Autonomous Rescue Drone System Complete Security Architecture

Executive Summary

This document synthesizes the complete security architecture for the autonomous rescue drone system, as detailed in the eight constituent specifications. The architecture is built on a defense-in-depth strategy, employing multiple, overlapping security layers to protect the system's critical assets—Flight Control, Mission Data, and Victim Privacy—even under active attack. The design directly addresses the highest-priority threats identified in the Threat Model, including remote command injection, GNSS spoofing, and firmware tampering.

1. Cohesive Security Framework

The architecture is not a collection of independent controls but an integrated framework where each component reinforces the others:

Cryptographic Core (Auth & Encryption): The use of HSM-stored keys for command signing and TLS 1.3 for communication forms the bedrock of trust for all data in transit.

Platform Integrity (Firmware & Physical): Secure boot and tamper detection ensure that the cryptographic core and the rest of the drone's software are running on a trusted hardware platform.

> **Resilience & Awareness (Redundancy, Anomaly Detection, Anti-Spoofing):** These layers ensure the system remains operational and can detect/respond to attacks that bypass the primary cryptographic and integrity controls.

2. Defense-in-Depth Against Critical Threats

Critical Threat	Layered Mitigations
Remote Command Injection	1. TLS 1.3 encrypted channel 2. ECDSA-signed commands 3. Nonce/Sequence replay protection 4. Multi-signature for critical commands
GNSS Spoofing	1. Multi-constellation/frequency GNSS 2. Galileo OSNMA authentication 3. INS/Visual Odometry cross-verification 4. Failover to GNSS-denied navigation
RF Jamming	1. Multi-band communication (LoRa, WiFi, Satellite) 2. Automatic failover based on link quality 3. Mesh networking 4. Autonomous operation during blackout
Firmware Tampering	1. Hardware-rooted secure boot 2. Signed firmware updates 3. Runtime integrity monitoring 4. Tamper detection and secure key storage

Threat Model & Risk Assessment

1. Assets to Protect

ASSET 1: Flight Control Authority

- **Description:** The ability to command drone movement, mission execution, and emergency actions
- **Value:** CRITICAL - Unauthorized control could crash drone, harm people, or disrupt rescue
- **Protection Requirements:** Authentication, authorization, integrity

ASSET 2: Telemetry & Sensor Data

- **Description:** Real-time position, video feeds, detection results, system health
- **Value:** HIGH - Contains operational details, victim locations (privacy-sensitive)
- **Protection Requirements:** Confidentiality, integrity, availability

ASSET 3: Mission Data

- **Description:** Search areas, waypoints, geofence boundaries, objectives
- **Value:** HIGH - Reveals operational plans and sensitive locations
- **Protection Requirements:** Confidentiality, integrity

ASSET 4: Cryptographic Keys & Certificates

- **Description:** Private keys for signing/decryption, session keys, certificates
- **Value:** CRITICAL - Compromise enables impersonation and unauthorized access
- **Protection Requirements:** Confidentiality, integrity, secure storage

ASSET 5: Firmware & Software

- **Description:** Flight control software, AI models, operating system

- **Value:** CRITICAL - Malicious firmware could create persistent backdoor
- **Protection Requirements:** Integrity, authenticity, trusted execution

ASSET 6: Physical Hardware

- **Description:** Drone airframe, sensors, communication modules, secure elements
- **Value:** HIGH - Physical access enables key extraction and hardware modification
- **Protection Requirements:** Tamper detection, secure facilities

2. Threat Actors

ACTOR 1: Nation-State Adversaries

- **Capabilities:** Advanced persistent threats, sophisticated tools, unlimited resources
- **Motivation:** Intelligence gathering, sabotage critical infrastructure
- **Likelihood:** LOW (rescue drones unlikely target, but possible in conflict zones)
- **Attack Vectors:** Supply chain compromise, zero-day exploits, GNSS spoofing

ACTOR 2: Terrorist Organizations

- **Capabilities:** Moderate technical skills, access to commercial tools, physical access
- **Motivation:** Disrupt rescue operations, cause casualties, create chaos
- **Likelihood:** MEDIUM (disaster zones can be targets)
- **Attack Vectors:** RF jamming, physical capture, improvised GPS spoofers

ACTOR 3: Hackers / Script Kiddies

- **Capabilities:** Publicly available tools, moderate skills, opportunistic
- **Motivation:** Challenge, notoriety, experimentation
- **Likelihood:** HIGH (drones are popular targets)
- **Attack Vectors:** Command injection, replay attacks, known vulnerabilities

ACTOR 4: Malicious Insiders

- **Capabilities:** Authorized access, knowledge of systems, physical presence
- **Motivation:** Sabotage, theft, coercion
- **Likelihood:** LOW (small team, vetting process)
- **Attack Vectors:** Credential abuse, firmware tampering, data exfiltration

ACTOR 5: Competitors / Corporate Espionage

- **Capabilities:** Moderate resources, technical expertise, surveillance tools
- **Motivation:** Steal intellectual property, competitive advantage
- **Likelihood:** LOW (competition scenario, but possible in commercial deployment)
- **Attack Vectors:** Eavesdropping, reverse engineering, social engineering

3. Threat Analysis Matrix

ID	Threat	Likelihood	Impact	Risk Score
T1	Remote Command Inject	High	Critical	**9/10 (Critical)**
T2	Man In The Middle Attack	High	High	**8/10 (High)**
T3	Replay Attack	Medium	Medium	**5/10 (Medium)**
T4	Key/Certificate Compromise	Low	Critical	**6/10 (Medium-High)**
T5	RF Jamming	High	Medium	**6/10 (Medium-High)**
T6	GNSS Spoofing	Medium	High	**6/10 (Medium-High)**
T7	Firmware Tampering	Low	Critical	**6/10 (Medium-High)**
T8	Physical Tampering	Medium	High	**6/10 (Medium-High)**
T9	Supply Chain Attack	Low	High	**6/10 (Medium-High)**
T10	Denial of Service	Medium	Medium	**5/10 (Medium)**

Authentication Specification

1. Introduction

This document specifies the authentication framework for the autonomous rescue drone system. It ensures that all commands, mission data updates, and access to sensitive telemetry are cryptographically verified, preventing unauthorized control and protecting the integrity of rescue operations. The system employs a public key infrastructure (PKI) with hardware security modules to achieve robust, scalable authentication.

2. Authentication Architecture Overview

The authentication strategy is based on a PKI model with mutual authentication between the Ground Control Station (GCS) and the drone fleet. This ensures that both ends of the communication link verify each other's identity before any data exchange.

- **Ground Control Station (GCS):** Acts as the command authority. It possesses a private key for signing commands.
- **Drone:** Possesses a unique public key certificate. It verifies the signature on all incoming commands from the GCS.
- **Mutual TLS (mTLS):** All communication channels utilize TLS 1.3 with mutual authentication, requiring both the GCS and the drone to present and validate X.509 certificates.

3. Cryptographic Command Signing & Verification

This is the primary defense against remote command injection attacks (T1 from Threat Model).

3.1. Command Format

Every command sent from the GCS to the drone must be a structured, signed JSON object:

```
{
  "command_id": "cmd_20251005_143022_001",
  "mission_id": "rescue_tunisia_earthquake_001",
  "timestamp": 1738687422,
  "nonce": "a3f59d8c7b2e1f4a6c8d9e0f1a2b3c4d",
  "sequence_num": 42,
  "command_type": "goto_waypoint",
  "parameters": {
    "lat": 36.8065,
    "lon": 10.1815,
    "alt": 50
  },
  "signature": "3045022100f3a2...b8c4"
}
```

Field	Description	Security Purpose
command_id	Unique identifier for the command	Logging, non-repudiation
nonce	Cryptographically secure random number (32-byte)	Replay attack protection
sequence_num	Monotonically increasing number	Replay and out-of-order command protection
signature	Digital signature of all other fields	Authenticity and integrity verification

3.2. Verification Process on Drone

Upon receipt, the drone performs the following steps:

1. Parse the incoming message
2. **Extract Signature:** Isolate the signature field from the concatenated message fields

3. **Compute Hash:** Calculate SHA-256(concatenated_fields)
4. **Verify Signature:** Perform ECDSA_Verify(H, signature, public_key_ground_station)
5. **Check Nonce & Sequence:** Validate the nonce has not been used recently and the sequence number is higher than the last accepted command
6. **Execute or Reject:** If all checks pass, execute the command. If any fail, reject the command and log a security event

4. Mutual Authentication & Secure Key Lifecycle

4.1. Mutual Certificate-Based Authentication

All communication channels (Satellite, LTE) use TLS 1.3 with mutual authentication.

- **Drone and GCS Certificates:** Both parties present X.509 certificates signed by a private, internal Certificate Authority (CA) dedicated to the rescue operation
- **Certificate Validation:** Validation includes checks for CA signature, expiration, revocation status (via CRL/OCSP), and expected Common Name

4.2. Secure Key Storage & Lifecycle

- **Hardware Security Modules (HSMs):** All private keys (for GCS signing and drone certificates) are stored in tamper-resistant HSMs (e.g., YubiKey for GCS, Microchip ATECC608A on the drone). Private keys are never exposed to the application OS
- **Key Rotation:**
  - Session Keys: Derived fresh for each TLS session (Perfect Forward Secrecy)
  - Device Certificates: Rotated annually
  - CA Root Key: Rotated every 5 years via an offline ceremony
- **Revocation:** A Certificate Revocation List (CRL) is maintained and distributed. Drones check the CRL before accepting connections from a GCS

5. Multi-Factor Authentication for Critical Operations

For high-risk commands that could compromise the mission or drone safety, a multi-signature scheme is required.

Commands requiring MFA:

- format_storage
- disable_safety_limits
- modify_geofence
- firmware_update

Process:

1. **Operator Signature:** The primary operator signs the command using their HSM
2. **Security Officer Signature:** A separate security officer independently signs the same command payload using a different HSM
3. **Verification:** The drone verifies both signatures before execution

Encryption Specification

1. Introduction

This document defines the encryption standards for the autonomous rescue drone system. The primary objective is to ensure end-to-end confidentiality and integrity of all data in transit between the drone and the Ground Control Station (GCS), protecting against eavesdropping and tampering, which are critical for mission success and victim privacy.

2. Encryption Architecture & Standards

The system employs a layered encryption approach, securing both the command channel and the data links.

- **Protocol:** TLS 1.3
- **Cipher Suite:** TLS_CHACHA20_POLY1305_SHA256
- **Key Exchange:** X25519 (Elliptic Curve Diffie-Hellman over Curve25519)

Rationale for Cipher Suite: ChaCha20 is a stream cipher known for high performance on embedded ARM processors (like those found in drones), and Poly1305 provides strong message authentication. This combination is efficient and secure for resource-constrained devices.

3. End-to-End Encrypted Communication

3.1. TLS 1.3 Handshake with Mutual Authentication

The connection establishment follows a standard TLS 1.3 handshake, modified for mutual authentication:

1. **ClientHello (Drone → GCS):** Drone sends supported ciphers and its ephemeral public key

2. **ServerHello (GCS → Drone):** GCS selects cipher suite, sends its ephemeral public key, and its certificate
3. **Certificate Verification (Drone → GCS):** Drone sends its certificate and a signature verifying the handshake
4. **Key Derivation:** Both parties use ECDH with their private keys and the other's public key to compute a shared secret
5. **Session Keys:** Session encryption keys are derived using HKDF from the shared secret

This process provides Perfect Forward Secrecy (PFS), ensuring that a compromise of the long-term private keys does not allow decryption of past recorded sessions.

3.2. Encrypted Message Format

Once the session is established, all messages are encrypted with the following structure:

- **Nonce:** A unique 96-bit value (never reused with the same session key)
- **Ciphertext:** ChaCha20_Encrypt(plaintext, session_key, nonce)
- **Authentication Tag:** Poly1305_MAC(ciphertext, session_key)

This format guarantees confidentiality (ciphertext) and integrity (authentication tag).

4. Data Classification & Encryption Mapping

Different types of data have varying sensitivity levels and are protected accordingly.

Data Type	Sensitivity	Encryption
Commands & Mission Updates	CRITICAL (Integrity)	TLS 1.3 + Cryptographic Signing
Real-Time Video Feeds	HIGH (Confidentiality)	TLS 1.3 (All streams)
Telemetry (Position, Health)	HIGH	TLS 1.3

Data Type	Sensitivity	Encryption
Victim Detection Data	HIGH (Privacy Sensitivity)	TLS 1.3

5. Key Management

- **Session Keys:** Ephemeral, derived during TLS handshake and destroyed after session termination
- **Long-term Private Keys:** Stored in Hardware Security Modules (HSMs) as specified in the Authentication Spec. Never exposed to system memory
- **Public Key Certificates:** Stored in drone firmware, verified at boot via secure boot process

Firmware Security Specification

1. Introduction

This document outlines the security measures to protect the drone's firmware and software from tampering, unauthorized modification, and execution of malicious code. Ensuring firmware integrity is critical, as a compromise could lead to a persistent backdoor, total system failure, or malicious behavior during rescue operations.

2. Secure Boot & Chain of Trust

A cryptographically verified chain of trust ensures that only authorized software is executed on the drone from power-on.

- **Immutable Bootloader:** The initial bootloader is stored in write-protected flash or ROM. It contains a public key
- **Verification Process:**
  - On power-up, the bootloader verifies the digital signature of the next-stage bootloader or operating system kernel using its embedded public key
  - This process repeats for each subsequent stage (e.g., kernel verifies root filesystem, drone application)
  - Any invalid signature halts the boot process, leaving the drone in a safe state

This process prevents the execution of any unsigned or tampered firmware.

3. Signed Firmware Updates

All firmware updates, whether for the flight controller, companion computer, or radio modules, are delivered as signed packages to prevent supply-chain attacks and malicious updates.

- **Update Package Format:** The update is a compressed archive containing the firmware image, a manifest file with version information, and a cryptographic signature
- **Verification on Drone:** Before applying an update, the drone:
  - Verifies the signature of the update package using the vendor's public key (stored in a secure element)
  - Checks the version number against the current firmware to enforce rollback protection
  - Only proceeds with the update if all checks pass

4. Runtime Integrity Monitoring

To detect runtime tampering, the system employs continuous integrity checks.

- **Measured Boot:** The boot process measures (hashes) each component as it loads and extends these measurements into a secure element (TPM)
- **Remote Attestation:** The GCS can challenge the drone to report its current measurements. The drone responds with the hashes signed by the TPM, allowing the GCS to verify its software state
- **File Integrity Monitoring (FIM):** Critical system files and configurations are monitored for changes using checksums

5. Hardware Root of Trust

The security measures above rely on a Hardware Root of Trust.

- **Secure Element:** A chip like the Microchip ATECC608A is used to:
  - Securely store the public keys for secure boot and firmware verification
  - Perform cryptographic operations in isolation
  - Provide a unique device identity
- **Trusted Platform Module (TPM):** Used for storing runtime integrity measurements and key generation

Redundancy and Failover Specification

1. Introduction

This document specifies the redundancy and failover strategies for the drone's communication and navigation systems. The primary goal is to ensure high availability and operational continuity even in the face of targeted attacks (like jamming) or component failures, which is paramount in life-critical rescue missions.

2. Multi-Band Communication Failover

To mitigate RF Jamming (T5), the drone is equipped with multiple, diverse communication systems. An intelligent link manager continuously monitors all channels and automatically fails over to the best available one.

2.1. Communication Stack

- **Primary Link (High Bandwidth):** 2.4 GHz WiFi for video streaming and complex data
- **Secondary Link (Long Range):** 900 MHz LoRa for basic commands and telemetry
- **Tertiary Link (Alternative High Bandwidth):** 5.8 GHz WiFi

2.2. Failover Protocol

1. **Continuous Monitoring:** The link manager monitors Signal-to-Noise Ratio (SNR), packet loss, and latency on all channels
2. **Detection:** If the primary channel degrades (packet loss >30% for 3 seconds), a jamming event is detected
3. **Switch:** The system automatically switches to the next-best available channel
4. **Notification:** The GCS is alerted: "Jamming detected on 2.4GHz, switched to 900MHz LoRa."

5. **Escalation:** If all terrestrial links are lost, the system attempts to establish a satellite connection

3. Redundant Navigation Systems

To mitigate GNSS Spoofing (T6) and failure, the drone employs sensor fusion and alternative positioning methods.

- **Primary:** Multi-constellation GNSS (GPS, Galileo, GLONASS)
- **Secondary:** Inertial Navigation System (INS) fused with Visual Odometry (VO)
- **Failover Behavior:** If GNSS spoofing is detected (discrepancy between GNSS and INS/VO position), the drone switches to INS/VO-based navigation and alerts the operator

4. Mesh Networking (Drone-to-Drone Relay)

For swarm operations, a LoRa-based mesh network provides communication redundancy.

- **Protocol:** AODV (Ad-hoc On-Demand Distance Vector) or BATMAN
- **Benefit:** If a drone's direct link to the GCS is jammed, it can relay its data through other drones in the mesh, creating a resilient, self-healing network

5. Autonomous Fail-Safe Behaviors

When all communication is lost, the drone enters a pre-programmed, intelligent autonomous mode.

- **If battery >50%:** Continue the current mission segment (e.g., complete the search pattern), buffering all data locally
- **If battery <50%:** Execute a Return-to-Home procedure via a pre-planned safe route
- **On reconnection:** The drone immediately transmits all buffered data to the GCS

Anomaly Detection Specification

1. Introduction

This document details the anomaly detection system designed to identify unusual patterns in drone behavior, telemetry, and commands. This serves as an advanced line of defense, potentially detecting novel attacks or system faults that other security measures might miss.

2. Architectural Overview

The system employs a two-tiered approach:

- **On-Drone Real-Time Detection:** Lightweight models for immediate reaction to critical anomalies (e.g., command flooding)
- **Ground-Based Deep Analysis:** More complex models on the GCS for analyzing aggregated telemetry and video data to identify subtle or complex threats

3. Telemetry & Behavioral Anomaly Detection

Using unsupervised learning models like Isolation Forest or Local Outlier Factor (LOF), the system establishes a baseline of normal drone behavior.

Features Monitored:

- Flight dynamics (unexpected acceleration, attitude)
- Power consumption profiles
- Sensor reading correlations (e.g., IMU vs. visual odometry)
- Command rate and type frequency

A significant deviation from the baseline, indicated by a high anomaly score, triggers a security alert.

4. Command & Control Anomaly Detection

This layer analyzes the stream of commands for suspicious patterns that might indicate a compromised GCS or an attacker testing the system.

Detection Rules:

- **Rate Limiting:** Enforce a maximum of 10 commands per second from any source
- **Impossible Commands:** Flag commands that are physically impossible (e.g., instant extreme velocity change)
- **Geofence Violation Attempts:** Detect repeated commands to fly outside the mission's geofence
- **Sequence Analysis:** Identify illogical command sequences that deviate from standard mission protocols

5. AI Model for Victim Detection Anomalies

The primary AI model for victim detection is also monitored for anomalies.

- **Contextual Inconsistency:** Flagging detections that are highly improbable for the environment (e.g., a "person" detected in the middle of a lake)
- **Performance Drift:** Monitoring the model's confidence scores for significant drops, which could indicate adversarial attacks or environmental conditions the model wasn't trained for

6. Response to Detected Anomalies

The response is proportional to the anomaly's severity and confidence.

- **Low Severity:** Log the event for later analysis by security personnel
- **Medium Severity:** Alert the operator with a request for confirmation
- **High Severity:** Trigger an immediate autonomous fail-safe action, such as hovering, aborting the current task, or initiating a return-to-home procedure

Anti-Spoofing Specification

1. Introduction

This document describes the measures to protect the drone from spoofing attacks, where an attacker forges signals to deceive the system. The two primary spoofing threats addressed are GNSS (GPS) Spoofing and Communication/Sensor Spoofing.

2. GNSS Anti-Spoofing Measures

A multi-layered approach is used to ensure the drone cannot be tricked by fake satellite signals.

2.1. Multi-Constellation & Multi-Frequency Receivers

The drone uses a GNSS receiver (e.g., u-blox ZED-F9P) that can simultaneously process signals from GPS (USA), Galileo (EU), GLONASS (Russia), and BeiDou (China).

It also uses multiple frequency bands (L1, L2, L5). An attacker must spoof all constellations and frequencies simultaneously, which is highly complex and requires expensive equipment.

2.2. Signal Authentication

Galileo OSNMA: The drone prioritizes Galileo signals when possible to utilize the Open Service Navigation Message Authentication (OSNMA) feature, which cryptographically authenticates the navigation data, making it impossible to generate valid fake signals without secret keys.

2.3. Cross-Checking with Independent Sensors

The GNSS position is continuously cross-verified with other sensors:

- **Inertial Navigation System (INS):** Uses an IMU for dead reckoning

- **Visual Odometry (VO):** Uses camera data to estimate movement relative to the ground
- **Spoofing Detection:** A significant and sustained discrepancy between the GNSS position and the INS/VO position indicates a likely spoofing attack

3. Communication & Sensor Anti-Spoofing

- **Cryptographic Authentication:** All commands and critical telemetry are cryptographically signed as defined in the Authentication Spec, preventing an attacker from spoofing legitimate commands
- **Sensor Data Validation:** Data from non-secure sensors (e.g., public ADS-B signals) is treated as untrusted and is not used for safety-critical decisions without validation

4. Fail-Safe Behavior on Spoofing Detection

Upon detecting a high-confidence spoofing event:

1. Immediately alert the GCS: "GNSS SPOOFING DETECTED"
2. Switch to GNSS-denied navigation mode (INS + Visual Odometry)
3. If alternative navigation is unreliable, execute a pre-defined safe behavior (hover, return to last known valid point)

Physical Security Specification

1. Introduction

This document outlines the physical security controls for the drone hardware and its operating environment. The goal is to prevent unauthorized physical access, tampering, theft, or damage that could compromise the drone, its data, or the cryptographic keys within it.

2. Drone Hardware Tamper Protection

Physical access to the drone could allow key extraction or hardware modification.

- **Tamper-Evident Enclosure:** The drone's chassis is sealed with security screws (Torx, Spanner) and tamper-evident seals. Any attempt to open the casing will leave visible evidence
- **Tamper Detection Switches:** Internal switches or a conductive mesh detect when the enclosure is opened, triggering an immediate secure wipe of all configurable memory and cryptographic keys stored in the secure element
- **Secure Debugging Interfaces:** All debug ports (JTAG, SWD) are disabled in production firmware via fuse bits, preventing easy firmware extraction or manipulation

3. Secure Key Storage

As defined in the Authentication Spec, all private keys are stored within a Hardware Security Module (HSM) or Secure Element (e.g., Microchip ATECC608A). These components are designed to be tamper-resistant, with active shielding that zeroizes the key storage upon detection of physical probing.

4. Supply Chain Security

To mitigate the risk of a malicious component being inserted during manufacturing (Supply Chain Attack, T9):

- **Component Vetting:** Source components from reputable, vetted suppliers
- **Binary Analysis:** Perform static analysis on third-party firmware and libraries
- **Hardware Authenticity:** Use components with factory-programmed cryptographic keys that the host processor can challenge for authenticity

5. Operational Base Security

The temporary field base where drones are deployed and stored must be secured.

- **Access Control:** Use RFID-enabled access control systems for storage areas
- **Surveillance:** Deploy security cameras to monitor drones and ground station equipment
- **Transportation:** Use locked, secure cases for transporting drones to and from the operation site

Residual Risk Assessment

After implementing the layered mitigation strategies detailed in this document, the overall risk to the autonomous rescue drone system is significantly reduced. The following table summarizes the residual risk for each identified threat, demonstrating that the system architecture is robust and resilient.

ID	Threat	Initial Risk Score	Residual Risk Score
T1	Remote Command Inject	**Critical 9/10**	**Low 2/10**
T2	Man In The Middle Attack	**High 8/10**	**Low 2/10**
T3	Replay Attack	**Medium 5/10**	**Very Low 1/10**
T4	Key/Certificate Compromise	**Medium-High 6/10**	**Low 2/10**
T5	RF Jamming	**Medium-High 6/10**	**Low 2/10**
T6	GNSS Spoofing	**Medium-High 6/10**	**Low 2/10**
T7	Firmware Tampering	**Medium-High 6/10**	**Low-Med 3/10**
T8	Physical Tampering	**Medium-High 6/10**	**Low-Med 3/10**
T9	Supply Chain Attack	**Medium-High 6/10**	**Low 2/10**
T10	Denial of Service	**Medium 5/10**	**Low 2/10**

Conclusion

This security architecture presents a robust, resilient, and comprehensive solution for securing autonomous rescue drones. By systematically addressing threats through cryptographic assurance, platform integrity, and operational resilience, it justifies every security decision with a clear rationale tied to risk mitigation.

The architecture employs a defense-in-depth strategy with multiple overlapping security layers:

- **Cryptographic Foundation:** TLS 1.3 encryption, ECDSA command signing, and HSM-based key storage provide the bedrock of trust
- **Platform Integrity:** Secure boot, signed firmware updates, and runtime integrity monitoring ensure only trusted code executes
- **Operational Resilience:** Multi-band communication failover, redundant navigation systems, and autonomous fail-safe behaviors maintain mission continuity under attack
- **Advanced Threat Detection:** Anomaly detection and anti-spoofing measures identify novel attacks and sophisticated deception attempts
- **Physical Security:** Tamper-evident enclosures, secure key storage, and operational security controls protect against physical compromise

This architecture not only meets the immediate needs of the competition but also provides a strong foundation for the secure deployment of autonomous systems in real-world, critical rescue operations. The systematic reduction of risk scores from Critical/High levels to Low levels demonstrates the effectiveness of the layered mitigation strategies.

The system is designed to remain operational and secure even when individual components are compromised, ensuring that rescue missions can continue and lives can be saved even in hostile environments.