Tab 1

# Setting up a Login DB Server

You could designate a PC for Login Server.

Environment settings

* Only run on **Windows PC**.
* Python **3.9+** is required

<https://www.python.org/downloads/>

* Install flask:

**pip install flask**

* Install sqlcipher3:

**pip install sqlcipher3-wheels**

Download Source Code

* git clone https://github.com/LGSDET/AimHigh\_DB.git

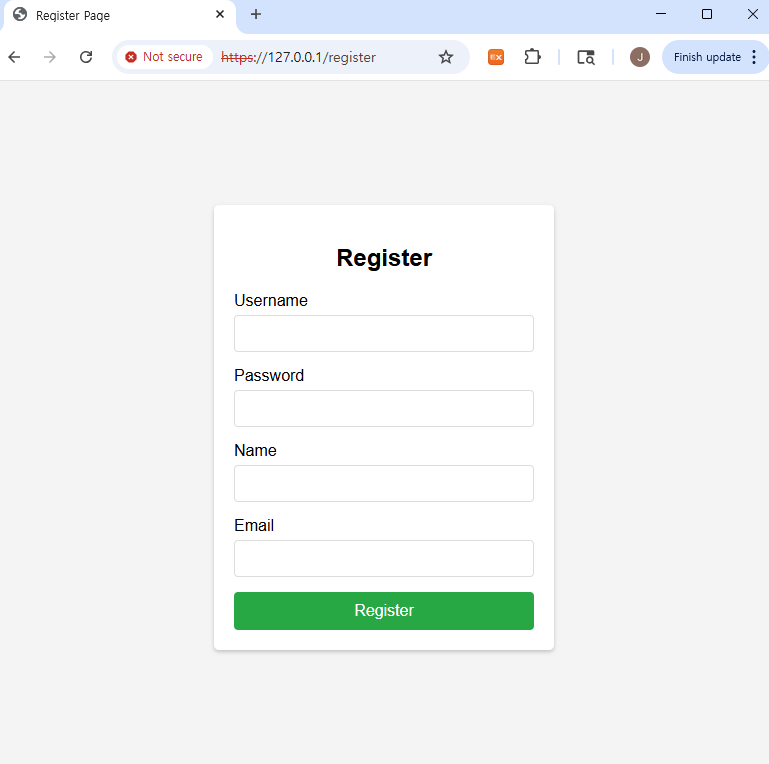
Run Server

* Go to the path for AimHigh\_DB
* Run [app.py](http://app.py)

| AimHigh\_DB> python .\app.py  ~\app.py:194: DeprecationWarning: ssl.PROTOCOL\_TLSv1\_2 is deprecated  context = ssl.SSLContext(ssl.PROTOCOL\_TLSv1\_2)  \* Serving Flask app 'app'  \* Debug mode: off  WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  \* Running on all addresses (0.0.0.0)  \* Running on https://127.0.0.1:443  \* Running on https://172.20.2.114:443  Press CTRL+C to quit |
| --- |

* Register a new user

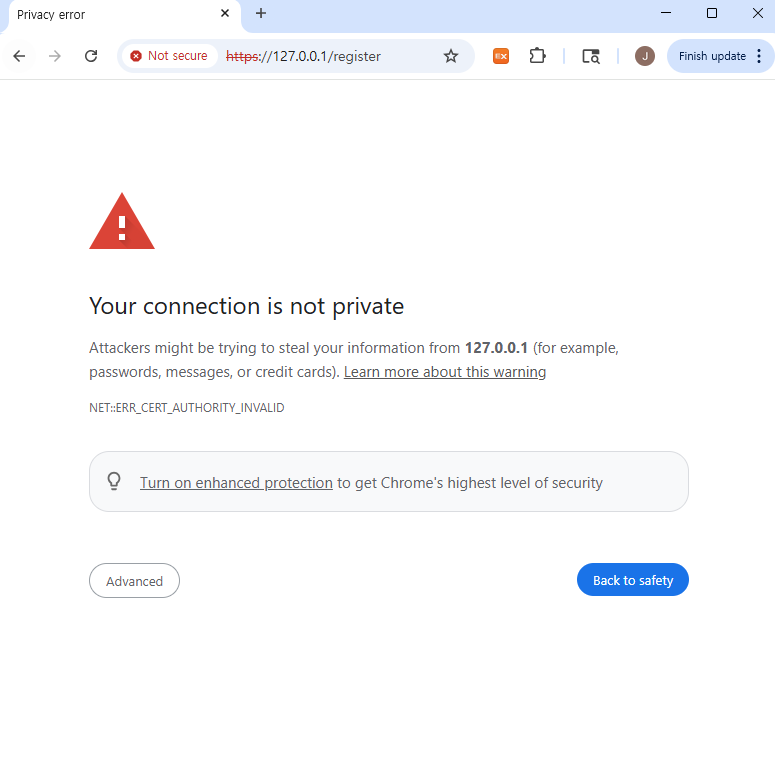
Go to <https://127.0.0.1/register>



Once you register your username, password and email address, you could log in our Flight Agent System.

Troubleshootings

* If you see the website below, it is normal because our cert is self-signed.



* If you fail “pip install flask” below, set virtual python environment.

| ex)  AimHigh\_DB> pip install flask  Collecting flask  Downloading flask-3.1.1-py3-none-any.whl (103 kB)  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━103.3/103.3 kB 56.7 kB/s eta 0:00:00  Collecting blinker>=1.9.0  Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)  Collecting markupsafe>=2.1.1  Downloading MarkupSafe-3.0.2-cp310-cp310-win\_amd64.whl (15 kB)  Collecting click>=8.1.3  Downloading click-8.2.1-py3-none-any.whl (102 kB)  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/102.2 kB ? eta -:--:--  ERROR: Exception:  Traceback (most recent call last):  File "C:\Users\Matthias Kim\AppData\Local\Programs\Python\Python310\lib\site-packages\pip\\_vendor\urllib3\response.py", line 438, in \_error\_catcher  yield  File "C:\Users\Matthias Kim\AppData\Local\Programs\Python\Python310\lib\site-packages\pip\\_vendor\urllib3\response.py", line 561, in read  data = self.\_fp\_read(amt) if not fp\_closed else b""  File "C:\Users\Matthias Kim\AppData\Local\Programs\Python\Python310\lib\site-packages\pip\\_vendor\urllib3\response.py", line 527, in \_fp\_read  return self.\_fp.read(amt) if amt is not None else self.\_fp.read()  File "C:\Users\Matthias Kim\AppData\Local\Programs\Python\Python310\lib\site-packages\pip\\_vendor\cachecontrol\filewrapper.py", line 90, in read  data = self.\_\_fp.read(amt) |
| --- |

1. **Create the virtual environment**

| AimHigh\_DB> python -m venv venv |
| --- |

1. **Activate the virtual environment**

| AimHigh\_DB> venv\Scripts\activate  (venv) C:\Your\Path\AimHigh\_DB> **pip install flask** |
| --- |

# Setting up a Remote Display App

Download Source Code

* git clone <https://github.com/LGSDET/AimHigh.git>

Build App

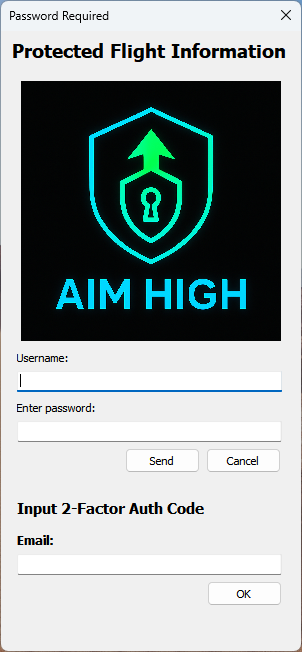
* Put the IPv4 address of login DB server on MFAuthentication.cpp as below.

| **ex)**  // POST  http->Post("https://127.0.0.1/auth", requestBody.get(), responseBody.get());  =>  http->Post("https://**127.20.10.8**/auth", requestBody.get(), responseBody.get()); |
| --- |

* Refer to the file: ***Remote User Interface Set Up.pdf*** from Canvas.

Run App

* If you run ADS-B-Display.exe successfully, you can only see the login window.



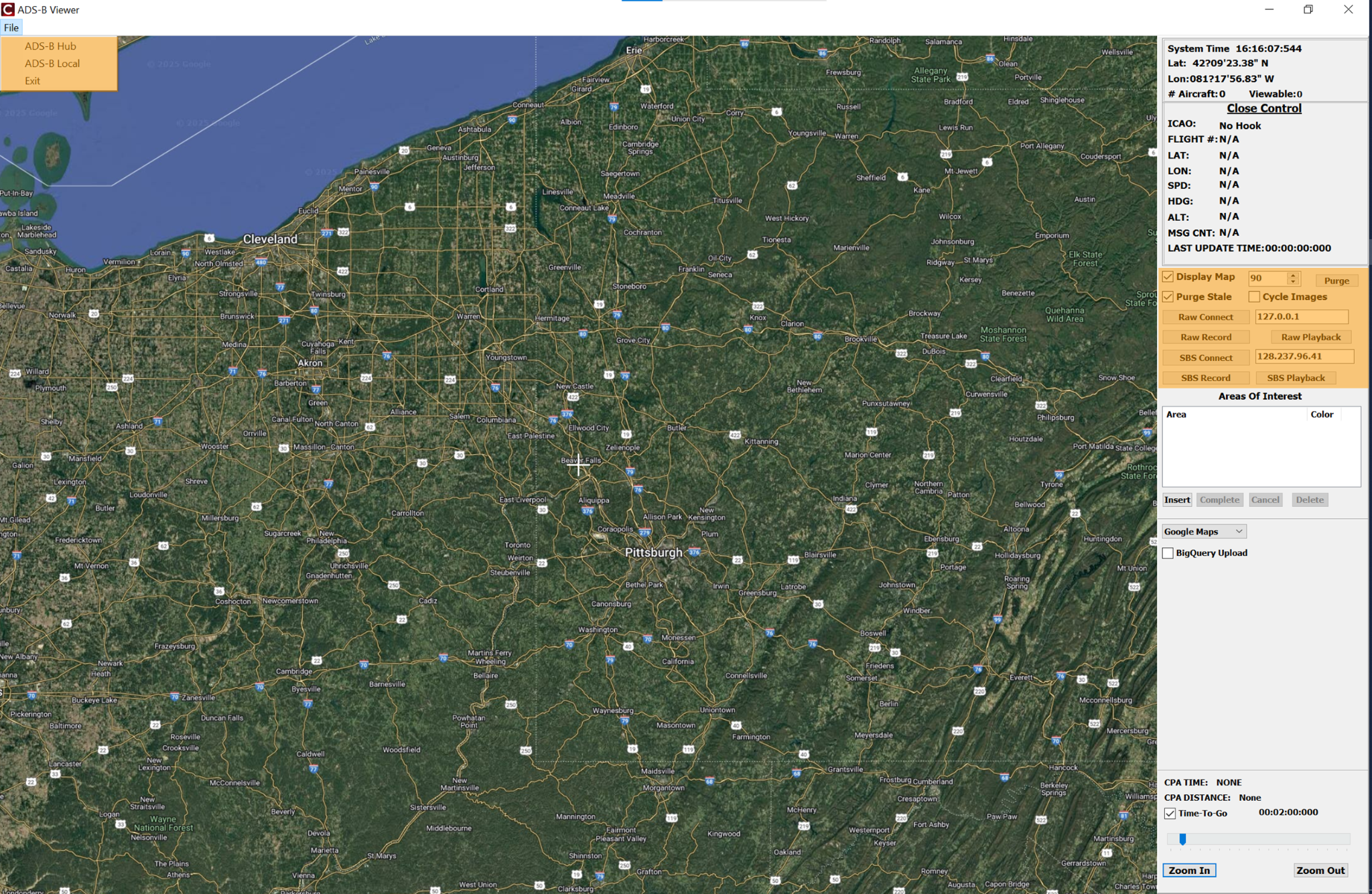
* If you want to use our flight Agent System, you must login first.
* Fill in the textbox for your username and password which you already registered.
* Click Send button.
* Check your email to find the passcode that our system sent, then put the code and click OK.
  + If you can't find the email, **please check your spam folder**.
* Once you login successfully first, you can see any flight data on our system.

# Secure Logger feature

Overview

* Secure Logger feature secures the log file content, preventing the log file from being modified.
* The log file will be encrypted line by line to make sure the line cannot be modified.
* The Log file’s checksum is calculated and saved checksum file.
* The User-specific encryption keys are saved on the Login Server

Log the ADS-B Display user’s behaviour



1. Press the buttons that you want to.
2. In the execution file’s folder, check the file in which format log-YYYYMMDD\_HHMMSS.enc

| 2025-06-04 16:22:48|INFO|405cf105b8994eb587a005de|ad3e72bae3a20ae3bddb5644ad7dc4bb|6b6426ced51fbaa0c0e055e1b281fdadd27b4a9e9ba7bcec4f0fa387b7d6b6a2a429b798b993c100436c78fc8166a8b68c|***SBSConnectButtonClick connect: 128.237.96.41:5002*** |
| --- |

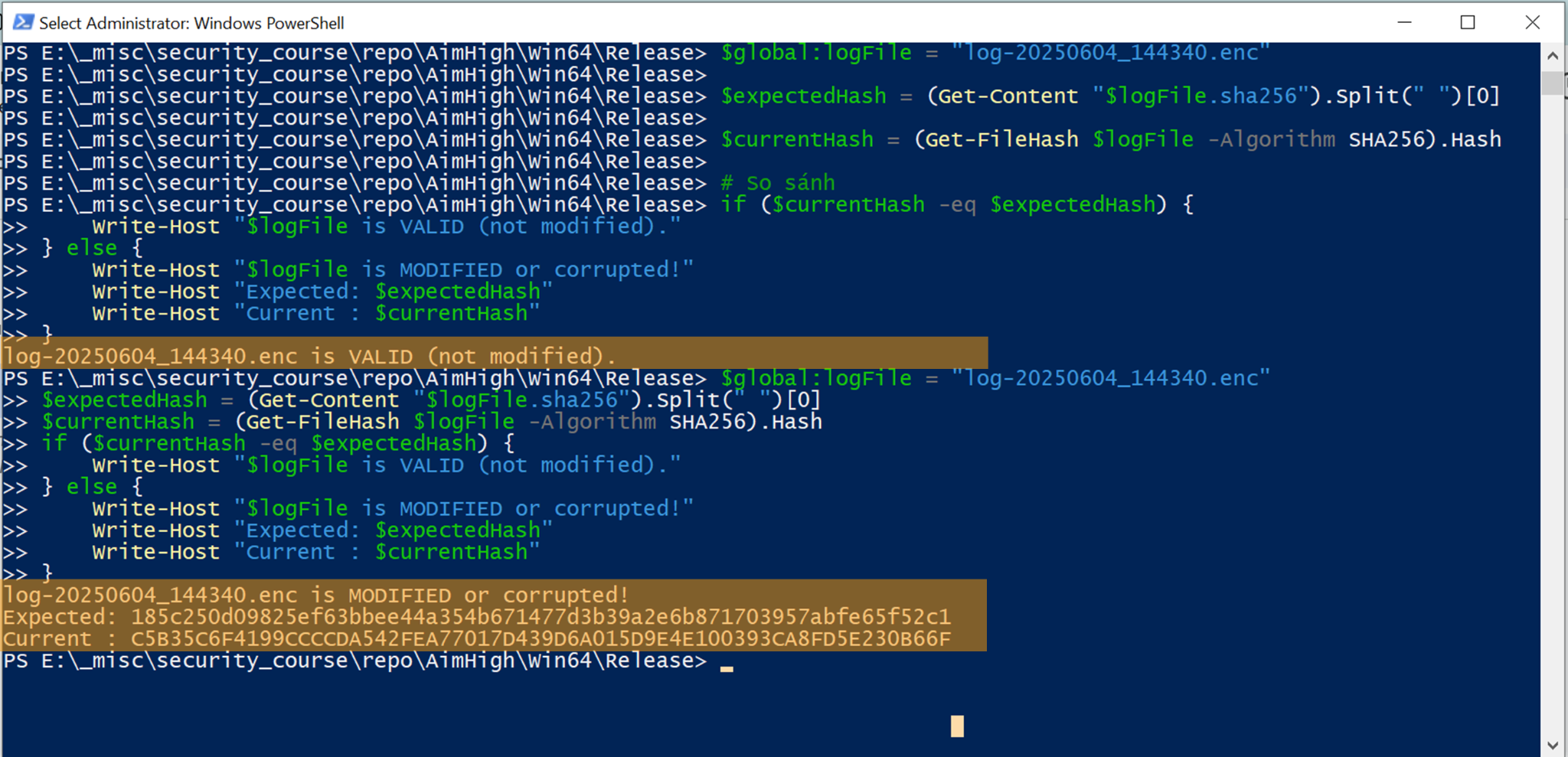
1. Try to modify the log content
2. Close the ADS-B Display
3. Check the file again, a new log has been appended to the file, and show that the log has been modified

| 2025-06-04 16:25:33|ERROR|f9212f80524582516b9babca|7795d23614d9ccd0924c627f1a9bd973|eb3fc552b0ce50c50552016d9d5416cf384e2f4f71b528625247adb753aef45e2578eaf1b773bbf83b2bce49939b49d76c6c9b0efe78757593a175478a0a24ace3ee94ea56c0472e22972c67cee7f8c543af9ffb4cc6bf37fb6e39e39bc4fe2b395d6eeaa476a0d9b8cc15cc76be5e67e00af137411b279853a95641da4592ee34382497b9f98486c2b7812ab0e6a0ef219b8399ecfa37b0e00e1292033c9d93f7e6be6d468ff0ebd832a532|***Decrypted message does not match at line 4: "SBSConnectButtonClick connect: 128.237.96.41:5002" (original) != "SBSConnectButtonClick connect: 128.237.96.41:5003" (modified)*** |
| --- |

Calculate checksum file

1. Press the buttons
2. In the execution file’s folder, check the file in which format log-YYYYMMDD\_HHMMSS.enc
3. Close the ADS-B Display
4. In the execution file’s folder, the checksum file has been created log-YYYYMMDD\_HHMMSS.enc**.*sha256***
5. Open PowerShell in the execution file’s folder, and run below commands  
   Note: You must update the file name

| $global:logFile = ***"log-20250604\_144340.enc"*** $expectedHash = (Get-Content "$logFile.sha256").Split(" ")[0] $currentHash = (Get-FileHash $logFile -Algorithm SHA256).Hash if ($currentHash -eq $expectedHash) {  Write-Host "$logFile is VALID (not modified)." } else {  Write-Host "$logFile is MODIFIED or corrupted!"  Write-Host "Expected: $expectedHash"  Write-Host "Current : $currentHash" } |
| --- |



# Secure Recording data file feature

Overview

* Secure Recording data file feature secures the recording data file
* The User-specific encryption keys are saved on the Login Server

Recording data

1. Press “Raw Connect” / ”SBS Connect” button
2. Press “Raw Record” / ”SBS Record” button, save the data file
3. Press “Stop Raw Recording” / ”Stop SBS Recording” button
4. Check the data file content, the data file content has been encrypted

| 30a46f663aea584c5a676301|4c8b8a3df8ad69f2383b5102f66dff62|4d7f147e34677ae0d713fb8895ee 37e64dc82c115e2f39ad20ba|7f6a02e50ef23f190d2c3a8bfe704f61|7cbb40ceb9ab97ee4ea5ba142c9f6905ddf50ec63173e6453955a7250851ae655788839b64901f9dffd20b91cf96201afa08ec2c341a2b159132b2b9832938e9cca0a65bfb422c4b4db10917e2a8579632e3fd3e83ebf1a2f8239372757cb13ea488fa2c2286 … |
| --- |

Playback data

1. Press “Raw Playback” / ”SBS Playback” button, open the data file (which has been encrypted)
2. The ADS-B Display will playback the flight data

# Setting up a Dump1090 & ADSBHub

Environment settings

* Running on **Raspberry Pi**
* Download the source code from Git

Download Source Code

* git clone: https://github.com/LGSDET/aim\_high\_dump1090

Dump 1090 Build App & Run

* cd aim\_high\_dump1090/

Note: Go to the path where you received the source code.

* Make.
* ./run\_dump.sh

ADSBHub.sh Setting

* cd /usr/bin
* sudo vi adsbhub.sh

Note: You need to change cmd to tls in adsb.sh. Please change the script below.

You need to delete the existing cmd and change it to the cmd below.

| # www.adsbhub.org  # version: 1.06  # ------------------------------------------------------------------  ckey='c$+66q!:B+],ChQAn3[hOlghd~VbT\*Qu-~YV6TPmTjQv-]pt72$gXOVhUX\_{e1mTwy^0T9of~d)3)BOLFY14%zIz!IY(YxU#BZ'  # Remote server  # cmd="nc -w 60 localhost 30002 | nc -w 60 128.237.82.206 5001"  # Remote server 2  # cmd="nc -w 60 localhost 30002 | nc -w 60 128.237.96.41 5001"  # Official configuration  # cmd="nc -w 60 localhost 30002 | nc -w 60 data.adsbhub.org 5001 -vv"  **# TLS enabled**  **cmd="openssl s\_client -connect localhost:30002 -noservername -tls1\_2 -ign\_eof -quiet </dev/null | tee /dev/stderr | nc -w 60 data.adsbhub.org 5001 -vv"**  #cmd="nc -w 60 localhost 30002 | openssl s\_client -connect data.adsbgub.org:5001 -tls1\_2 -noservername -ign\_eof | tee /dev/stderr" |
| --- |

* /usr/bin/adsbhub.sh &