# 🧩 Full Threat Modeling Workflow for Embedded + Multi-Cloud Products

**🔰 Phase 1: Preparation**

1. **Define Goals and Scope**
   * What are you protecting? (Device, firmware, cloud APIs, connectivity, data, updates, etc.)
   * Why are you doing threat modeling? (Compliance, security-by-design, new release)
   * Define the **attack surface**: Physical, Network, Firmware, Cloud, OTA.
2. **Identify Stakeholders**
   * Involve cross-functional teams:
     + Firmware Engineers
     + DevOps / Cloud Architects
     + Cybersecurity
     + Product Management
     + Compliance (esp. for regulated industries)
3. **Document System Overview**
   * **Block diagrams** of device architecture
   * **Data flow diagrams (DFDs)** between device ↔ edge ↔ cloud
   * Include:
     + Sensors
     + Actuators
     + Communication modules (WiFi, BLE, LoRa, etc.)
     + Microcontroller/SoC
     + Multi-cloud backends (AWS, Azure, GCP, etc.)
     + CI/CD & OTA pipeline
     + Companion apps/web portals

**🏗️ Phase 2: Model the System**

1. **Create a Detailed Data Flow Diagram (DFD)**
   * Include:
     + External entities: user, attacker, OTA provider
     + Processes: cloud functions, device logic, OTA agent
     + Data stores: device memory, cloud databases
     + Trust boundaries: between local device, cloud VPCs, and 3rd-party APIs
2. **Identify Trust Boundaries**
   * Physical (device boundary)
   * Firmware/OS sandboxing
   * Cloud zones (e.g., AWS VPC to Azure function boundary)
   * OTA and DevOps flows
   * Identity federation boundaries (SAML, IAM, device identity)
3. **Define Assets**
   * Firmware/bootloader
   * Credentials, keys, tokens
   * Sensor data
   * Actuator control channels
   * Device identity
   * Telemetry
   * Cloud functions, APIs, containers

**🔐 Phase 3: Identify Threats**

1. **Apply Threat Modeling Methodologies**
   * **STRIDE** (for software and cloud layers)
   * **LINDDUN** (for privacy-focused embedded or regulated environments)
   * **PASTA or Attack Trees** (for deep risk-centric views)
   * Use OWASP Threat Dragon, Microsoft TMT, IriusRisk, or a custom board.
2. **Walk Through Each STRIDE Category**
   * For every component & data flow, ask:
     + **S**poofing: Can identities be faked?
     + **T**ampering: Can the firmware/data be modified?
     + **R**epudiation: Are logs/audits present?
     + **I**nformation Disclosure: Are sensitive data exposed?
     + **D**enial of Service: Can this be taken offline?
     + **E**scalation of Privilege: Can low-priv users gain root?
3. **Include Multi-Cloud Specific Threats**
   * Misconfigurations across clouds
   * Inconsistent IAM roles/policies
   * Lack of visibility/logs across platforms
   * Data exfiltration via cross-cloud routing
   * Vendor lock-in exploits
4. **Include Embedded-Specific Threats**
   * JTAG/UART debug port access
   * Firmware downgrade attacks
   * Physical tampering or side-channel attacks
   * Radio jamming, signal injection
   * Power glitch/replay attacks
   * Secure boot bypass

**📉 Phase 4: Analyze and Prioritize Threats**

1. **Rate Threats**
   * Use **DREAD**, **CVSS**, or a custom risk formula
   * Evaluate:
     + Impact
     + Likelihood
     + Exploitability
     + Discoverability
   * Score and sort into Critical / High / Medium / Low
2. **Map Threats to Assets**
   * Build a threat-asset matrix
3. **Assess Attack Paths**
   * Use attack trees or kill chains to visualize threat propagation
   * Model real-world attacker scenarios

**🛡️ Phase 5: Mitigate and Validate**

1. **Propose Security Controls**
   * Mitigation techniques by layer:
     + Device: Secure boot, HW-backed crypto, anti-rollback
     + Comm: mTLS, VPNs, token validation
     + Cloud: IAM policies, segmentation, container hardening
     + OTA: Signed updates, rollback protection
   * Cross-cloud: CSPM (Cloud Security Posture Mgmt), multi-cloud firewalls, unified IAM
2. **Validate Controls**
   * Penetration testing (cloud and firmware)
   * Fuzzing embedded inputs
   * Code reviews for OTA/firmware components
   * Threat model validation workshops

**🔁 Phase 6: Document and Maintain**

1. **Generate Reports**
   * Architecture diagram
   * Identified threats (with STRIDE tags)
   * Risk scoring matrix
   * Mitigation map
2. **Share with Teams**
   * Add to product documentation
   * Review during sprint planning / release reviews
3. **Revisit Threat Model Periodically**
   * When:
     + Architecture changes
     + Adding cloud providers
     + Firmware version changes
     + Regulatory changes

**🎯 Bonus: Automation + Tools**

| **Area** | **Recommended Tool(s)** |
| --- | --- |
| Diagramming | Draw.io, Threat Dragon, Lucidchart |
| Firmware Analysis | Ghidra, Binwalk, Frida, QEMU |
| Cloud Threats | Wiz, Orca Security, Prowler, ScoutSuite |
| CI/CD Security | GitHub Advanced Security, Prisma Cloud |
| Risk Management | IriusRisk, RiskLens |

