Flight Agent System Security

Triple S

Security Team 2
June 9. 2025

Team Introduction

TripleS - Security Team 2

No	Name	Role
0	Bradley Schmerl	Mentor
1	Sungyoung Choi	Team Leader
2	Taemin Noh	Learning Manager
3	Hwajung Lee	Lunch Manager
4	Soyoon Kim	Photographer
5	Pradeep Kumar C	Time Manager

Scheduling and Role Assignment

“Parallel When Needed, Unified When It Matters”

Phase	Milestone	Key Activity	Assignee	May 22	23	24	25	26	27	28	29	30	31	June 1	2	3	4	5	6	7	8	9
Phase 1	Environment Setup	ADSBHub Account Setup	Noh	●																		
		BigQuery Setup	Choi	●																		
		RUI Setup	Pradeep	●																		
		Raspberry Pi Configuration	Lee, Kim		●																	
	Existing System Analysis	Code Review(Display 2, dump 3)	All			●	●	●	●													
		System Architecture Design/Analysis	All						●	●												
									●	●												
	New Feature Development	Feature Design	Pradeep, Choi, Kim								●	●										
		Feature Implementation / Development	Pradeep, Choi, Kim									●	●	●	●	●	●					
		Test Case Development	Pradeep, Choi, Kim														○	●				
	Vulnerability Analysis & Remediation	Vulnerability Analysis	Noh, Lee							●	●	●	●	○								
		Security Patch Development	Noh, Lee										●	●	●	●	●					
		Test Case Development	Noh, Lee													○	○	●				
	System Integration & Quality Review	Component Integration	All														○	●	●			
		Integrated System Testing	All															○	●			
		Overall Quality Review	All															○	●			
	Presentation	Artifact Documentation	All																○	●	●	
		Presentation and Q&A	Kim																	○	●	●

Planned == Actual ●
Delayed ●
Early started ●
Planned only ○

Data Flow Diagram(DFD)

Asset

- RAW Data(Reliable Data)
- Google Cloud Credential Key
- RUI Program
- Dump1090 service

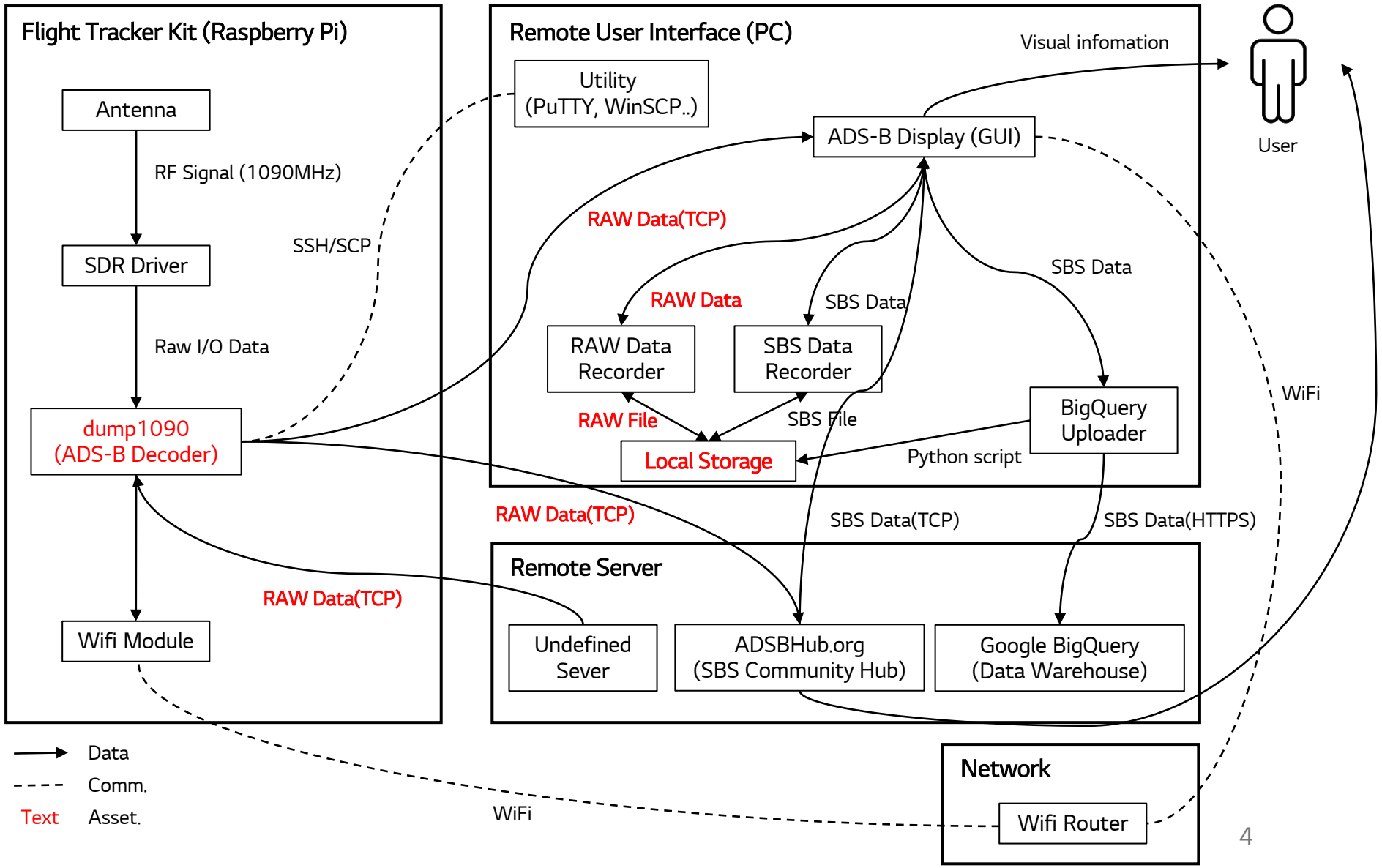
Security Goals

[SG-1] Integrity
RAW data should be ensured not tampered

[SG-2] Credential
Google Cloud Credential Key must be kept secure

[SG-3] Integrity
RUI must ensure the program runs only as intended

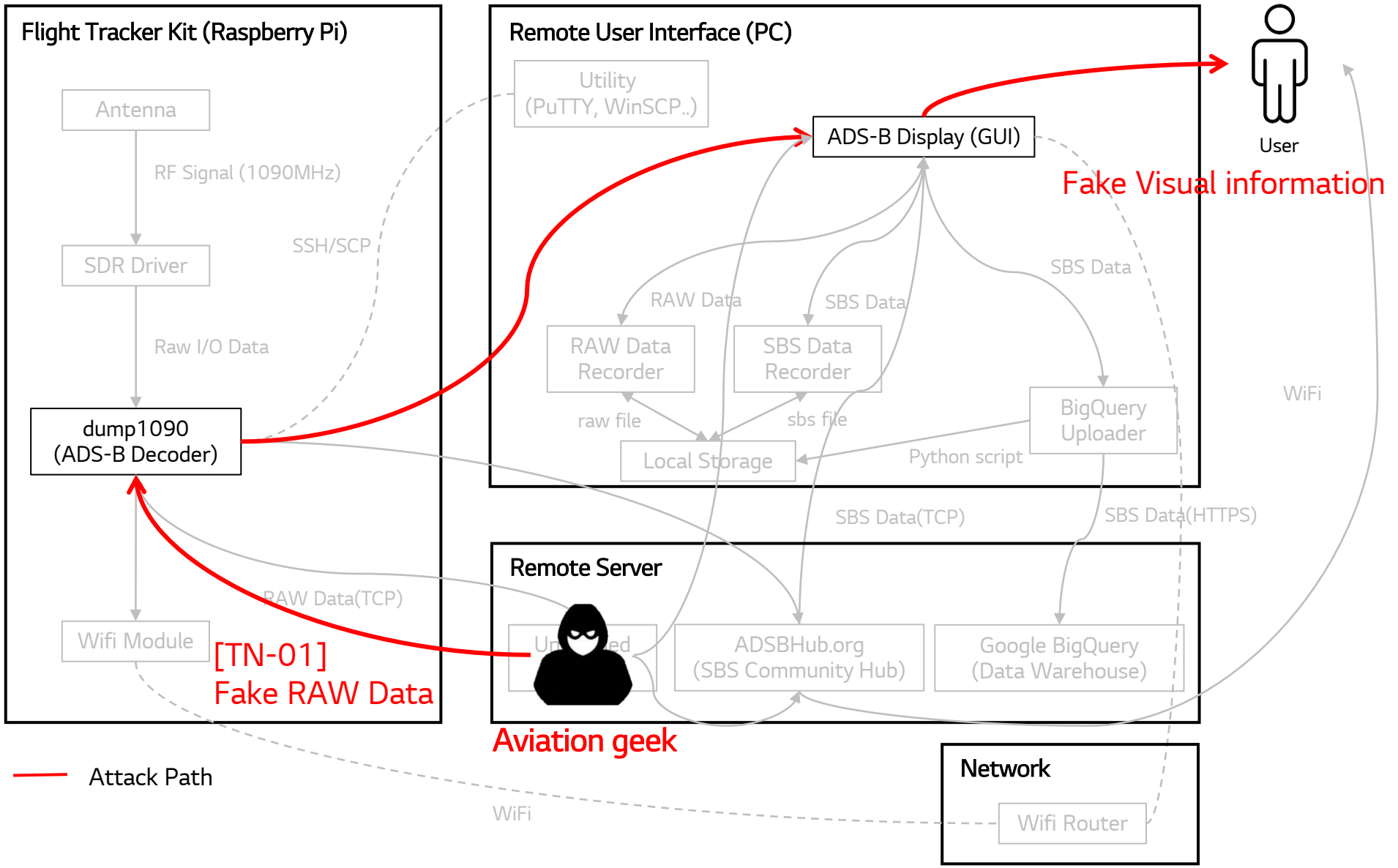
[SG-4] Authentication
Only authenticated users should be allowed to access the dump1090 service



Threat Modeling(hTMM : STRIDE+PnG)

Spoofing

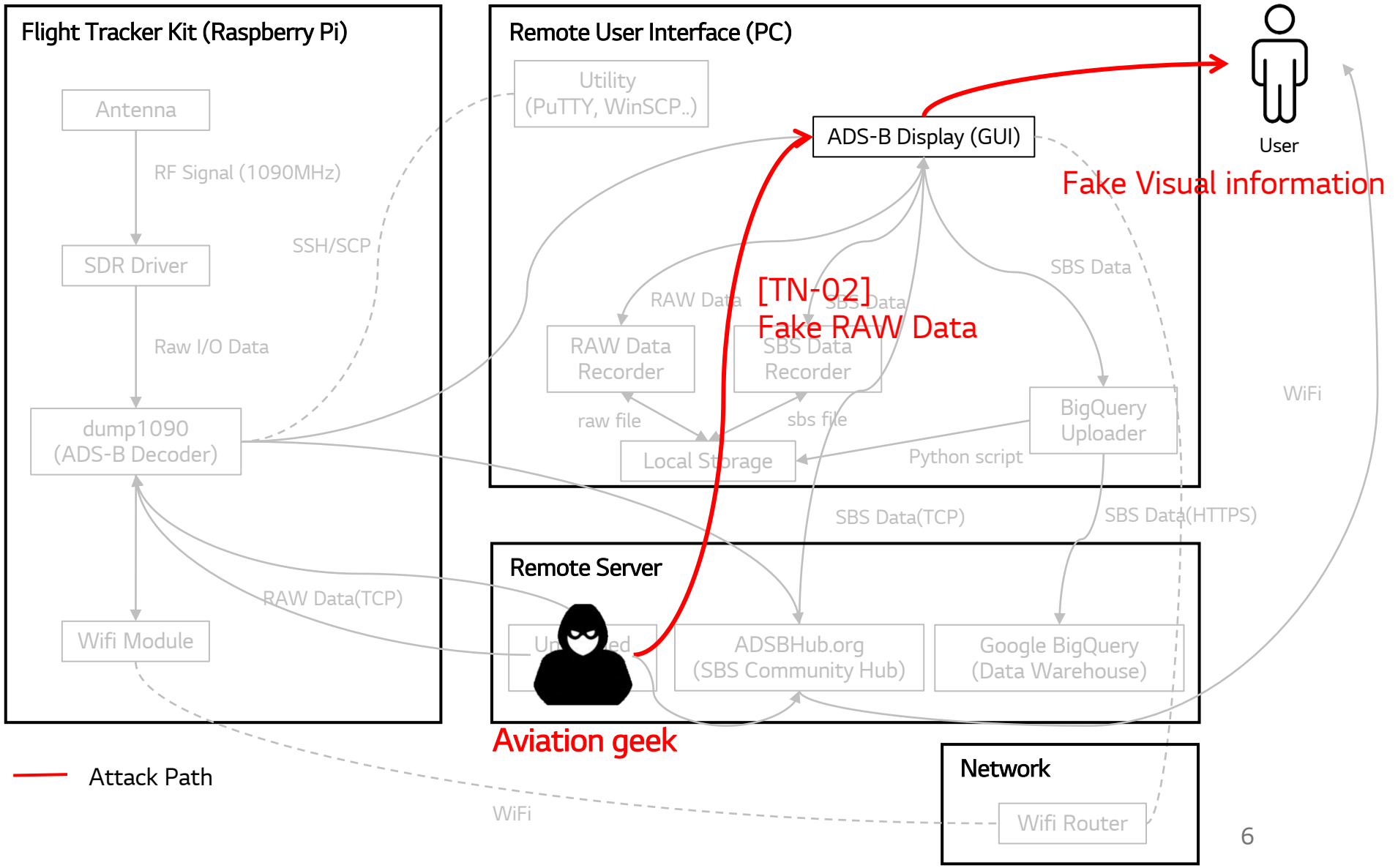
Persona	Aviation geek
Motivation	Studying aviation data for fun and knowledge
Goal	Supply fake ADS-B dat a
Method	Send tampered or proc essed data in RAW for mat to port 30001
Expected Impact	False aircraft shown, Trust lost



Threat Modeling(hTMM : STRIDE+PnG)

Spoofing

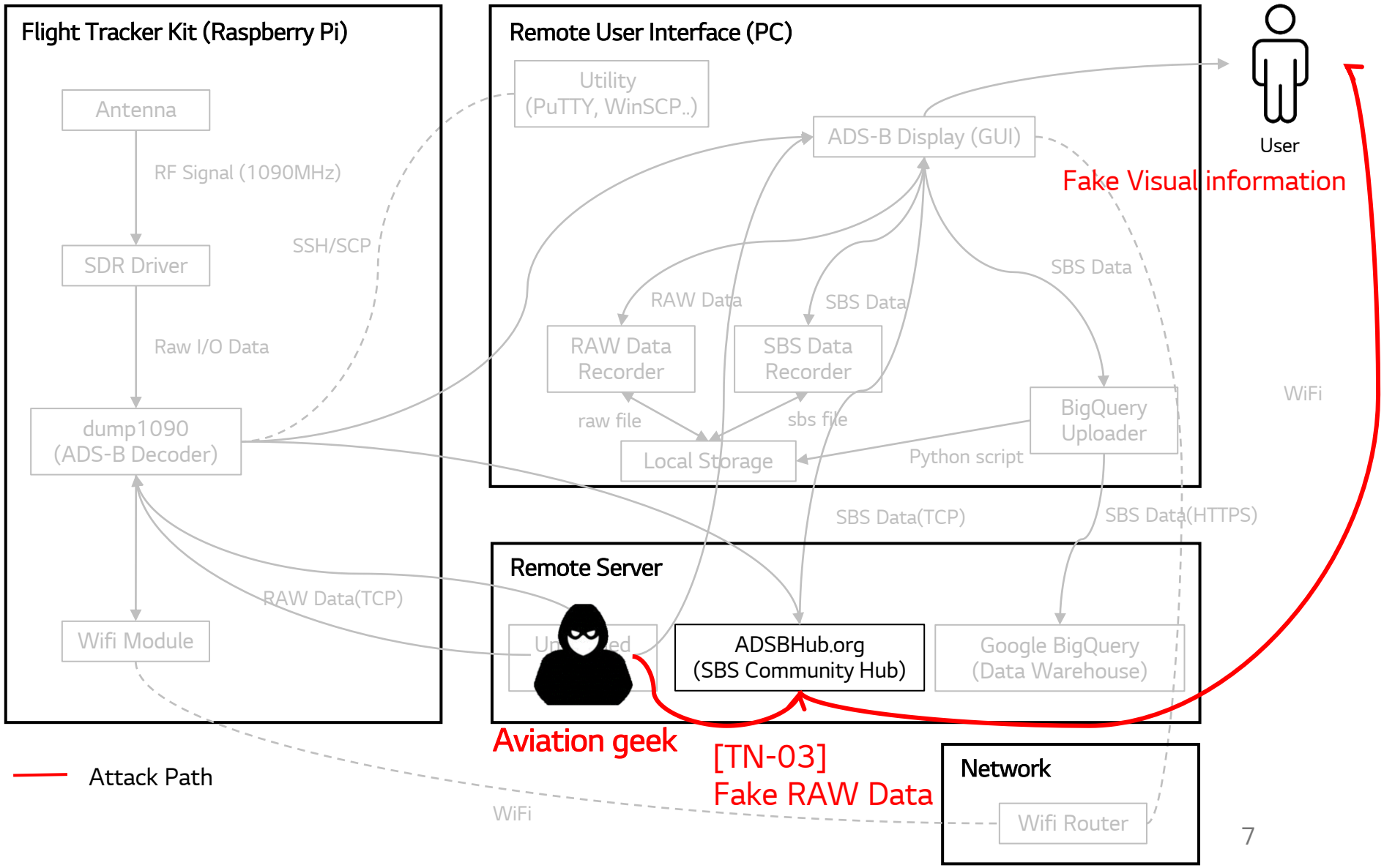
Persona	Aviation geek
Motivation	Studying aviation data for fun and knowledge
Goal	Supply fake ADS-B dat a
Method	Send tampered or proc essed data in RAW for mat to port 30001
Expected Impact	False aircraft shown, Trust lost



Threat Modeling(hTMM : STRIDE+PnG)

Spoofing

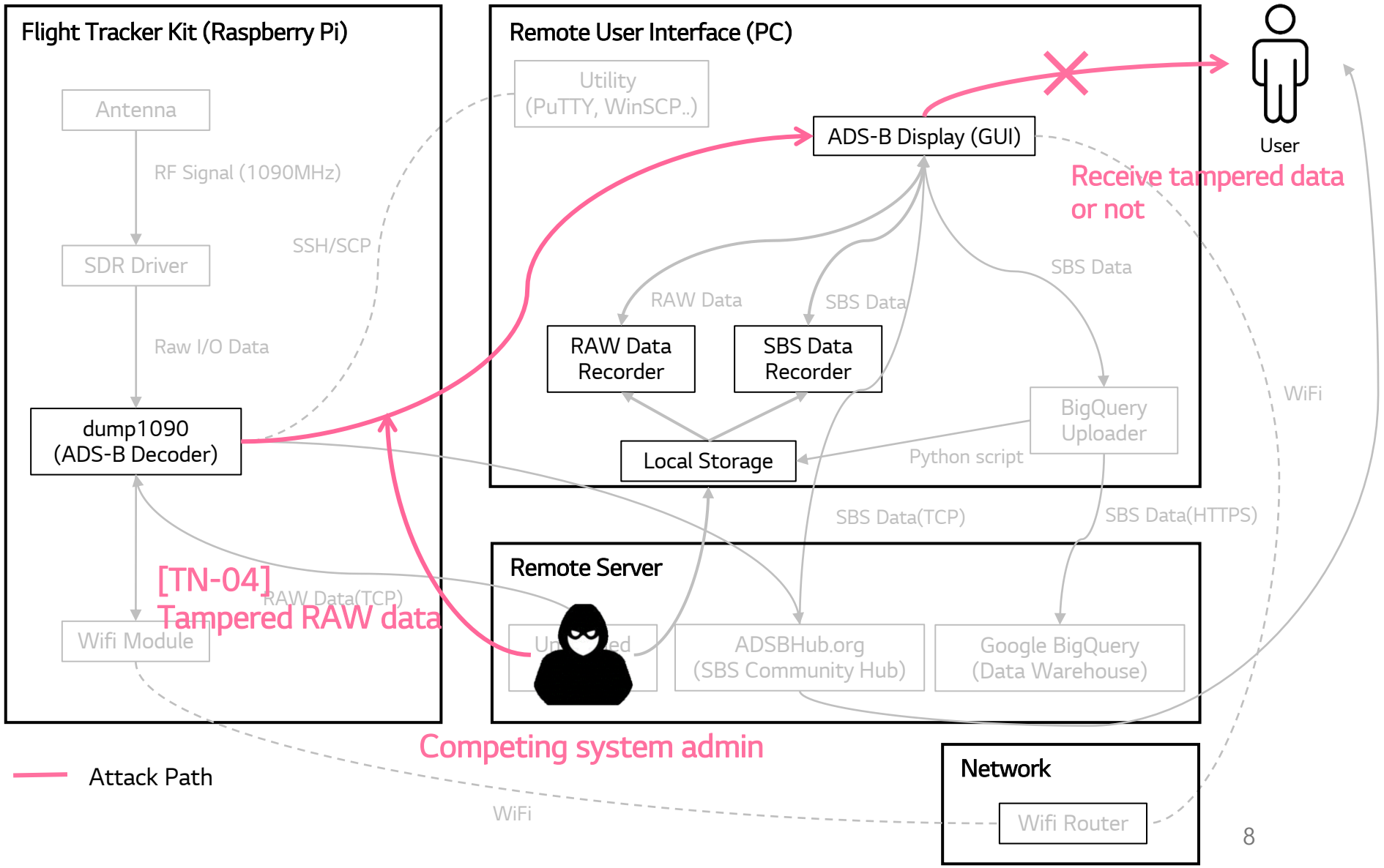
Persona	Aviation geek
Motivation	Studying aviation data for fun and knowledge
Goal	Supply fake ADS-B dat a
Method	Send tampered or proc essed data in RAW for mat to port 30001
Expected Impact	False aircraft shown, Trust lost



Threat Modeling(hTMM : STRIDE+PnG)

Tampering

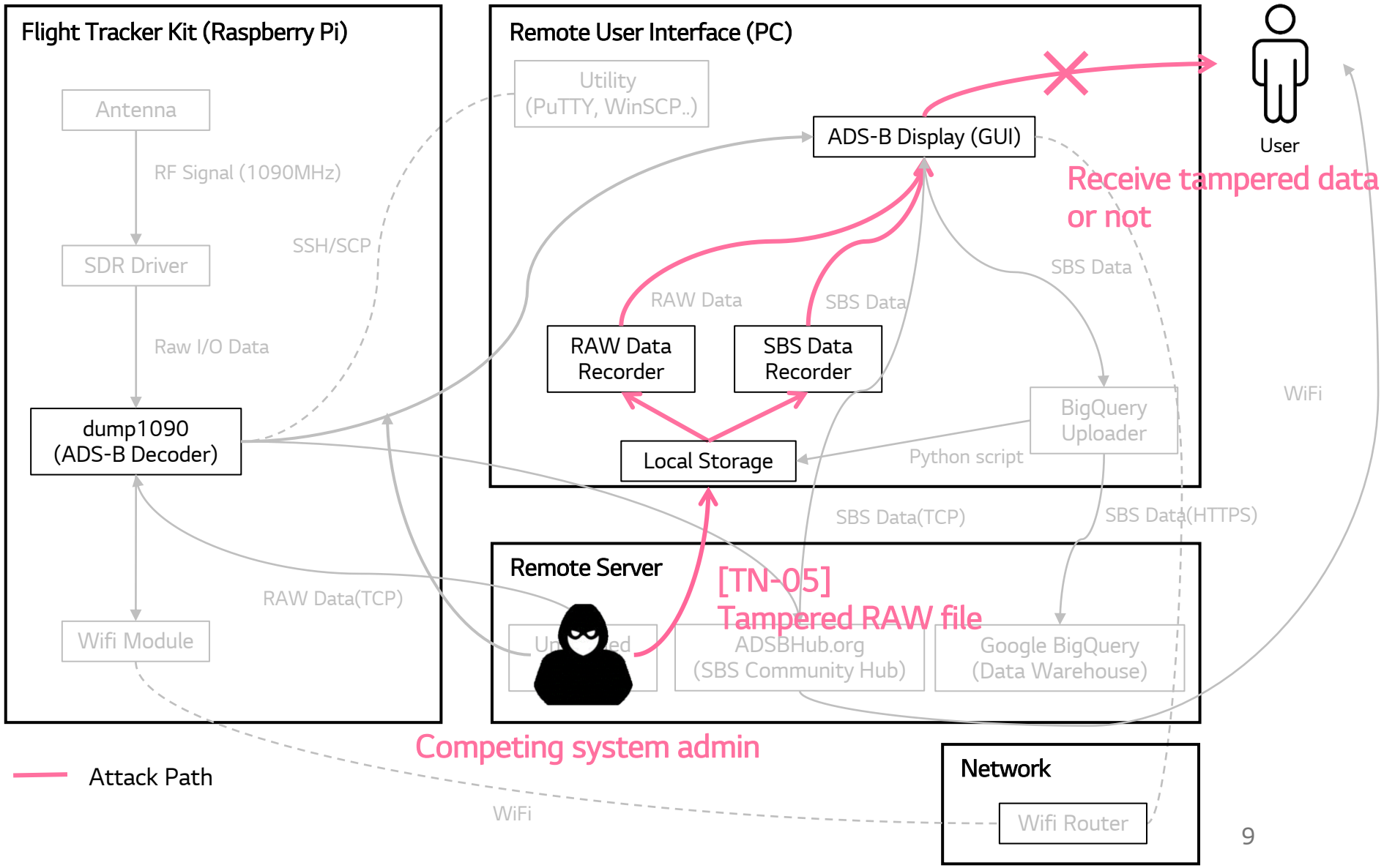
Persona	Competing system admin
Motivation	Making users use a diffe rent(competing service
Goal	Make it seem like the service is down
Method	Distribution of incorrect sample files online Modification of network data via man-in-the-mid dle (MITM) attacks
Expected Impact	Data is not visible to the user on the UI Program aborted



Threat Modeling(hTMM : STRIDE+PnG)

Tampering

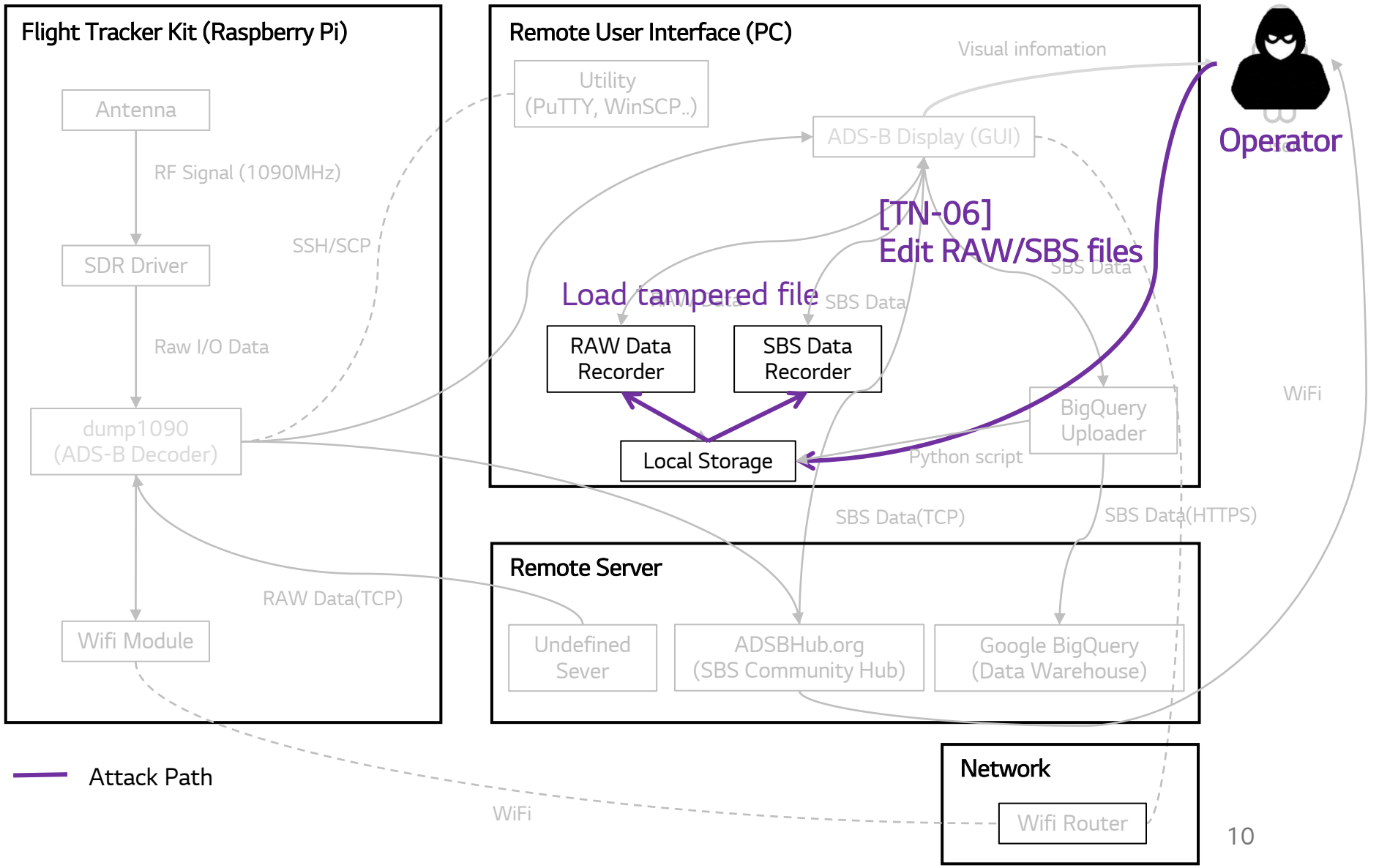
Persona	Competing system admin
Motivation	Making users use a diffe rent(competing service
Goal	Make it seem like the service is down
Method	Distribution of incorrect sample files online Modification of network data via man-in-the-mid dle (MITM) attacks
Expected Impact	Data is not visible to the user on the UI Program aborted



Threat Modeling(hTMM : STRIDE+PnG)

Repudiation

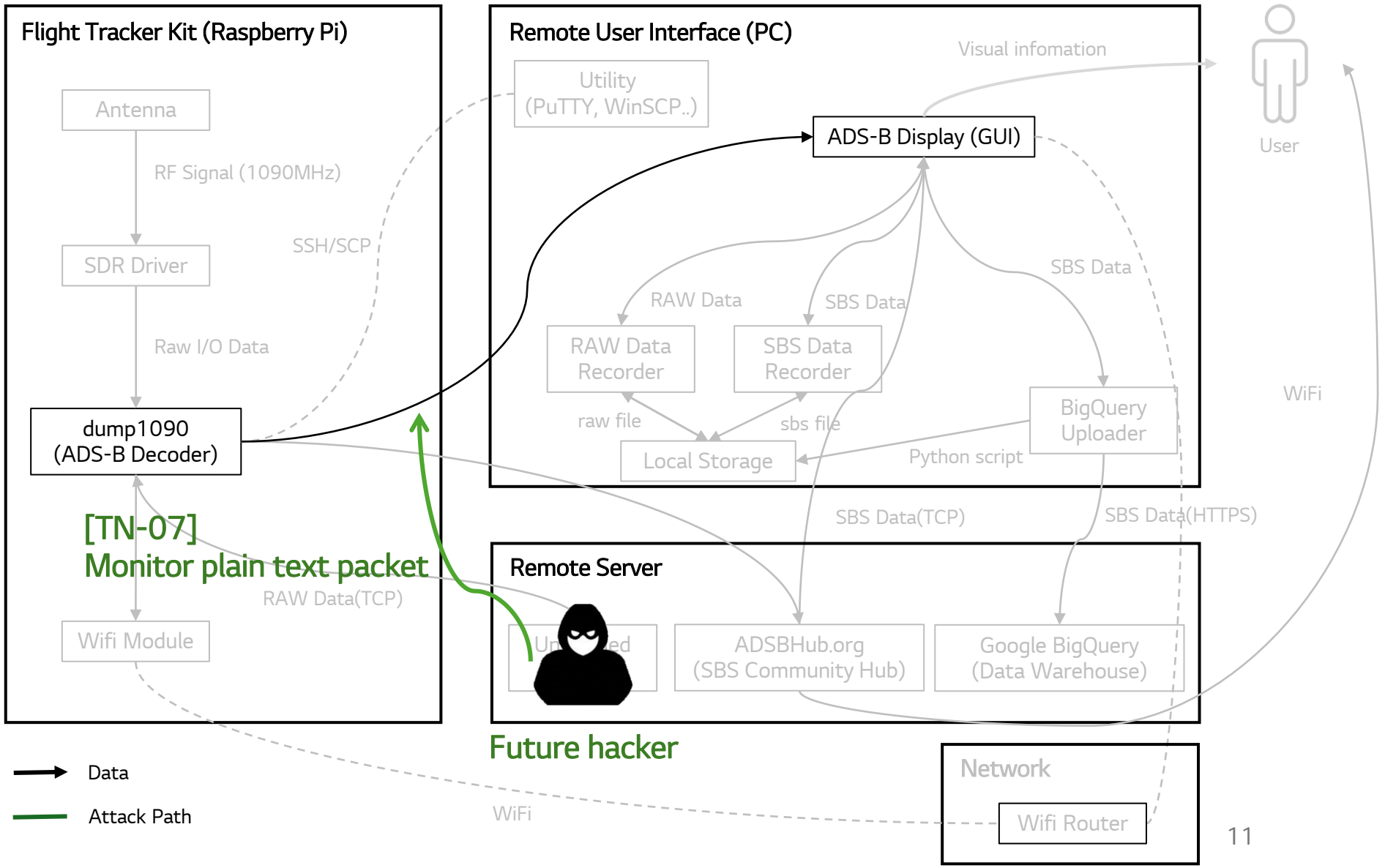
Persona	Operator
Motivation	Avoid accountability
Goal	Manipulate save/load
Method	Delete logs / Fake sav e
Expected Impact	No audit trail Compliance failure



Threat Modeling(hTMM : STRIDE+PnG)

Information Disclosure

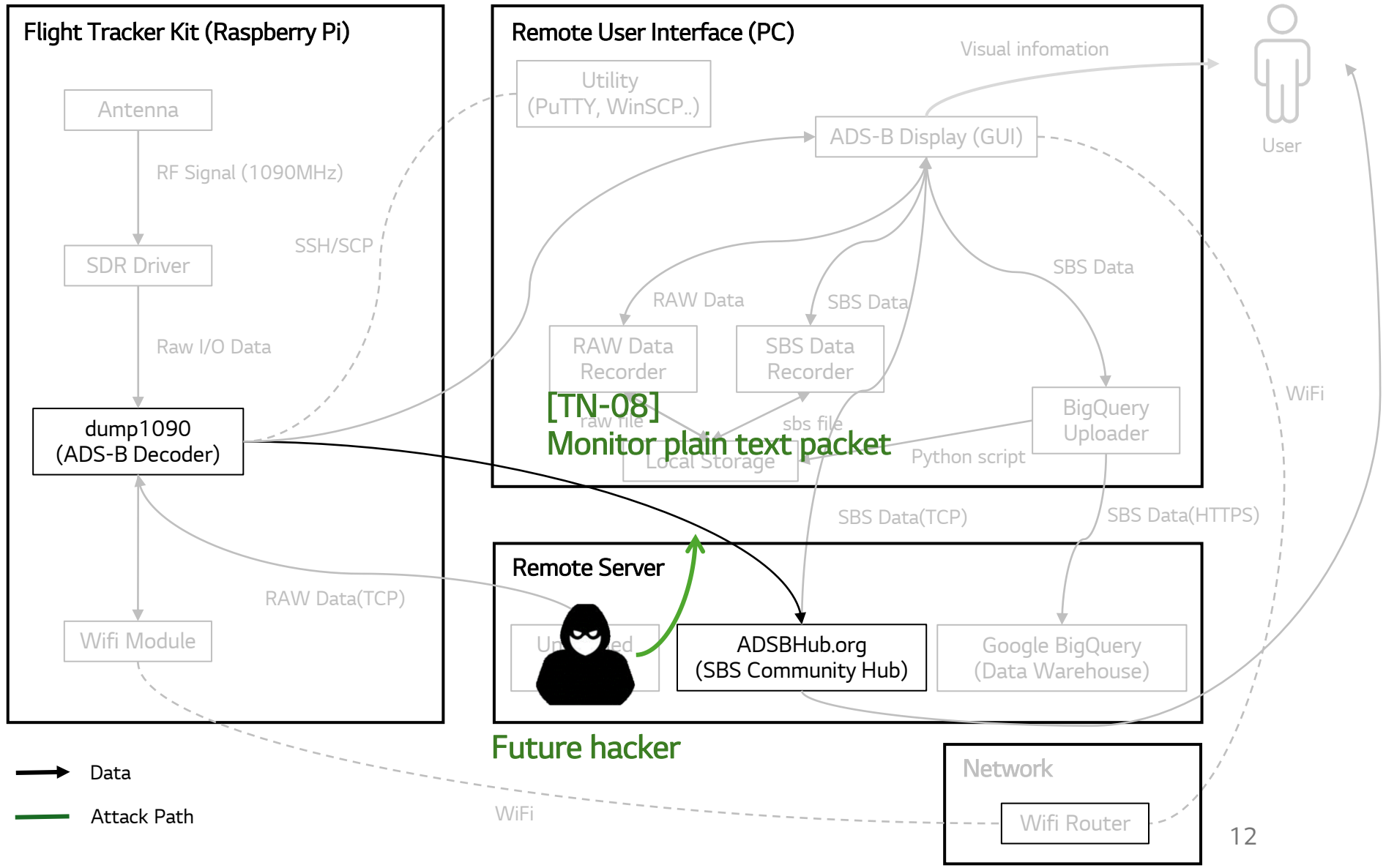
Persona	Future Hacker
Motivation	Information gathering prior to launching a cy berattack
Goal	Monitor the data exchanged by the user
Method	Network packet sniffi ng w/ Wireshark
Expected Impact	An attacker gathers in formation on users wh o receive dump data



Threat Modeling(hTMM : STRIDE+PnG)

Information Disclosure

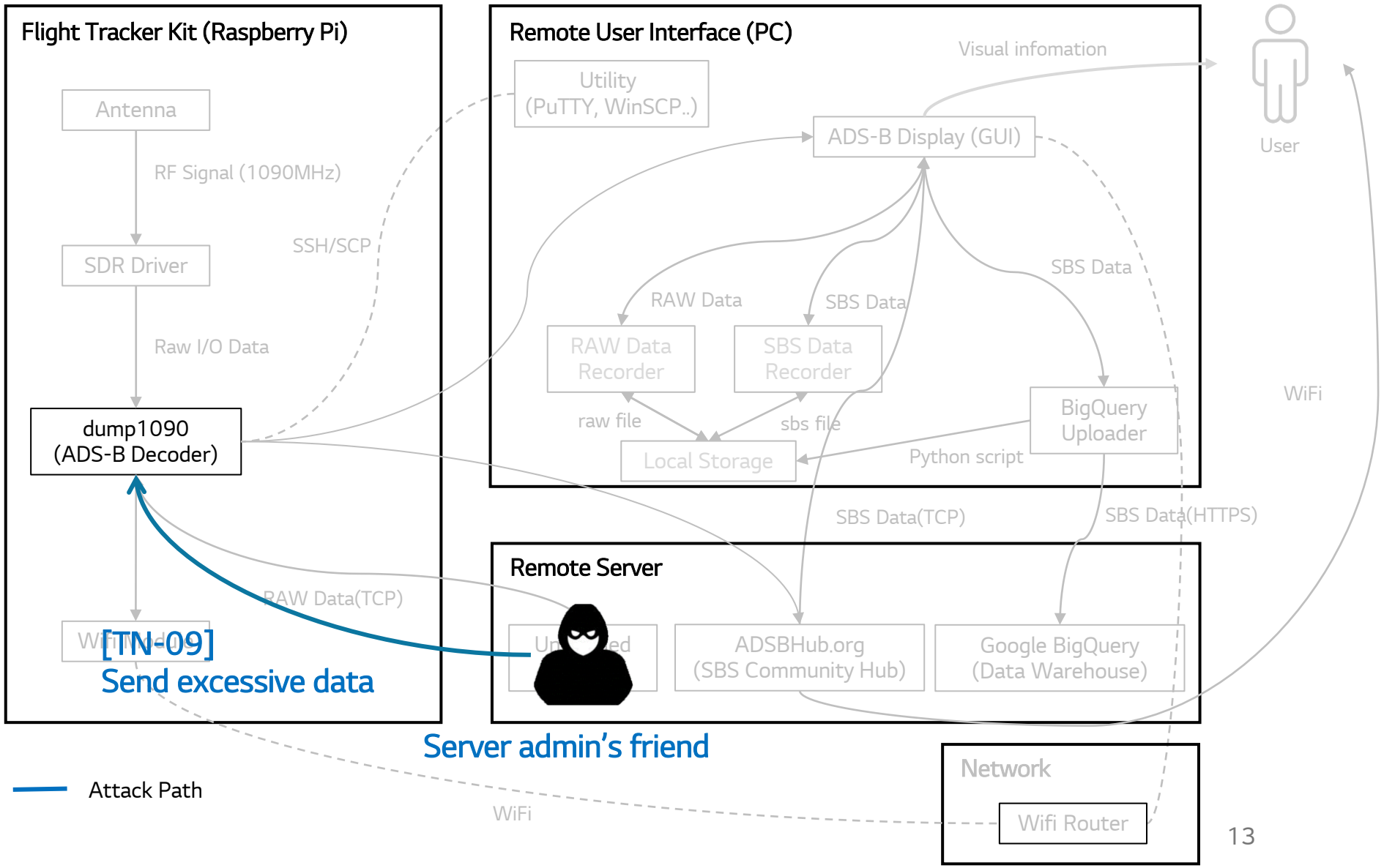
Persona	Future Hacker
Motivation	Information gathering prior to launching a cy berattack
Goal	Monitor the data exchanged by the user
Method	Network packet sniffi ng w/ Wireshark
Expected Impact	An attacker gathers in formation on users wh o receive dump data



Threat Modeling(hTMM : STRIDE+PnG)

Denial of Service

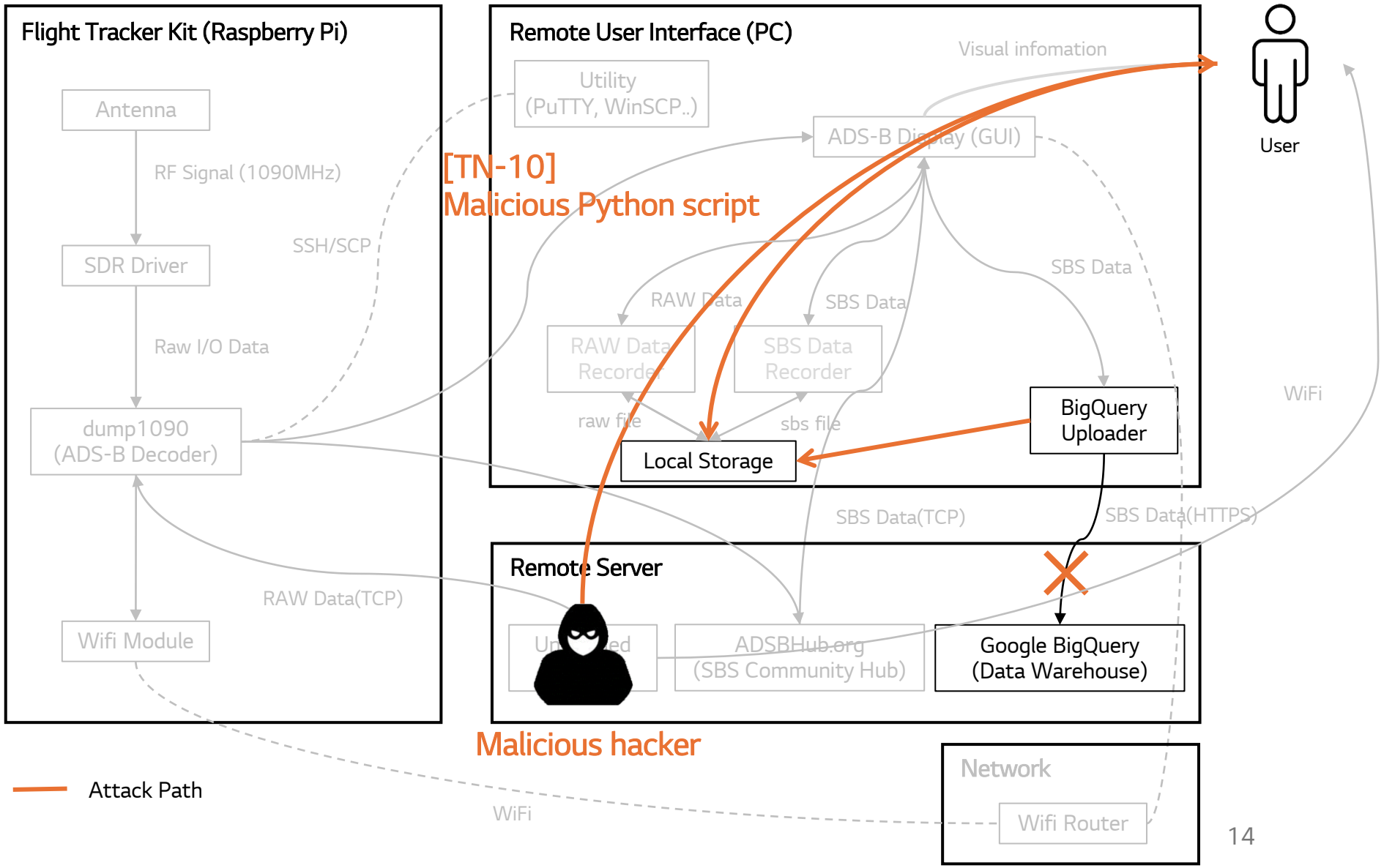
Persona	The server admin's fri end
Motivation	Personal revenge
Goal	Exhaust dump resourc es
Method	Send excessive data using automative script or tool
Expected Impact	Slowdown or crash Service



Threat Modeling(hTMM : STRIDE+PnG)

Elevation of Privilege

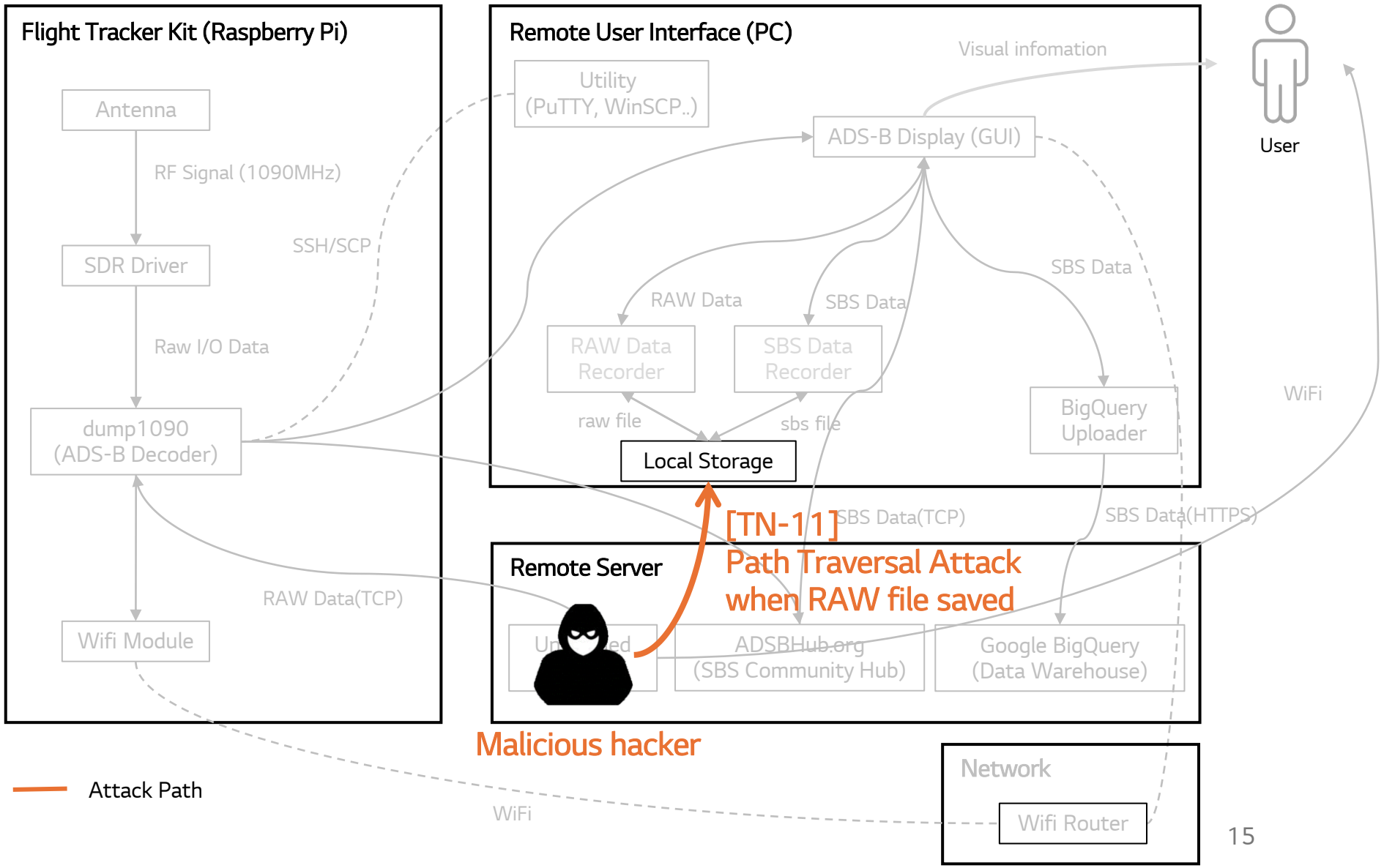
Persona	Malicious hacker
Motivation	Profit
Goal	Obtaining system control and extracting GCS credential
Method	Download malicious Python script via phishing emails or Git repository
Expected Impact	Privilege gain System compromise



Threat Modeling(hTMM : STRIDE+PnG)

Elevation of Privilege

Persona	Malicious hacker
Motivation	Profit
Goal	Obtaining system control and extracting GCS credential
Method	Download malicious Python script via phishing emails or by distributing through a fake Git repository
Expected Impact	Privilege gain System compromise



Risk Assessment : Prioritizing

TN ID	Threat Summary	Impact	Priority	Priority Basis
TN-01	**Spoofed aircraft signal transmission (attacker → dump1090)**	Destruction of data trust foundation Incorrect decision-making based on forged data Inducing user/organization confusion Distortion of alarm and automation judgment Loss of customer trust → Risk of contract termination	3	Since the system relies on data accuracy, Spoofing shake its foundation, leading to real-time malfunctions.
TN-02	Spoofed aircraft signal transmission (attacker → RUI)		-	This spoofing was not prioritized because it required a phishing attack that would lead the victim to enter the attacker's IP address.
TN-03	Spoofed aircraft signal transmission (attacker → ADSBHub)		-	
TN-04	**Data tampering during transmission (dump1090 → ADSBHub)**	Integrity breach, automation malfunction, analysis error, system corruption	2	Since the system relies on data accuracy, Tampering shake its foundation, leading to real-time malfunctions.
TN-05	**Tampered playback file provision (attacker → PC)**	Analysis system contamination, malicious file, misjudgment, attacker influence		
TN-06	Responsibility for file manipulation (attacker → PC)	Change untraceable, tampering unprovable, unclear accountability, trust degradation	4	There may be issues with responsibility when a problem occurs, but it is not fatal to the system.
TN-07	Plain-text communication interception (dump1090 → RUI)	Unintended data leak	5	Mitigated by TLS (to some extent)
TN-08	Plain-text communication interception (dump1090 → ADSBHub)	Communication interception risk	-	Hard to access, low damage potential, blocked in internal network
TN-09	Induction of high traffic to port 30001 (attacker → dump1090)	Temporary overload, delayed reception	-	No impact on system reliability
TN-10	**Execution of malicious file/script (attacker → PC)**	Root privilege escalation, total system control loss, sensitive data exfiltration	1	Elevation of Privilege poses fundamental risks due to the potential for complete system takeover
TN-11	**Modification/deletion of key system files (attacker → RUI)**	File tampering/deletion, privilege escalation, service disruption		

Security Requirements

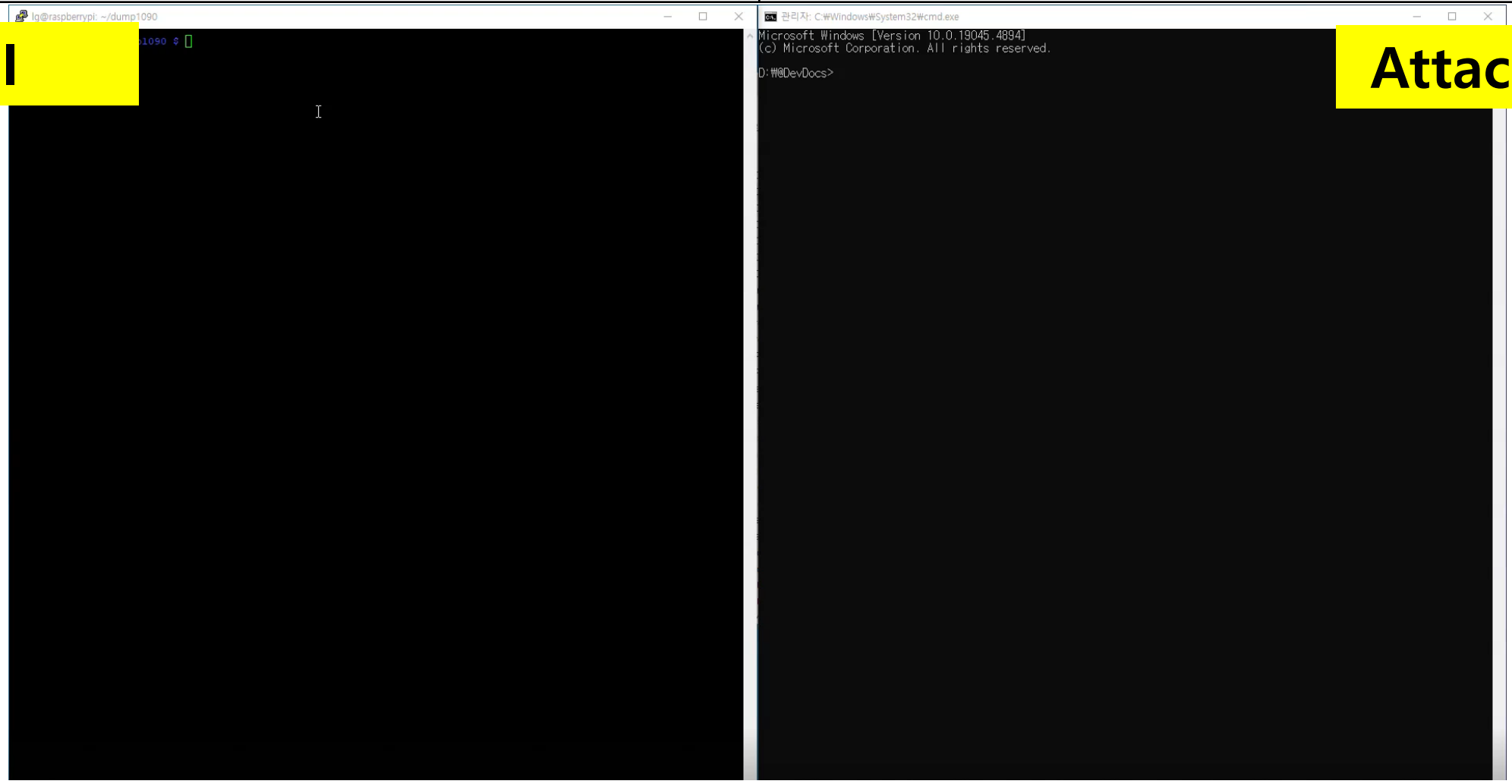
SR#	TN#	SG#	Security Requirement
SR-01	TN-01 TN-09	SG-01	Access control for port 30001 of dump1090
SR-02	TN-04 TN-07	SG-01	Data communication between dump and RUI must be encrypted, ensured protection against tampering
SR-03	TN-05 TN-06	SG-01	It must be possible to verify the integrity of the RAW data file to detect any tampering
SR-04	TN-10	SG-02 SG-03	The system shall verify the integrity and authenticity of Python scripts before execution
SR-05	-	SG-04	As part of the business policy, an authentication system must be implemented to manage users
SR-06	TN-11	SG-03	The system must enforce that data is not written outside the intended storage location.
SR-07	-	SG-04	The server must enforce the principle of least privilege.

To achieve the security objectives, apply a **defense-in-depth** strategy

Mitigation

SR#	Security Requirement	Mitigation Method
SR-01	Access control for port 30001 of dump1090	**Filtering external incoming IPs based on whitelist**

Before



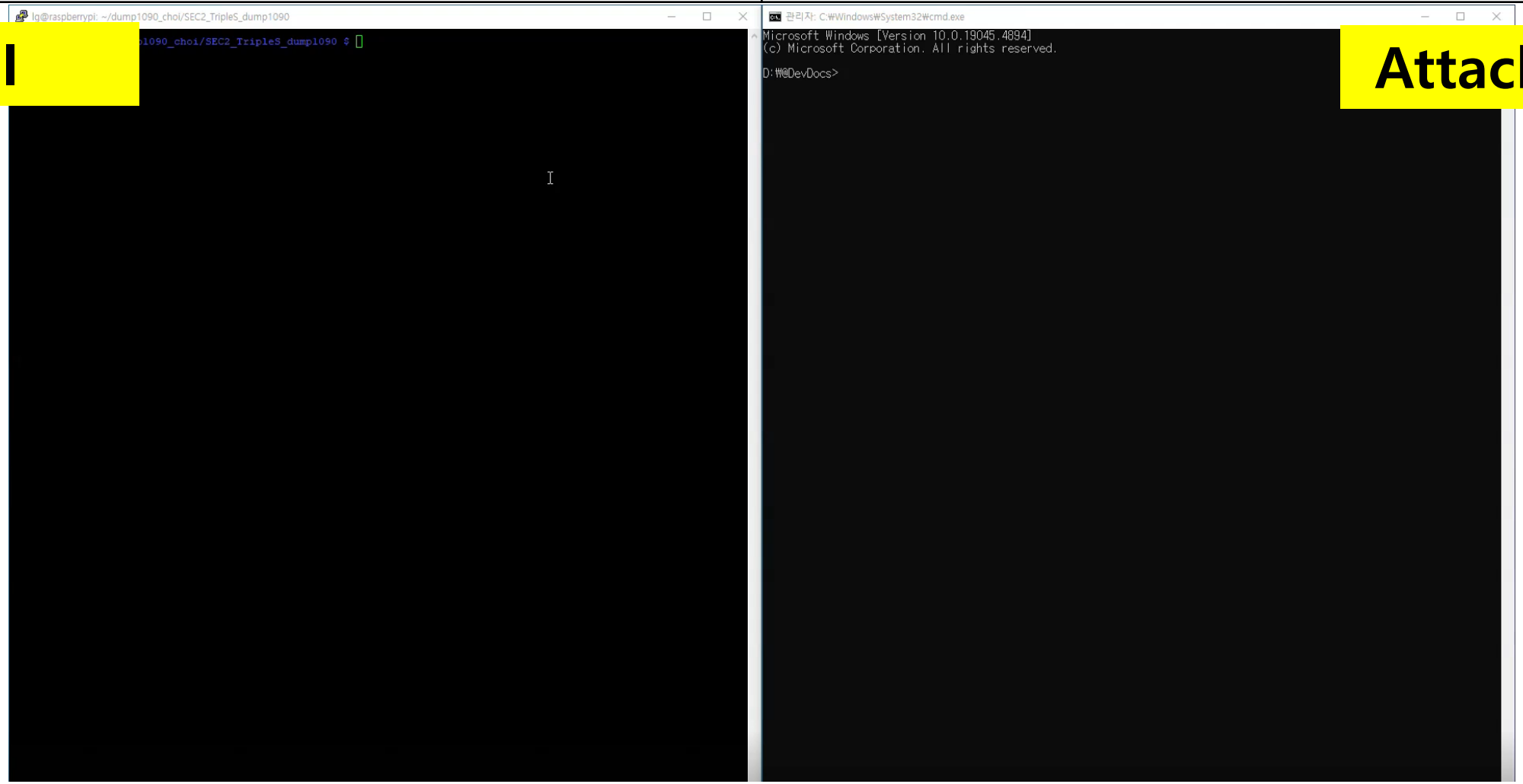
Mitigation

SR#	Security Requirement	Mitigation Method
SR-01	Access control for port 30001 of dump1090	**Filtering external incoming IPs based on whitelist**

PI

Attacker

After

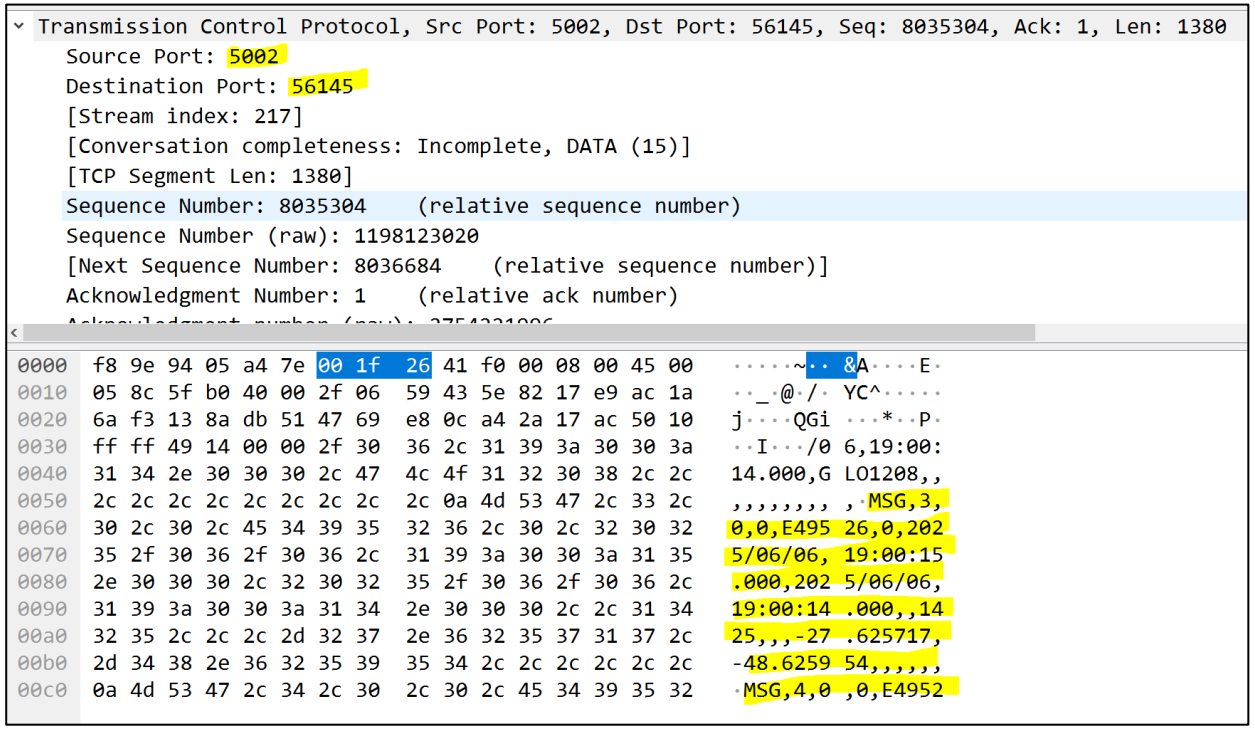


Mitigation

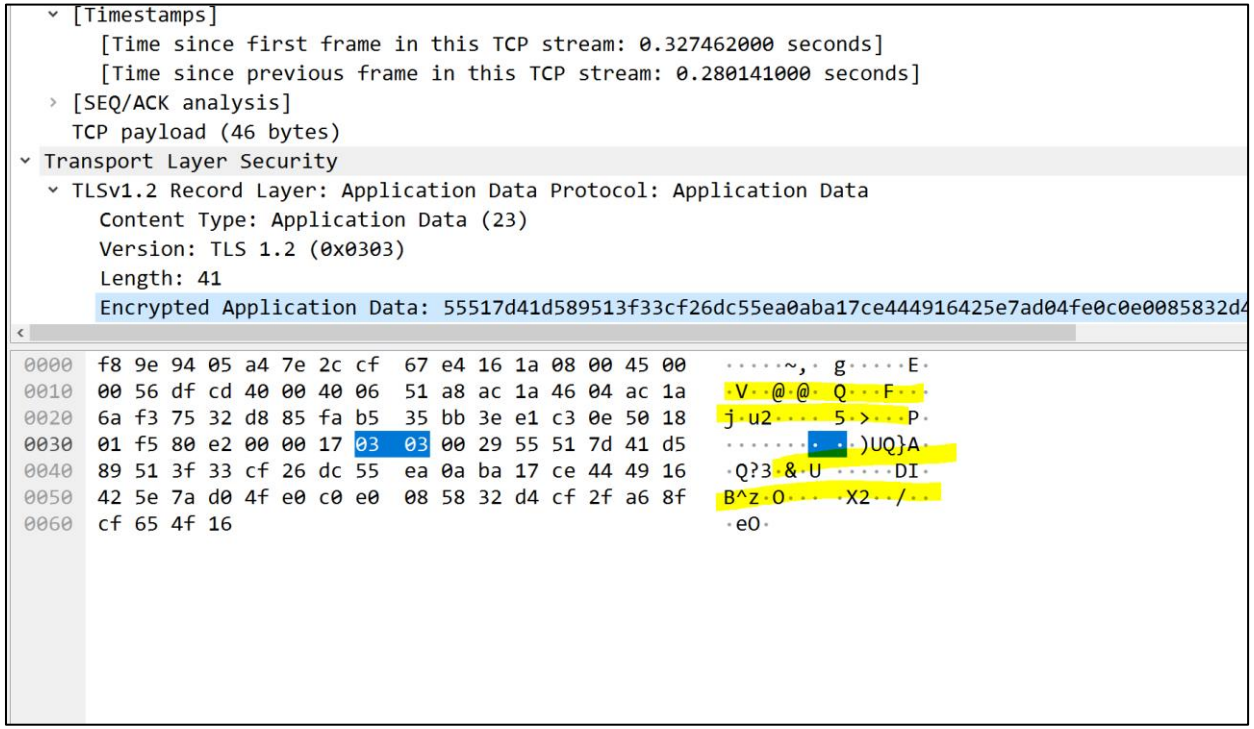
SR#	Security Requirement	Mitigation Method
SR-02	Data comm.. between dump and RUI must be encrypted, ensured protection against tampering	Apply TLS to prevent forgery and ensure encrypted communication

Wireshark view

Before



After



Plain text can be seen with sniffing

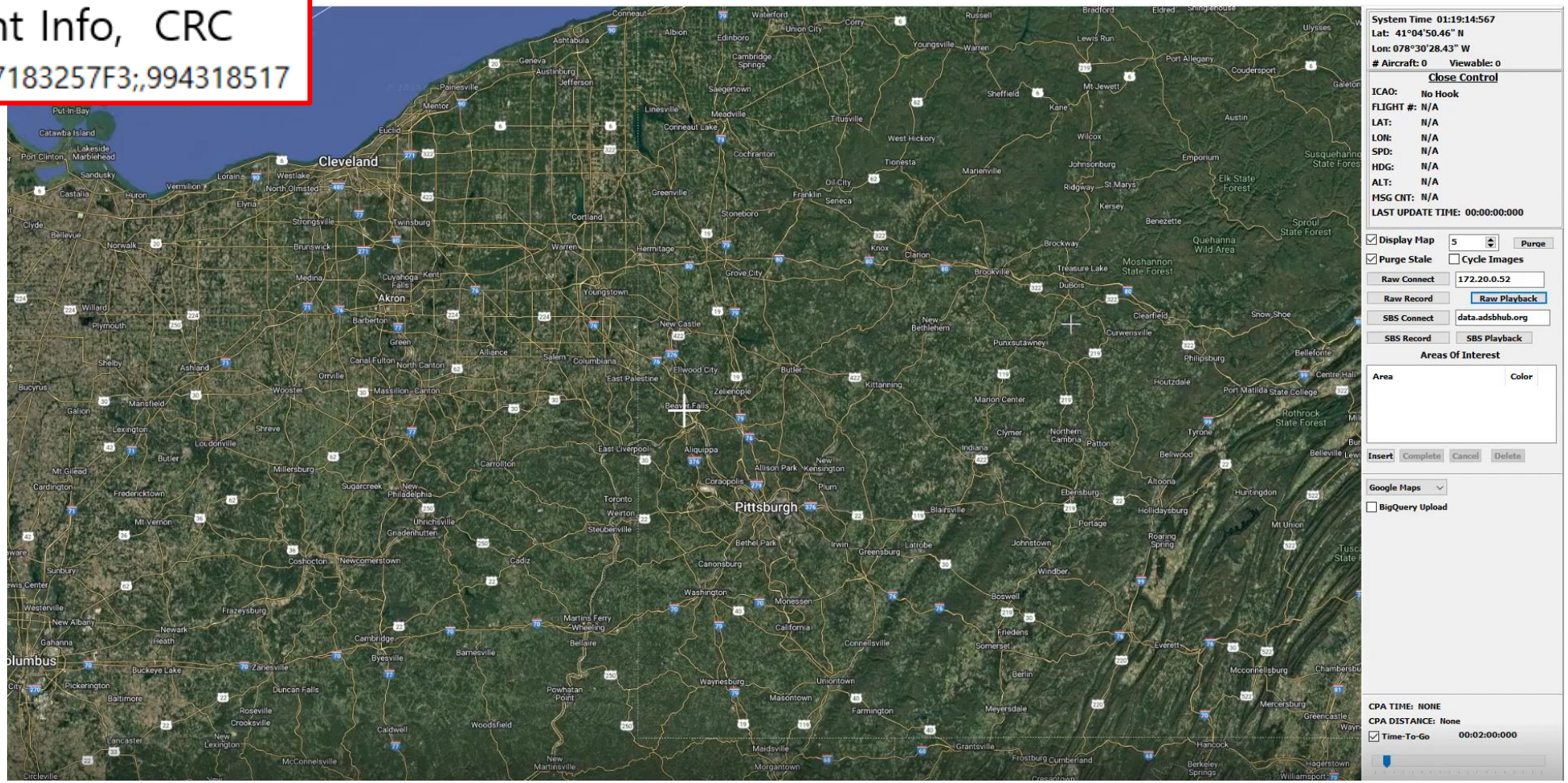
Encrypted text can not be seen with sniffing

Mitigation

SR#	Security Requirement	Mitigation Method
SR-03	It must be possible to verify the integrity of the RAW data file to detect any tampering	Perform CRC validation by timestamp when saving or loading RAW data.

Timestamp, Flight Info, CRC

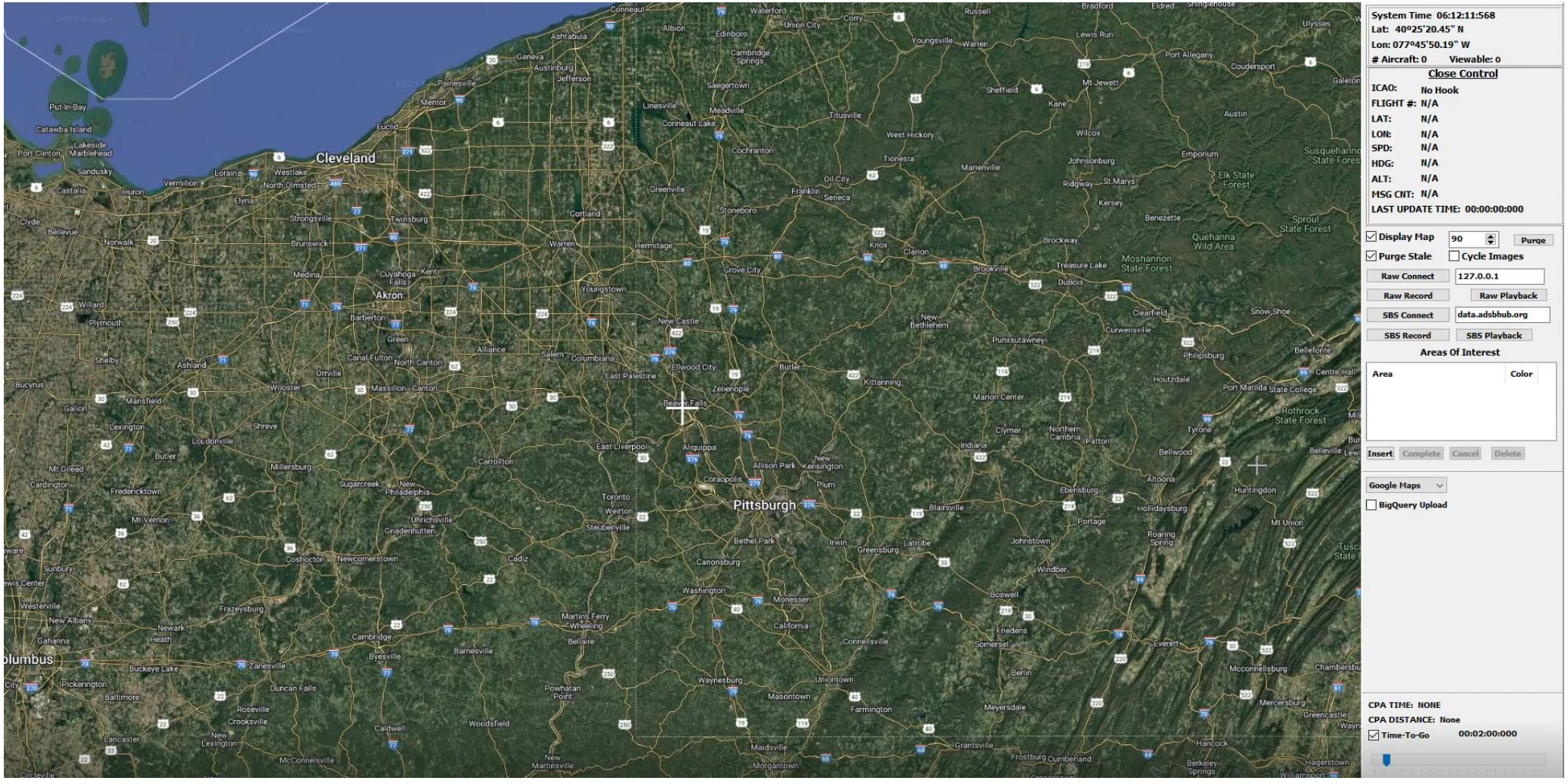
13393410788092,*02E617183257F3;,994318517



Mitigation

SR#	Security Requirement	Mitigation Method
SR-04	The system shall verify the integrity and authenticity of Python scripts before execution	Compare the hash value of the Python script before execution to ensure integrity

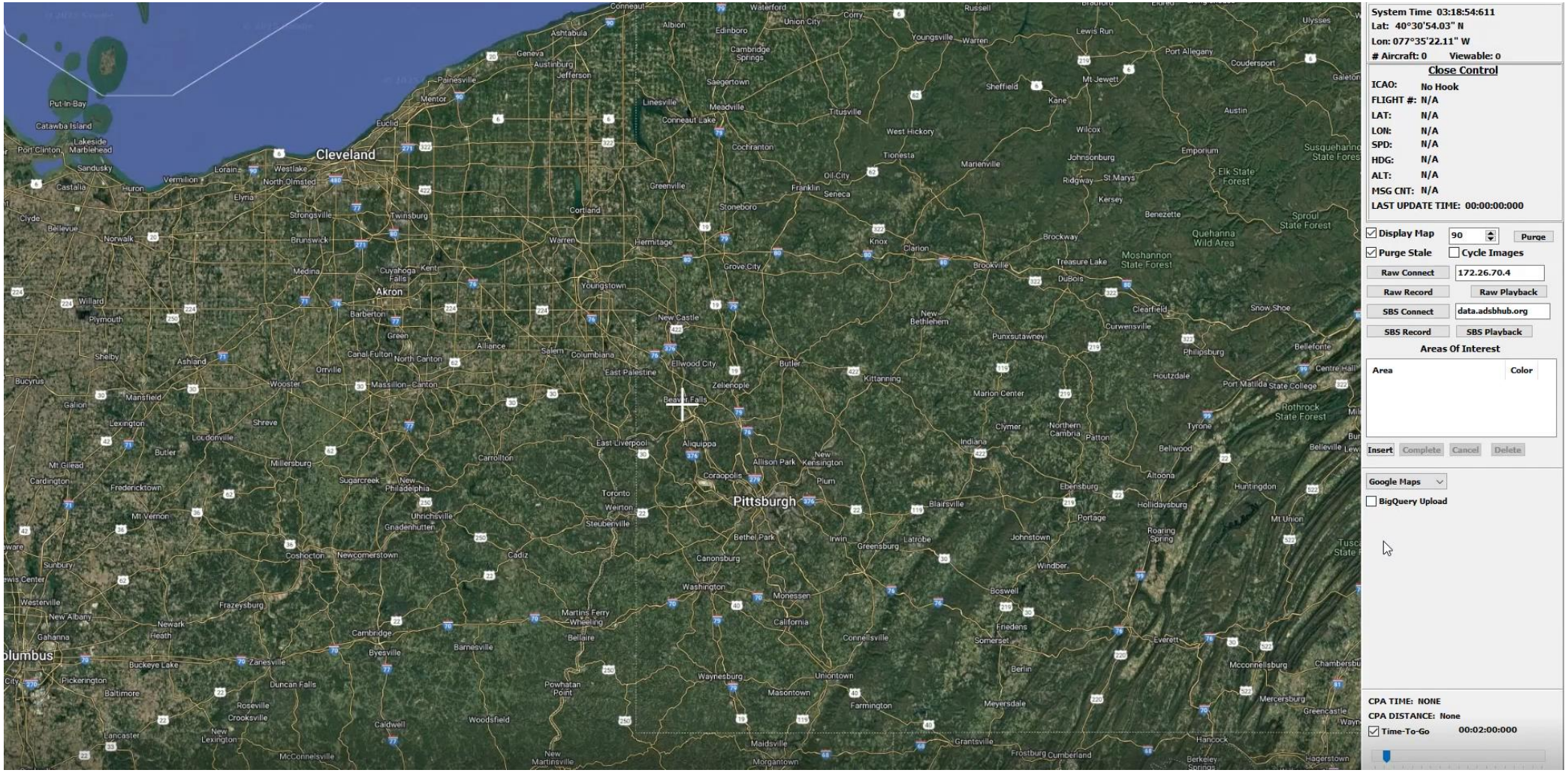
Before



Mitigation

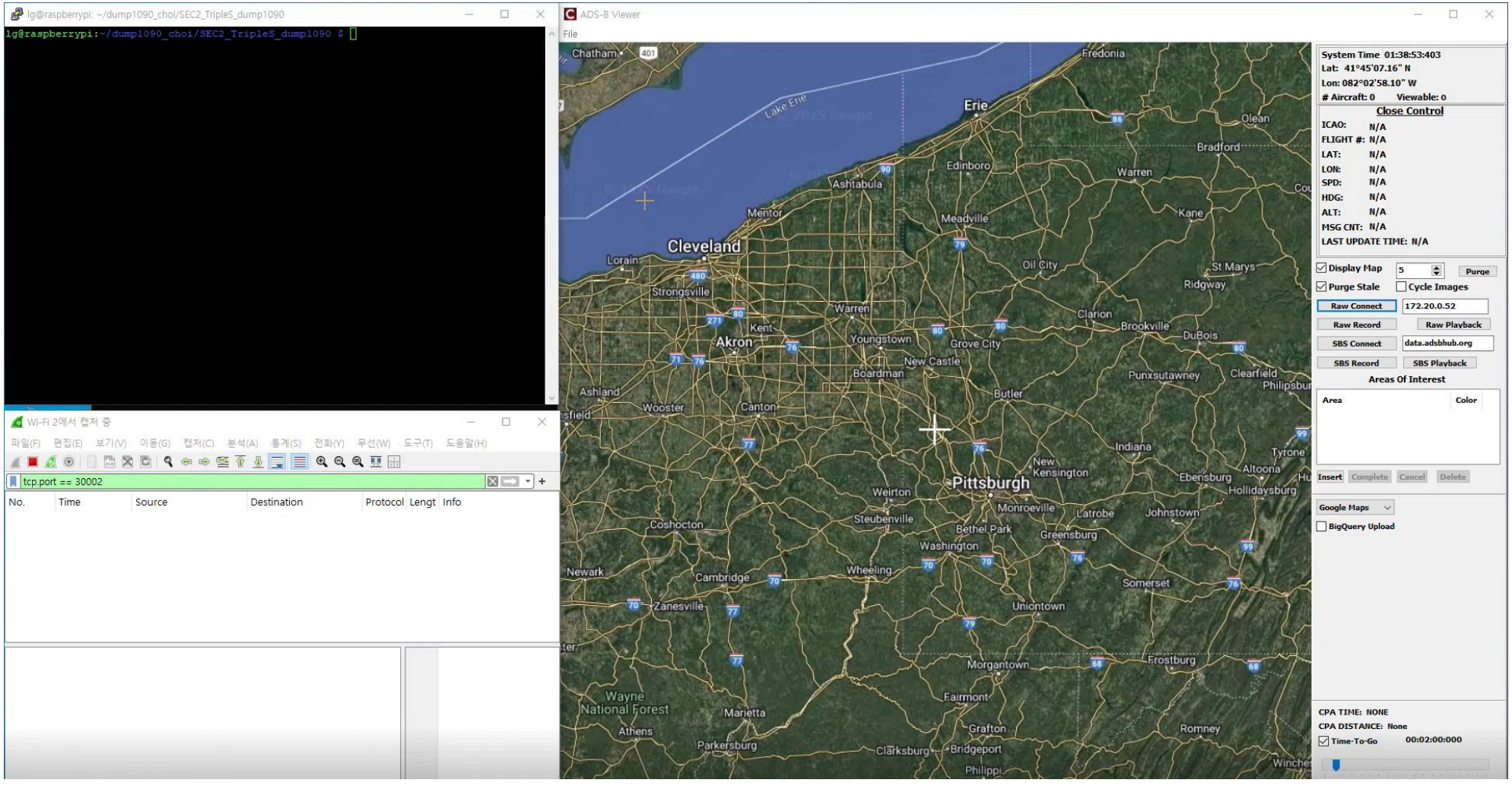
SR#	Security Requirement	Mitigation Method
SR-04	The system shall verify the integrity and authenticity of Python scripts before execution	Compare the hash value of the Python script before execution to ensure integrity

After



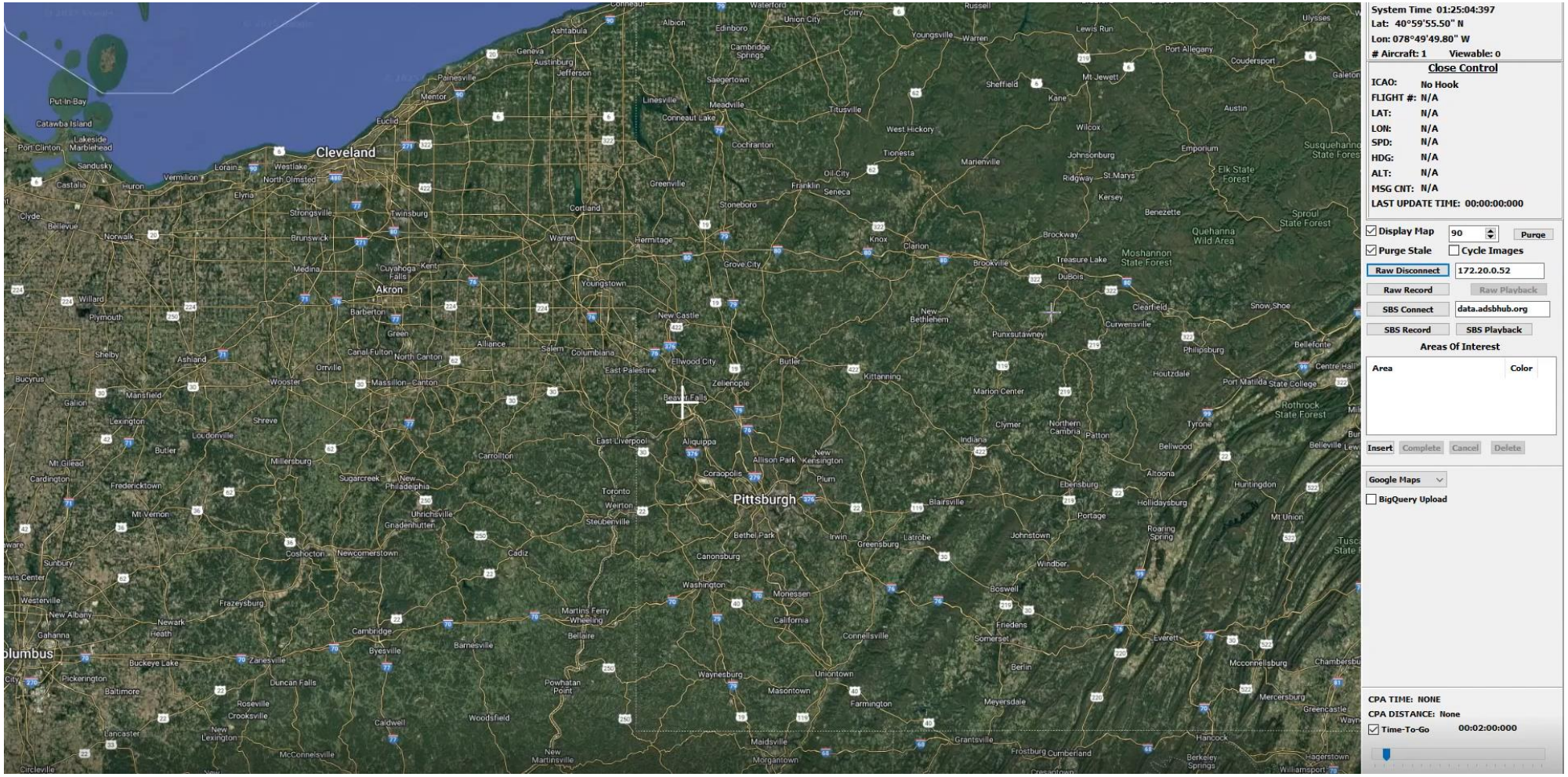
Mitigation

SR#	Security Requirement	Mitigation Method
SR-05	As part of the business policy, an authentication system must be implemented to manage users	Implement a login function to allow access only to authorized users.



Mitigation

SR#	Security Requirement	Mitigation Method
SR-06	The system must enforce that data is not written outside the intended storage location.	Restrict the storage path for RAW files.



Mitigation

SR#	Security Requirement	Mitigation Method
SR-07	The server must enforce the principle of least privilege.	Create a service account and grant it only the permissions necessary for service operation.

	Service	Guest
Public Key	Read	Read
Private Key	Read	-
Password	Read/Write	-

```
lg@raspberrypi:~ $ ls -la /etc/ssl/tripleS/
total 20
drwxr-xr-x 2 root tripleS 4096 Jun  6 07:33 .
drwxr-xr-x 5 root root    4096 Jun  3 20:12 ..
-rw-r--r-- 1 root tripleS 1322 Jun  3 20:13 cert.pem
-rw-rw---- 1 root tripleS  138 Jun  6 07:33 config.json
-rw-r----- 1 root tripleS 1704 Jun  3 20:12 key.pem
lg@raspberrypi:~ $ cat /etc/ssl/tripleS/key.pem
cat: /etc/ssl/tripleS/key.pem: Permission denied
lg@raspberrypi:~ $ sudo -u tripleS cat /etc/ssl/tripleS/key.pem
-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQChF7I90mefVeQv
I4ZbxEIsXkrsXEmFlHezfi3aA3Igr+wNXulhsclrJ3GUySOJ9MWP7VUulq+xQzYZ
ka2+c2LAKBhP+0l/MDivW0GgkGKoio0QUbCunXJr3n+xtvpPOJZbWM+fdljDvcLo
```

Secure coding & Supply Chain Security

In-house Vulnerability Scanning System : Snyk.
It shows known vulnerability of the open source library
We are going to apply vulnerability patch using vulnerability list.

Threat Level

CRITICAL	HIGH
0	1
MEDIUM	LOW
85	0

Issue details

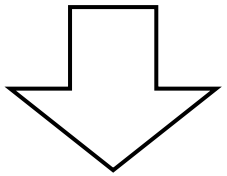
Aligned with OWASP/CWE

	SCORE ↓				PROJECT	EXPLOIT MATURITY	COMPUTED FIXABILITY	INTRODUCED	SNYK PRODUCT
H	804	Vulnerability Hardcoded Secret	Not Available	CWE-547	hwajung7lee/SEC2_TripleS_dump1090(master)		Not Applicable	Jun 8, 2025	Snyk Code
M	581	Vulnerability Potential buffer overflow from usage of unsafe function	Not Available	CWE-122	hwajung7lee/SEC2_TripleS_RUI(main)		Not Applicable	Jun 8, 2025	Snyk Code
M	581	Vulnerability Potential buffer overflow from usage of unsafe function	Not Available	CWE-122	hwajung7lee/SEC2_TripleS_RUI(main)		Not Applicable	Jun 8, 2025	Snyk Code
M	581	Vulnerability Use After Free	Not Available	CWE-416	hwajung7lee/SEC2_TripleS_RUI(main)		Not Applicable	Jun 8, 2025	Snyk Code
M	581	Vulnerability Use After Free	Not Available	CWE-416	hwajung7lee/SEC2_TripleS_RUI(main)		Not Applicable	Jun 8, 2025	Snyk Code
M	581	Vulnerability Potential buffer overflow from usage of unsafe function	Not Available	CWE-122	hwajung7lee/SEC2_TripleS_RUI(main)		Not Applicable	Jun 8, 2025	Snyk Code

SCRM (A-SQUARE-Based Security Assessment)

Component : OpenSSL 3.5
Application Context : TLS channel for transmitting user ID/password during authentication

- Defined Role : Supports secure user authentication
- Security : Utilizes verified cryptographic algorithms
- Maintainability : Maintained by the OpenSSL Project foundation
- Release : Long-Term Support (LTS) versions available
- Risk & Mitigation : Incompatibility with past API and Update it
- Operational Trustworthiness : Security scanning tools integrated in CI/CD pipeline



* Identified risks are manageable and mitigated
* Meets all A-SQUARE evaluation criteria

Accepted and safely integrated as a trusted component

Learning points

Hwajung

I realized that it is important to identify the entry points of a system from an attacker's perspective, rather than just searching for vulnerabilities in the code.

Sungyoung

Implementing encryption and TLS, it was interesting to see how it actually worked in the code phase.

Pradeep

Learned that AI can help in clearing majority of concepts and helps in developing concepts quickly, it can also mislead and can not answer some queries correctly like specific to open source code like indy TLS.

Taemin

Through various vulnerability analyses and security designs, I have come to realize the importance of collaboration and systematic thinking skills.

Soyoon

It was a new and challenging experience, unlike previous projects. It's a pity I couldn't apply SecDevOps or a logging system.

Thank you

Triple S

Why was hTMM used in the threat modeling phase, and why was STRIDE specifically applied?”

By incorporating AI feedback throughout the security design,
including threat identification, vulnerability classification, and data flow protection,
we were able to build a more reliable system.

Even though it was an English document-based project,
AI enabled us to accurately organize concepts,
and in particular, the security requirements analysis and documentation process
was carried out efficiently.

Why was a centralized login mechanism based on the Pi selected instead of aircraft-specific authentication?

Way1	RUI-Based Login	Raspberry Pi-Based Login	RUI Sends Credentials to Pi (Securely)
Pros	- Centralized UI, consistent login experience. - Flexibility in UI and authentication logic. - Potentially simpler Raspberry Pi code.	- Control sending RAW data(Stop sending data if not authenticated) - Enhanced security, credential storage on Raspberry Pi.	- Centralized UI in RUI. - Security: Authentication handled by Raspberry Pi. - Simplified Raspberry Pi code. - Clear separation of concerns.
Cons	- Network dependency: Requires connection before RUI is usable - Security considerations: RUI handles credentials (even temporarily) - Potential redundancy if Pi also needs authentication.	- **No UI** (tied to Pi). - Increased Raspberry Pi resource usage.	- Network dependency. - **Complexity of secure transmission (TLS/SSL).** - Potential latency due to authentication process.

What was the primary purpose of utilizing AI?

By incorporating AI feedback throughout the security design,
including threat identification, vulnerability classification, and data flow protection,
we were able to build a more reliable system.

Even though it was an English document-based project,
AI enabled us to accurately organize concepts,
and in particular, the security requirements analysis and documentation process
was carried out efficiently.

Q & A

How can an attacker replace a victim's files?

1st step:

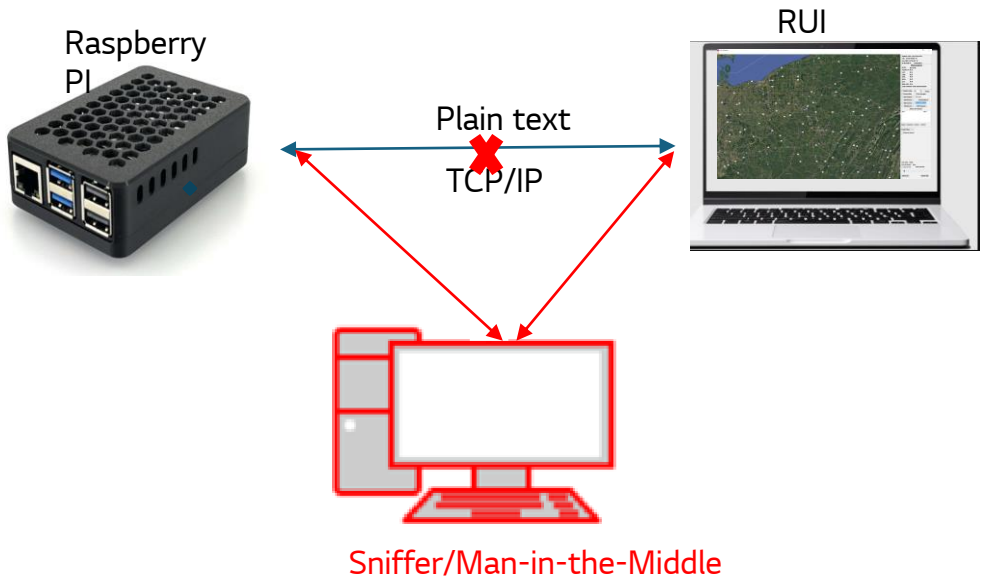
The attacker impersonates the operator and sends a phishing email to the victim.

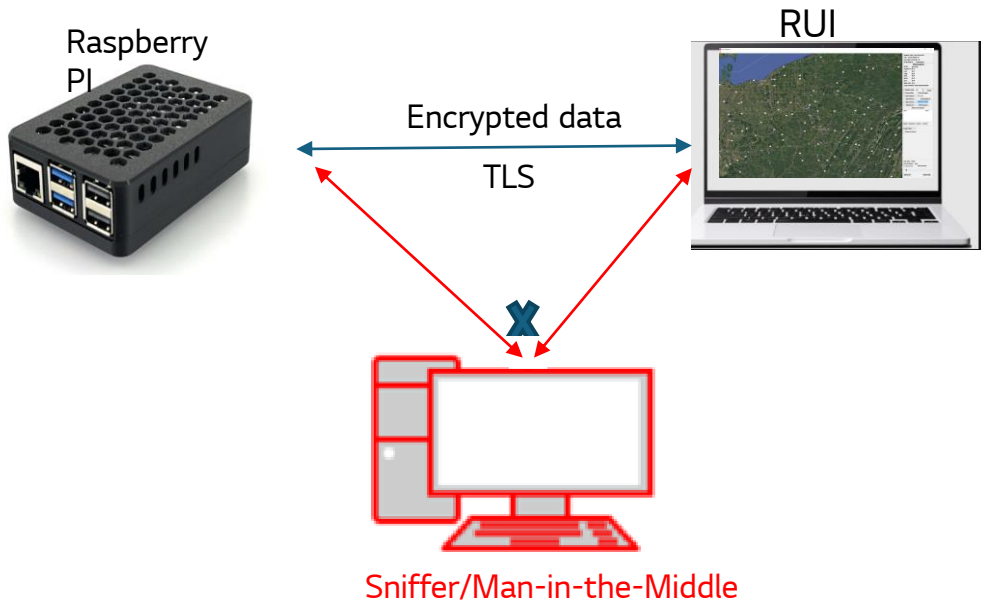
2nd step:

The victim thinks he is the operator and downloads the file and overwrites it.

Do you think this came from the operator?

TLS





Protocol: TCP (Transmission Control Protocol) is used for reliable, connection-oriented communication.
Socket: The Raspberry Pi acts as a server, listening for incoming TCP connections on a specific port (e.g., 30003 or 5001). The RUI (Windows PC) acts as a client, initiating a TCP connection to the Raspberry Pi's IP address and port.
Data Format: Raw ADS-B data is sent as plain text over the TCP connection. This means the data is **unencrypted** and can be intercepted and read by anyone with access to the network traffic.
Security: There is no encryption or authentication. Any device that can connect to the Raspberry Pi's IP address and port can receive the ADS-B data

How TLS/SSL Protects Against Sniffing and Manipulation:
Encryption: TLS encrypts the data being transmitted, making it unreadable to a sniffer. Even if the sniffer captures the packets, it won't be able to see the contents.
Authentication: TLS uses certificates to verify the identity of the server (and optionally the client). This prevents man-in-the-middle attacks where an attacker tries to impersonate the server.
Integrity Protection: TLS includes mechanisms to detect if the data has been tampered with during transit. If a sniffer modifies a packet, the TLS connection will be broken.

SCRM (A-SQUARE-Based Security Assessment)

Component : OpenSSL 3.5
Application Context : TLS channel for transmitting user ID/password during authentication

- Defined Role
  - Provides TLS 1.3-based encrypted communicatio n
  - Ensures confidentiality and integrity of ID/passw ord transmission
  - Supports secure user authentication
- Security
  - Utilizes verified cryptographic algorithms
  - Offers optional FIPS-compliant mode
  - Regularly releases patches for critical CVEs
- Maintainability and Community Activity
  - Maintained by the OpenSSL Project foundation
  - Active community (GitHub issues/PRs)
  - Timely security updates
  - Long-term sustainability ensured
- Release Strategy
  - No fixed release cycle; somewhat flexible
  - Long-Term Support (LTS) versions available
  - Emergency security patches provided when necessary
- Risk & Mitigation
  - Risks : Past CVEs, API incompatibility, irregular release sc hedule
  - Mitigations : Use of static analysis tools (e.g., Coverity, T rivy), automated monitoring of security updates
- Operational Trustworthiness
  - Security scanning tools integrated in CI/CD pipeline
  - Certificate and cipher suite validation
  - Integrity of TLS components maintained at runtime

* Identified risks are manageable and mitigated
* Meets all A-SQUARE evaluation criteria

Trust Decision : Accepted and safely integrated as a trusted component