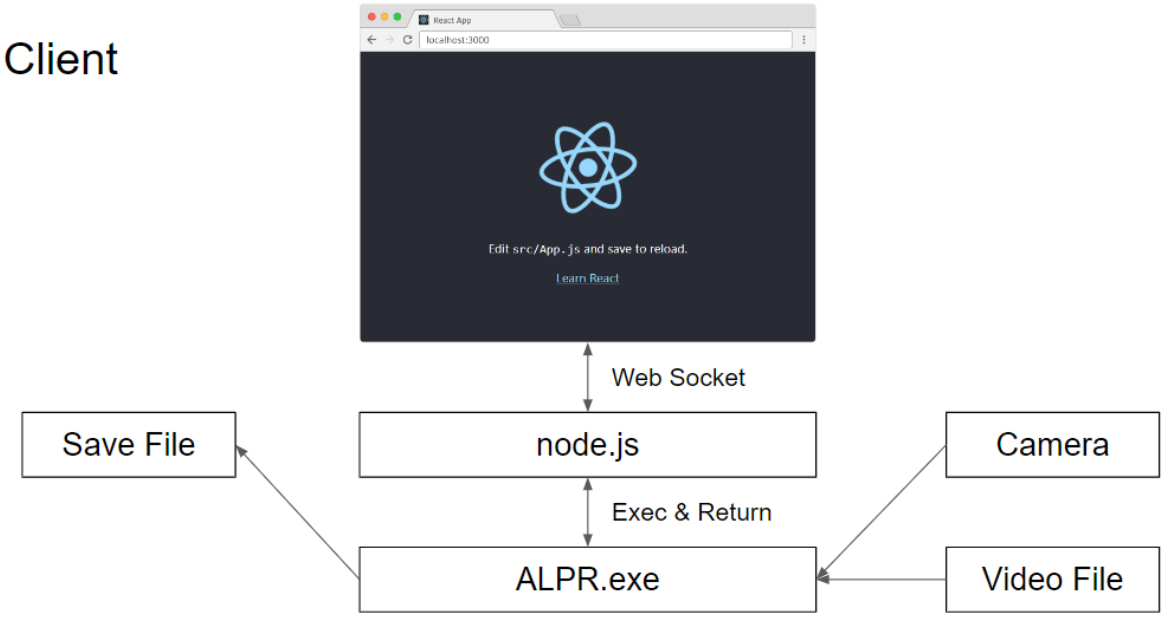
You can find the details in the “ALPR REQ. tracking” in the “Team3_purple_ALPR_Phase.1_Requirements.pdf” document in the deliverables.

11. Architecture Design

11.1 Server

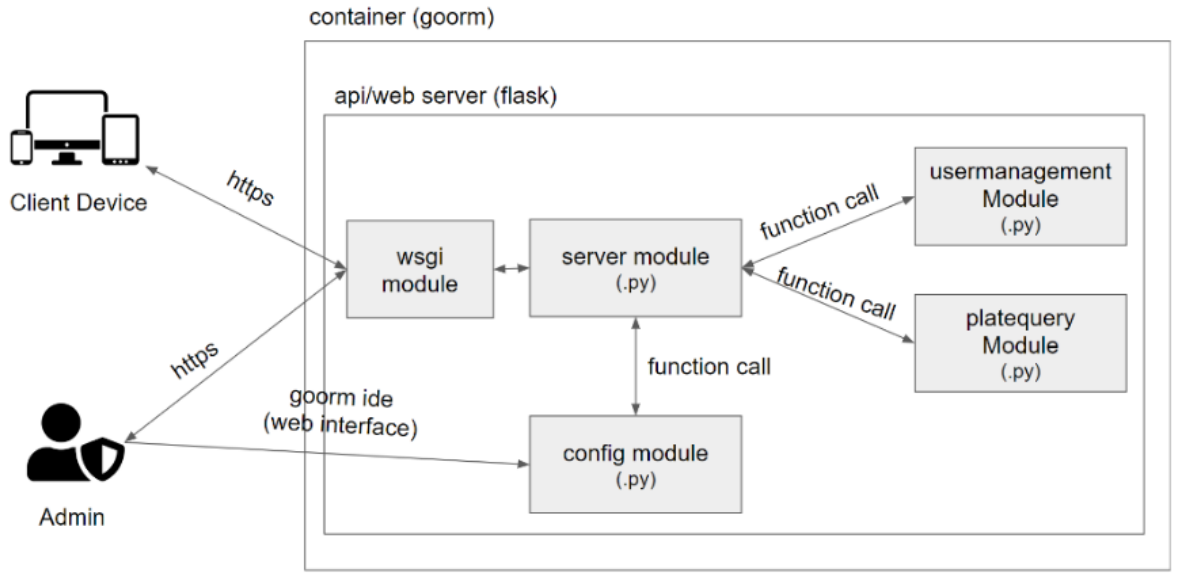


11.2 Client

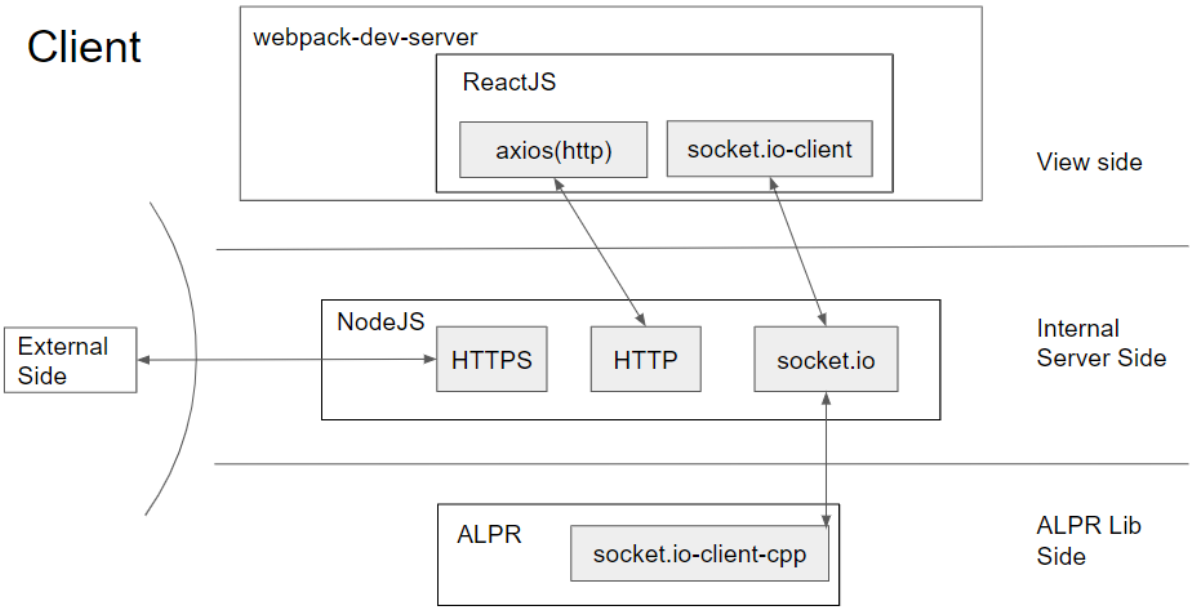


12. SW implementation

12.1 Server SW structure



12.2 Client SW structure



13. SW vulnerability evaluation

13.1 open source scanning

OWASP Dependency Check result is below.

Project: ALPR Project

Scan Information (show less):

- *dependency-check version*: 7.1.1
- *Report Generated On*: Tue, 5 Jul 2022 12:27:13 +0900
- *Dependencies Scanned*: 40 (39 unique)
- *Vulnerable Dependencies*: 0
- *Vulnerabilities Found*: 0
- *Vulnerabilities Suppressed*: 0
- *NVD CVE Checked*: 2022-07-05T12:27:06
- *NVD CVE Modified*: 2022-07-05T09:00:01
- *VersionCheckOn*: 2022-06-16T08:22:01

Analysis Exceptions

Unable to read yarn audit output.

Summary

Display: Showing All Dependencies (click to show less)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
App.js				0		0
App.test.js				0		0
Newtonsoft.Json:8.0.3		pkg:nuget/Newtonsoft.Json@8.0.3		0		4
OpenCVDeviceEnumerator.exe				0		2
TimeMem-1.0.exe	cpe:2.3:a:time_project:time:1.0:*:*:*:*:*:*:*			0	Low	4
alpr.exe				0		2

country_info.js				0		0
dashboard-layout.js				0		0
display_config.js				0		0
illegal_plate.js				0		0
index.js				0		0
index.js				0		0
input_source.js				0		0
libdb181.dll				0		4
liblept-DLL.dll				0		2
login.js				0		0
main.js				0		0
notfound.js				0		0
openalpr-opencv4.zip: AlprNet.csproj				0		2
openalpr-opencv4.zip: AlprNetGuiTest.csproj				0		2
openalpr-opencv4.zip: AlprNetTest.csproj				0		2
openalpr-opencv4.zip: packages.config				0		0
opencv_videoio_ffmpeg455_64.dll				0		2

opencv_videoio_msmf455_64.dll				0		2
opencv_videoio_msmf455_64d.dll				0		2
opencv_world455.dll	cpe:2.3:a:opencv:opencv:455:*:*:*:*:*:*:*			0	High	4
opencv_world455d.zip: opencv_world455d.dll	cpe:2.3:a:opencv:opencv:455:*:*:*:*:*:*:*			0	High	4
package.json				0		0
package.json				0		0
plate_image.js				0		0
plate_list.js				0		0
qrious.min.js				0		0
save_option.js				0		0
server.js				0		0
setupProxy.js				0		0
severity-pill.js				0		0
ssl-config.js				0		0
video_view.js				0		0
yarn.lock				0		2

13.2 secure coding rule

FlawFinder result is below.

ANALYSIS SUMMARY:

```
Hits = 1922
Lines analyzed = 795750 in approximately 6.74 seconds (118107 lines/second)
```

```
Physical Source Lines of Code (SLOC) = 508562
Hits@level = [0] 2024 [1] 600 [2] 987 [3]  47 [4] 287 [5]   1
Hits@level+ = [0+] 3946 [1+] 1922 [2+] 1322 [3+] 335 [4+] 288 [5+]   1
Hits/KSLOC@level+ = [0+] 7.75913 [1+] 3.77928 [2+] 2.59949 [3+] 0.65872 [4+]
0.566303 [5+] 0.00196633
Minimum risk level = 1
```

Team3_purple_ALPR_Phase.1_FlawFinder_Result.pdf

14. Test

14.1 Test methodology

We decided to perform a requirement-based test.
Test cases were created for requirements that are system level, software level, and whose priority is not optional.

14.2 Test Result & Bugs

1st Run

- test case pass % : 77% (14/18)
- test case NT(Not Tested) % : 10% (2/20)
- bug : 4

2nd Run

- test case pass % : 100% (18/18)
- test case NT(Not Tested) % : 10% (2/20)
- bug : 0

NT(Not Tested) means
that a case in which a test case for a requirement has been created, but is not currently testable due to the reasons of the verification environment.

You can find the details in the “Requirement based Test” and “Test Bugs” in the “Team3_purple_ALPR_Phase.1_Test.pdf” document in the deliverables.

15. Deliverables

15.1 Source code

Team3_purple_ALPR_Phase.1_source code

15.2 Requirements

Team3_purple_ALPR_Phase.1_Requirements.pdf

15.3 Threat modeling

Team3_Purple_ALPR_Phase.1_ThreatModeling.pdf

15.4 Risk rating

Team3_purple_ALPR_Phase.1_RiskRating.pdf

15.5 Secure coding rule result

Team3_purple_ALPR_Phase.1_FlawFinder_Result.pdf

15.6 Test cases/Test Bug

Team3_purple_ALPR_Phase.1_Test.pdf

15.7 Seeded vulnerability list

Team3_purple_ALPR_Phase.1_Seeded_Vulnerability_list.pdf

16. Vulnerability Reporting

If you find any vulnerabilities or have questions about our ALPR system, please email the Purple Team.

email : kibongs.song@gmail.com