MAVLink Security Hardening for Production Deployments

PX4 v1.17

MAVLink is an open communication protocol designed for lightweight, low-latency communication between drones and ground stations. By default, all MAVLink messages are unauthenticated. This is intentional for development and testing, but **production deployments must enable** message signing to prevent unauthorized access.

WARNING

Without message signing enabled, any device that can send MAVLink messages to the vehicle (via radio, network, or serial) can execute any command, including shell access, file operations, parameter changes, mission uploads, arming, and flight termination.

What Is at Risk

When MAVLink signing is not enabled, an attacker within communication range can:

Capability	MAVLink mechanism
Execute shell commands	`SERIAL_CONTROL` with `SERIAL_CONTROL_DEV_SHELL`
Read, write, or delete files	MAVLink FTP protocol
Change any flight parameter	`PARAM_SET` / `PARAM_EXT_SET`
Upload or overwrite missions	Mission protocol
Arm or disarm motors	`MAV_CMD_COMPONENT_ARM_DISARM`

Capability	MAVLink mechanism
Terminate flight (crash)	`MAV_CMD_DO_FLIGHTTERMINATION`
Trigger emergency landing	Spoofed `BATTERY_STATUS`
Reboot the vehicle	`MAV_CMD_PREFLIGHT_REBOOT_SHUTDOWN`

All of these are standard MAVLink capabilities used by ground control stations. Without signing, there is no distinction between a legitimate GCS and an unauthorized sender.

Hardening Checklist

1. Enable Message Signing

Message signing provides cryptographic authentication for all MAVLink communication. See Message Signing for full details.

Steps:

1. Connect the vehicle via **USB** (key provisioning only works over USB).
2. Provision a 32-byte secret key using the SETUP_SIGNING message.
3. Set MAV_SIGN_CFG to **1** (signing enabled on all links except USB) or **2** (signing on all links including USB).
4. Provision the same key on all ground control stations and companion computers that need to communicate with the vehicle.
5. Verify that unsigned messages from unknown sources are rejected.

> **INFO**
>
> `MAV_SIGN_CFG=1` is recommended for most deployments. This enforces signing on telemetry radios and network links while allowing unsigned access over USB for maintenance. USB connections require physical access to the vehicle, which provides equivalent security to physical key access.

2. Secure Physical Access

- Protect access to the SD card. The signing key is stored at `/mavlink/mavlink-signing-key.bin` and can be read or removed by anyone with physical access.
- USB connections bypass signing when `MAV_SIGN_CFG=1`. Ensure USB ports are not exposed in deployed configurations.

3. Secure Network Links

- Do not expose MAVLink UDP/TCP ports to untrusted networks or the internet.
- Place MAVLink communication links behind firewalls or VPNs.
- Segment MAVLink networks from business or public networks.
- When using companion computers, audit which network interfaces MAVLink is bound to.

4. Understand the Limitations

- **No encryption**: Message signing provides authentication and integrity, but messages are sent in plaintext. An eavesdropper can read telemetry and commands but cannot forge them.
- **Allowlisted messages**: A small set of safety-critical messages (RADIO_STATUS, ADSB_VEHICLE, COLLISION) are always accepted unsigned.
- **Key management**: There is no automatic key rotation. Keys must be reprovisioned manually via USB if compromised.

Integrator Responsibility

PX4 is open-source flight controller firmware used by manufacturers and system integrators to build commercial and custom drone platforms.

Securing the communication links for a specific deployment is the responsibility of the system integrator. This includes:

- Choosing appropriate radio hardware and link security
- Enabling and managing MAVLink message signing
- Restricting network access to MAVLink interfaces

- Applying firmware updates that address security issues
- Evaluating whether the default configuration meets the security requirements of the target application

PX4 provides the tools for securing MAVLink communication. Integrators must enable and configure them for their deployment context.

Edit on GitHub

Previous page
Message Signing

Next page
Protocols/Microservices