

# **Automatic License Plate Recognition (ALPR) System Assessment for Team2**

## **Team3 PURPLE**

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

# Table of Contents

<b>1. Plan</b>	<b>4</b>
<b>1.1 Roles &amp; Responsibilities</b>	<b>4</b>
<b>1.2 Communication Ground Rules</b>	<b>4</b>
<b>1.3 Communication System and tools</b>	<b>4</b>
<b>1.4 Timeline</b>	<b>4</b>
<b>1.5 Target discovery and reconnaissance</b>	<b>5</b>
<b>1.6 Tools &amp; Techniques</b>	<b>5</b>
<b>1.7 Effort Allocation</b>	<b>5</b>
<b>2. Assessment</b>	<b>6</b>
<b>2.1 Executive Summary</b>	<b>6</b>
<b>2.2 Executive Constraint</b>	<b>7</b>
<b>2.3 Evaluation Narrative</b>	<b>7</b>
<b>2.4 Evaluation Result</b>	<b>13</b>
<b>3. Retrospect</b>	<b>15</b>
<b>4. Deliverables</b>	<b>16</b>
<b>4.1 Found vulnerability list</b>	<b>16</b>

Version	Date	Status	Author
1.0	2022.07.14	1st release	team3

# 1. Plan

## 1.1 Roles & Responsibilities

- Team Lead
  - Kibong Song (송기봉)
- Vulnerability Assessment and Documentation
  - Youngmi Choi (최영미), Seungkyu Lee (이승규)
- Reverse Engineering
  - Seongsik Kim (김성식), Haenggi Lee (이행기)
- Penetration Testing
  - Kibong Song (송기봉), Gyunggi Moon (문경귀)
- Mentor
  - David Belasco (데이비드)

## 1.2 Communication Ground Rules

1. All team members will treat each other with respect and will always provide accurate and truthful information.
2. Meetings should start and end on time, and an agenda should be prepared in advance and distributed to all participants; stay on topic.
3. Mentor and team members share and discuss the progress through online video meetings once a week.

## 1.3 Communication System and tools

1. Team3 will use the google drive to share all documents with other team members at the same time.
2. Team members communicate with mentors in real time using slack.

## 1.4 Timeline

The expected timeline and deliverables of the assessment of team2 will be achieved over a period of approximately two weeks.

1. week1 (From July 4th to July 8th)
  - a. review all artifacts of team2
  - b. establish the vulnerability assessment plan
2. week2 (From July 11th to July 14th)
  - a. start looking for vulnerabilities
  - b. prepare the assessment report for found vulnerability

## 1.5 Target discovery and reconnaissance

Our target is ALPR system of team2.

We collect and review publicly available information about the target. Such information can be found in Team2's github(<https://github.com/hyecahn/2022-security-specialist-team2>).

## 1.6 Tools & Techniques

1. reverse engineering techniques
  - a. Runtime analysis plan to perform using the tool to monitor the network activity between the server and the client.
    - i. Wireshark
  - b. Runtime analysis plan to perform using the tool to analyze a encrypted DB
    - i. db\_dump
    - ii. Hex View
  - c. Since the source code of the other team has been provided, static analysis will be performed through code review.
    - i. We plan to focus on logic error, Insecure configuration, cryptographic vulnerability and Input validation error
2. penetration testing techniques
  - a. We plan to exploit the client app uses popular open source software
    - i. OWASP ZAP: target to Django
    - ii. Metasploitable in Kali: target to opencv-python
  - b. We plan to use input validation for various inputs.
3. Research
  - a. We plan to research some specific DB since the client app uses a specific version of open source software.
    - i. <https://www.exploit-db.com/>
    - ii. <https://www.cvedetails.com/>
4. Fuzz testing
  - a. We plan to use fuzzing tools to mutate the sample JPEG images of license plates
    - i. Dumb fuzzing with ZZUF
    - ii. JPEG fuzzer <https://github.com/h0mbre/Fuzzing>
5. vulnerability evaluation
  - a. We select CVSS as the vulnerability evaluation model.

## 1.7 Effort Allocation

In order to produce meaningful results within limited resources, it was decided to analyze the areas with high threat risk first.

From the attacker's point of view, the security goal and threat modeling results provided by team2 were first reviewed.

Based on the risk assessment provided by Team 2, efforts for critical/high-grade requirements were first determined to analyze the security flaws of high-risk requirements.

SR-01, critical : The system shall allow an officer to access the ALPR system through a secure web interface by two factor authentication.

- secure web interface (https, certification by CA)
- two factor authentication

SR-03, high : The system shall grant the admin user to access and modify configuration files

- access and modify the configuration file

SR-04, high : The Plate DB must be encrypted data

- encrypted plates DB

## 2. Assessment

### 2.1 Executive Summary

Assessment target : ALPR system implemented by team2.

The ALPR system was analyzed using techniques and tools discussed in the planning stage, and eight security vulnerabilities were found.

Priority	Vulnerability	Impact	Recommendation
High (CVSS 8.7)	(VID-1) If we can set request.session.error, then we have access to add a new user in the same privilege as super user.	An attacker can compromise user credential information.	Fix improper error handling; Apply least privileged access control
High (CVSS 7.5)	(VID-4) The key value used for DB encryption in the Lookup server is the ASCII string '2Team_AhnLab'; it's hard-coded and predictable.	An attacker can disclose personal identifiable information.	Remove the hard-coded key value; Change to a stronger one and store it in a secure place
High (CVSS 7.9)	(VID-8) 0 can be entered as the maximum user setting value in 'server.conf'. This causes the server to either return an internal error or hang.	An attacker can tamper configuration information to stop the service.	Check setting input value whether it is within the normal range before use
Medium (CVSS 6.5)	(VID-2) If the user id is '..' then a video or image file cannot be uploaded.	Path traversal allows unauthorized users to access files.	Recommend that you use user ID as user folder path.
High (CVSS 7.3)	(VID-3) If you rename the sample video file to a long name and upload it, an internal error occurs the moment the license plate is recognized.	An attacker can trigger malfunction of various components by providing malformed data.	recommend limiting the file length in the upload function.
High (CVSS 7.6)	(VID-5) If you upload an image file with comments as 'Ahnlab', it gives a	An attacker can trigger malfunction of various components by providing malformed	Remove unnecessary code through code review

	shutdown command to the server.	data.	
High (CVSS 7.6)	(VID-6) It is possible to delete files uploaded by another user.	Uploaded contents can be deleted by anyone who logged in the system.	Apply access control so that users can only control their own data
Medium (CVSS 5.8)	(VID-7) Using the user's session id, you can login without password or 2 factor authentication.	An attacker can use another user's session ID to log in and use the system without a separate authentication process.	Set session validity period for user authentication In case of django, the session validity period can be set through the COOKIE_AGE value (default : 2 weeks)

The found vulnerability was able to damage 9 of the 10 assets identified by team2.

By exploiting the vulnerability of VID-01, The attacker can use the system like a regular user by adding multiple unauthorized IDs and passwords. In this way, the attacker can modulate the information(ID, Password) in the user DB.

- AID-06 : ID
- AID-07 : Password

By exploiting the vulnerability of VID-04, all information of the plate DB has been discovered and DB tampering is possible, so damage can be applied to five assets identified by team2.

- AID-01 : License Plate Number
- AID-02 : Status
- AID-03 : Owner Name
- AID-04 : Owner Date of Birth
- AID-05 : Owner Street Address/ Location
- AID-10 :Owner City, State and Zip Code

By exploiting the vulnerability of VID-08, It is possible to stop the system by changing the configuration value that affects the system operation.

- AID-08 : Configuration File

In conclusion, we confirmed the results of compromising the declared security goal.

## 2.2 Executive Constraint

- validation of Certificates is excluded from the analysis.
- The analysis of the simultaneous connection of multi-user was excluded as an issue of the executable environment.

## 2.3 Evaluation Narrative

The critical/high risks of the threats derived at the time of development were analyzed first.

### 1. secure web interface between client and server

- a. runtime analysis
  - i. techniques & tool : wireshark
  - ii. rationale : Commonly used wireshark is used to verify TLS application and packet verification between networks

- iii. activities :
  - 1. Capture packets with wireshark while running server and client
  - 2. Check client and server communication packets
- iv. result : TLS was applied.
- v. found vulnerability ID : n/a

## 2. two factor authentication

- a. runtime analysis
  - i. techniques & tool : wireshark
  - ii. rationale : Commonly used wireshark is used to verify TLS application and packet verification between networks
  - iii. activities :
    - 1. Capture packets with wireshark while running server and client
    - 2. Check client and server communication packets
  - iv. result : User ID/Password exposed in login packet transmission (POST) between web interface and client. But, we can't prove how to check other users' network packets, so we excluded it from the vulnerability.

Id	Time	Source	Destination	Protocol	Length	Info
535	210.965..	127.0.0.1	127.0.0.1	TCP	56	[TCP Keep-Alive ACK] 8000 → 53579 [ACK] Seq=72574 Ack=3332 Win=102
536	210.198..	10.58.58.128	10.58.58.128	TCP	44	[TCP ACK] 8000 → 53579 [ACK] Seq=72574 Ack=497 Win=2619136 Len=0
537	210.198..	10.58.58.128	10.58.58.128	TLS	144	[TLS Handshake] 8000 → 53579 [ACK] Seq=72574 Ack=497 Ack=655 Win=2619549
538	210.198..	10.58.58.128	10.58.58.128	TLSv1.3	1775	[TLS segment of a reassembled PDU]
539	210.198..	10.58.58.128	10.58.58.128	TCP	44	8000 → 53882 [ACK] Seq=7652 Ack=13056 Win=2607104 Len=0
540	210.198..	10.58.58.128	10.58.58.128	TCP	44	8000 → 53885 [ACK] Seq=497 Ack=666 Win=2619648 Len=0
541	210.200..	10.58.58.128	10.58.58.128	TCP	44	8000 → 53885 [INST, ACK] Seq=497 Ack=666 Win=0 Len=0
542	210.204..	10.58.58.128	10.58.58.128	HTTP	190	POST /login/login/ HTTP/1.1 (application/x-www-form-urlencoded)
543	210.204..	10.58.58.128	10.58.58.128	TCP	44	8000 → 53882 [ACK] Seq=7652 Ack=13202 Win=2607104 Len=0
544	210.491..	10.58.58.128	10.58.58.128	HTTP	873	HTTP/1.1 302 Found
545	210.491..	10.58.58.128	10.58.58.128	TCP	44	53882 → 8000 [ACK] Seq=13202 Ack=8481 Win=2611200 Len=0
546	210.498..	10.58.58.128	10.58.58.128	HTTP	1639	HTTP/1.1 /alipay/ HTTP/1.1
547	210.701..	10.58.58.128	10.58.58.128	TCP	44	8000 → 53882 [ACK] Seq=6481 Ack=14817 Win=2605312 Len=0
548	210.701..	10.58.58.128	10.58.58.128	TLSv1.3	203	[TLS segment of a reassembled PDU]
549	210.701..	10.58.58.128	10.58.58.128	TCP	44	53882 → 8000 [ACK] Seq=14817 Ack=8640 Win=2610944 Len=0
550	210.701..	10.58.58.128	10.58.58.128	TLSv1.3	440	[TLS segment of a reassembled PDU]
551	210.701..	10.58.58.128	10.58.58.128	TCP	44	53882 → 8000 [ACK] Seq=14817 Ack=9036 Win=2610688 Len=0
552	210.701..	10.58.58.128	10.58.58.128	TCP	8236	8000 → 53882 [PSH, ACK] Seq=9036 Ack=14817 Win=2605312 Len=8192 [T]
553	210.701..	10.58.58.128	10.58.58.128	TCP	44	53882 → 8000 [ACK] Seq=14817 Ack=17228 Win=2602496 Len=0
554	210.701..	10.58.58.128	10.58.58.128	TCP	8236	8000 → 53882 [PSH, ACK] Seq=17228 Ack=14817 Win=2605312 Len=8192 [T]
555	210.701..	10.58.58.128	10.58.58.128	TCP	44	53882 → 8000 [ACK] Seq=14817 Ack=25420 Win=2594304 Len=0
556	210.701..	10.58.58.128	10.58.58.128	TLSv1.3	719	HTTP/1.1 200 OK [text/html]

[HTTP request 8/10]  
 [Prev request in frame: 474]  
 [Response in frame: 544]  
 [Next request in frame: 546]  
 File Date: 2023-06-20  
 Form URL Encoded: application/x-www-form-urlencoded  
 Form item: "csrftoken" = "f8YgZlw90SoE1Gr8poek84waSMKuduFXABNsIMYboS3HzFlqGlyOiaeF5sY50H"  
 Form item: "username" = "kibongs.song"  
 Key: username  
 Value: kibongs.song  
 Form item: "password" = "1ge12345"  
 Key: password  
 Value: 1ge12345  
 Community ID: 1:k4Ftp/cCjboVyy0t4hiyInSq5Rfo=

- v. found vulnerability ID : n/a
- b. runtime analysis
  - i. techniques & tool : cookies
  - ii. rationale : Authenticates users through sessions. Therefore, to check the cookie information to see if the mitigation for the weak point of the session is applied.
  - iii. activities :
    - 1. multi user login
    - 2. Check session id value for each user through developer tools in chrome
    - 3. Enter other user session id values in cookies to access the webpage
  - iv. result : Can use another user's session ID to log in and use the system without a separate authentication process
  - v. found vulnerability ID : VID-7

## 3. access and modify the configuration file

- a. static analysis
  - i. techniques & tool : Code Review
  - ii. rationale :
    - 1. Use CodeReview to check input validation
  - iii. activities :
    - 1. Conduct a code review on the part of the server source code that handles the change in the value of the configuration file "server.conf".

2. Verify that a value is assigned to a variable without validation of the value entered by the user
  3. Check if UI processes invalid input
  4. Check the range of inputs in the UI, but make sure that not all invalids are checked
  5. Click the update button without input values(MaxUser, Confidence Level)
  6. An internal server error(500) occurs
- iv. result :
1. the MaxUser value is set to 0 in “server.conf” file and
  2. The ALPR system will not work until the server operator notices this and changes it to the correct value.
- v. found vulnerability : VID-8
- b. runtime analysis
- i. techniques & tool : DB Tool (DB Browser for sql)
  - ii. rationale : As a way to classify admin accounts in Phase.1, a flag for classifying admins is added to the user db table. Assuming that other teams have done the same, searched the db table.
  - iii. activities :
    1. Check whether user DB file password is applied
    2. Check whether user DB data encryption is applied
  - iv. result : User DB file was not encrypted, and Admin flag among user db data is not encrypted. Can acquire Admin privileges by changing the Admin flag. but, we can't prove how to access the DB, so we excluded it from the vulnerability
  - v. found vulnerability ID : n/a

#### 4. encrypted plates DB

- a. static analysis
- i. techniques & tool : code review, IDA, db\_dump, Hex View
  - ii. rationale :
    1. Use IDA to determine the key value used for encryption
    2. Use db\_dump, Hex View for DB lookup and data parsing
  - iii. activities :
    1. Open licenseplate.db to verify that it is encrypted
    2. Decompile Server.exe to IDA to find out the key
    3. Found out key is "2Team An Lab"

```

86     else if ( (*unsigned int (__fastcall **)(__int64, int *))(v57 + 824))(v57, v66)
87         || (v66[0] = 1, (*unsigned int (__fastcall **)(__int64, const char *, __int64))(v57,
88             v57,
89             code,
90             1164)) )
91     {
92         printf("DB Encrypt Error\n");
93         result = 0xFFFFFFFF164;
94     }

```

```

.rdata:000000014000DB30 ; const char code[12]
.rdata:000000014000DB30 code db '2', 'T', 'e', 'a', 'm', '.', 'A', 'h', 'n', 'L', 'a'
.rdata:000000014000DB30                                         ; DATA XREF: ProcessClient(Void *)+B9 to
.rdata:000000014000DB30 db 'b'
.rdata:000000014000DB3C align 20h

```

4. After db\_dump (decrypt with key value input) for DB decoding and query, plate DB information is found with HexView

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00004420 6F 72 65 72 20 53 70 6F 72 74 0A 74 65 61 6C 00 orer Sport.teal.
00004430 47 58 56 32 39 34 31 00 47 58 56 32 39 34 31 0A GXV2941.GXV2941.
00004440 4F 77 6E 65 72 20 57 61 6E 74 65 64 0A 30 31 2F Owner Wanted.01/
00004450 30 36 2F 32 30 32 33 0A 43 68 72 69 73 74 6F 70 06/2023.Christop
00004460 68 65 72 20 46 69 73 68 65 72 0A 30 36 2F 31 31 her Fisher.06/11
00004470 2F 31 39 38 31 0A 39 30 32 39 20 47 6C 6F 72 69 /1981.9029
00004480 61 20 53 74 72 61 76 65 6E 75 65 0A 45 61 73 74 a Stravenue.East
00004490 20 41 6E 64 72 65 77 2C 20 4D 44 20 36 37 33 35 Andrew, MD 6735
000044A0 39 0A 32 30 30 33 0A 43 61 64 69 6C 61 63 0A 9.2003.Cadillac.
000044B0 45 6E 63 6F 72 65 0A 62 6C 61 63 6B 00 4B 36 36 Encore.black.K66
000044C0 33 31 39 4B 00 4B 36 36 33 31 39 4B 0A 4F 77 6E 319K.K66319K.Own
000044D0 65 72 20 57 61 6E 74 65 64 0A 30 34 2F 32 38 2F er Wanted.04/28/

```

- iv. result : All personal information in the plate DB was leaked by decrypting the encrypted DB.
- v. **found vulnerability ID : VID-4**

After that, the part randomly selected by our team was analyzed.

### 1. plate image/video recognition and processing

- a. static analysis
  - i. techniques & tool : Code Review
  - ii. rationale : Use CodeReview to identify vulnerabilities in ALPR system code
  - iii. activities :
    - 1. Make a small image with width and height less than or equal to 1
    - 2. Upload it
    - 3. Then we have access to add a new user in the same privilege as admin
  - iv. result :
    - 1. The 'Create' button is displayed when uploading an image
  - v. **found vulnerability ID : VID-1**
- b. static analysis
  - i. techniques & tool : Code Review
  - ii. rationale : Use CodeReview to identify vulnerabilities in client code
  - iii. activities :
    - 1. Make a small image with width and height less than or equal to 1
    - 2. Upload it
    - 3. Then we have access to add a new user in the same privilege as admin
    - 4. Add a new user - ID is '..'
    - 5. Try to upload a video or image file
    - 6. Get an error from Django framework
  - iv. result : If the user id is '..' then a video or image file cannot be uploaded
  - v. **found vulnerability ID : VID-2**
- c. static analysis
  - i. techniques & tool : Code Review
  - ii. rationale : Use CodeReview to identify vulnerabilities in ALPR system code
  - iii. activities :
    - 1. Check if there is any code that can force the lookup server to shutdown on server-side
    - 2. Check if there is any code that sends the 'shutdown' command on the client-side
    - 3. Send shutdown message when the comment of image is the same as app\_name(in this case: Ahnlab)
  - iv. result :
    - 1. Add comment to image using 'imagemagick' library

2. The lookup server is shutdown when uploading the image created in step 1

#### v. found vulnerability ID : VID-5

d. static analysis



5. Play the file and then, happen certain exceptions & stop when to recognize the first plate. The system tries to store isolated plate image. Total path is very long so, It is crashed.

```
update error: Storage can not find an available filename for "media/gord36k/aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa_3aPWILOW261_iWdL937.jpg". Please make sure that the corresponding file field allows sufficient 'max_length'.
video released
destroy.....
end of update
close connection
video released
destroy.....
video released
destroy.....
```

- iv. result : There is no input validation about upload file type and length.
  - v. **found vulnerability ID : VID-3**

v. found vulnerability ID : VID-3

e. Fuzz testing

- i. techniques & tool : zzuf, JPEG fuzzer
  - ii. rationale : Users can input various images into the OpenALPR library. Professor Jeff recommended the application of fuzzing when evaluating projects.
  - iii. activities :
    - 1. Dumb fuzzing: Mutate a sample license plate image(us-1.jpg) using zzuf and upload it.

```

[14/Jul/2022 13:15:32] "GET /alpr/remove_vehicle_history?filename=us-1_dumb_fuzz.jpg" [IP/1.1" 302 0
Corrupt JPEG data: bad Huffman code
close connection
Traceback (most recent call last):
  File "C:\WS\CMU\Project\Git_repo\2022-security-specialist-team2\client\studio\webapp\alpr\views.py", line 142, in get_frame
    _, jpeg = cv2.imencode('.jpg', image)
cv2.error: OpenCV(4.6.0) D:\a\opencv-python\opencv-python\opencv\modules\imgcodecs\src\loadsave.cpp:976: error: (-215:Assertion failed) !image.empty() in function 'cv::imencode'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.10_3.10.1520.0_x64_qbz5n2kfra8p0\lib\wsgiref\handlers.py", line 188, in run
    self.finish_response()
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.10_3.10.1520.0_x64_qbz5n2kfra8p0\lib\wsgiref\handlers.py", line 188, in finish_response
    for data in self.result:
  File "C:\WS\CMU\Project\Git_repo\2022-security-specialist-team2\lib\site-packages\django\utils\text.py", line 396, in compress_sequence
    for item in sequence:
  File "C:\WS\CMU\Project\Git_repo\2022-security-specialist-team2\client\studio\webapp\alpr\views.py", line 434, in get_frame
    _, frame = stream.get_frame()
  File "C:\WS\CMU\Project\Git_repo\2022-security-specialist-team2\client\studio\webapp\alpr\views.py", line 149, in get_frame
    self.conn.close()
AttributeError: 'VideoStream' object has no attribute 'conn'
Exception in thread Thread-721 (update):
Traceback (most recent call last):
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.10_3.10.1520.0_x64_qbz5n2kfra8p0\lib\threading.py", line 1016, in _bootstrap_inner
    self.run()
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.10_3.10.1520.0_x64_qbz5n2kfra8p0\lib\threading.py", line 953, in run
    self._target(*self._args, **self._kwargs)
  File "C:\WS\CMU\Project\Git_repo\2022-security-specialist-team2\client\studio\webapp\alpr\views.py", line 320, in update
    height, w, c = self.frame.shape
AttributeError: 'NoneType' object has no attribute 'shape'
Exception ignored in: <function VideoStream.__del__ at 0x00000242F6FC0820>
Traceback (most recent call last):
  File "C:\WS\CMU\Project\Git_repo\2022-security-specialist-team2\client\studio\webapp\alpr\views.py", line 133, in __del__
    if self.video:
AttributeError: 'VideoStream' object has no attribute 'video'

```

2. JPEG fuzzing: Mutate a sample license plate image(us-1.jpg) using a tool which subtly alters valid JPEGs and upload it  
**JPEG fuzzer:** <https://github.com/h0mbre/Fuzzing>

```

[14/Jul/2022 13:17:55] "GET /alpr/remove_vehicle_history?filename=us-1.jpeg_fuzz.jpg" [IP/1.1" 302 0
close connection
Exception in thread Thread-735 (update):
Traceback (most recent call last):
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.10_3.10.1520.0_x64_qbz5n2kfra8p0\lib\threading.py", line 1016, in _bootstrap_inner
    self.run()
Traceback (most recent call last):
  File "C:\WS\CMU\Project\Git_repo\2022-security-specialist-team2\client\studio\webapp\alpr\views.py", line 142, in get_frame
    _, jpeg = cv2.imencode('.jpg', image)
cv2.error: OpenCV(4.6.0) D:\a\opencv-python\opencv-python\opencv\modules\imgcodecs\src\loadsave.cpp:976: error: (-215:Assertion failed) !image.empty() in function 'cv::imencode'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.10_3.10.1520.0_x64_qbz5n2kfra8p0\lib\wsgiref\handlers.py", line 188, in run
    self.finish_response()
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.10_3.10.1520.0_x64_qbz5n2kfra8p0\lib\wsgiref\handlers.py", line 188, in finish_response
    for data in self.result:
  File "C:\WS\CMU\Project\Git_repo\2022-security-specialist-team2\lib\site-packages\django\utils\text.py", line 396, in compress_sequence
    for item in sequence:
  File "C:\WS\CMU\Project\Git_repo\2022-security-specialist-team2\client\studio\webapp\alpr\views.py", line 434, in get_frame
    _, frame = stream.get_frame()
  File "C:\WS\CMU\Project\Git_repo\2022-security-specialist-team2\client\studio\webapp\alpr\views.py", line 149, in get_frame
    self.conn.close()
  File "C:\WS\CMU\Project\Git_repo\2022-security-specialist-team2\client\studio\webapp\alpr\views.py", line 320, in update
    height, w, c = self.frame.shape
AttributeError: 'VideoStream' object has no attribute 'conn'
AttributeError: 'NoneType' object has no attribute 'shape'
Exception ignored in: <function VideoStream.__del__ at 0x00000242F6FC0820>
Traceback (most recent call last):
  File "C:\WS\CMU\Project\Git_repo\2022-security-specialist-team2\client\studio\webapp\alpr\views.py", line 133, in __del__
    if self.video:
AttributeError: 'VideoStream' object has no attribute 'video'

```

- 3.

- iv. result : Just caught in opencv library assertions upon both fuzzing  
v. found vulnerability ID : n/a

## 2. Django

### a. runtime analysis

- i. techniques & tool : General browsers
    - ii. rationale : for sending GET request
    - iii. activities : file delete GET requests were made from result of ZAP active scan, then send them through the browser. e.g. <https://ahnlab2.lge.com:8000/alpr/remove?id=xx>
    - iv. result : Contents uploaded by another user could be deleted. If deletion failed due to absence of target, Server error (500) returned.
    - v. found vulnerability ID : VID-6
  - b. vulnerability scanning
    - i. techniques & tool : OWASP ZAP
    - ii. rationale : ZAP provides “spider scan” as crawling and “active scan” which tries to find out known vulnerabilities.
    - iii. activities : run “active scan” for ahnlab2.lge.com:8080
    - iv. result : Can get directory list provided by server
    - v. found vulnerability ID : n/a
  - c. research Exploit DB and Vulnerability DB
    - i. techniques & tool : research
    - ii. rationale : The client app uses a specific version of open source software.
    - iii. activities :
      - 1. <https://www.exploit-db.com/> : target to Django 4.0.5 and opencv-python 4.6.0.66
      - 2. <https://www.cvedetails.com/> : target to Django 4.0.5 and opencv-python 4.6.0.66
    - iv. result : Known vulnerabilities are already patched to Django 4.0.5 and opencv-python 4.6.0.66 E(they are latest)
    - v. found vulnerability ID : n/a
3. opencv-python
- a. penetration testing
    - i. techniques & tool : Metasploitable in Kali
    - ii. rationale : The client app uses popular open source software.
    - iii. activities : Search the metasploit database for an exploit in opencv-python
    - iv. result : No search result returned.
    - v. found vulnerability ID : n/a

## 2.4 Evaluation Result

In conclusion, We found eight vulnerabilities and evaluated them with CVSS scoring methodology.

We used a tool that automatically calculates scores.  
<https://itschool-info-lab.github.io/cvss/>

## CVSS v3.1 Base Score Calculator

ATTACK VECTOR (공격 벡터)	ATTACK COMPLEXITY (공격의 복잡성)	PRIVILEGES REQUIRED (필요한 권한)	USER INTERACTION (사용자 참여 정도)
 Network	 Low	 None	 None
 Adjacent	 High	 Low	 Required
 Local		 High	
 Physical			
SCOPE (공격 범위)	CONFIDENTIALITY (기밀성)	INTEGRITY (무결성)	AVAILABILITY (가용성)
 Changed	 High	 High	 High
 Unchanged	 Low	 Low	 Low
	 None	 None	 None
SEVERITY-SCORE-VECTOR			
CVSS:3.1/AV:_/AC:_/PR:_/UI:_/S:_/C:_/I:_/A:_			

You can find the found vulnerabilities in detail on the  
[“Team3\\_purple\\_ALPR\\_Phase.2\\_Assessment\\_for\\_Team2\\_Found\\_Vulnerability\\_list.pdf”](#).

### 3. Retrospect

While developing the project in phase 1, We tried to apply what I learned in class to the project if possible, but when we started implementation, we were more focused on implementing the function. It was difficult to maintain a security mindset throughout the development lifecycle.

If I do this task again in the future, I would like to try injecting various vulnerabilities by further analyzing the attack surface.

In phase 2, we tried to analyze and approach them using the artifactory provided by other teams. However, under the pressure to discover security flaws in a short period of time, it was difficult to consider the various tools and methodologies learned in the class, and it seemed to focus only on reverse engineering with provided source codes.

Below is a short comment from the team3 members at the end of the CMU class.

- It is difficult to tell security relevant failures from other functional failures. (Haenggi, Brian)
- It requires more experience to apply the attack techniques we have practiced in the course. (Seungkyu, Kibong)
- Choosing secure OSS packages reduces the risks from unintentional vulnerabilities. (Seongsik, Haenggi)
- Threat modeling is not only a set of technical activities, but also an opportunity to collaborate to build more secure systems. (Youngmi, Brian)

## **4. Deliverables**

### **4.1 Found vulnerability list**

[Team3\\_purple\\_ALPR\\_Phase.2\\_Assessment\\_for\\_Team2\\_Found\\_Vulnerability\\_list.pdf](#)