FRONT /> CODE

ROBOTICS & CYBERSECURITY

robotics engineers

OT/ICS practitioners

security engineers

FEB 2026

Robotics and Cybersecurity
Engineering Trust in Autonomous and Networked Machines

An extensive survey of the modern threat landscape, standards, and practical engineering controls

Focus: industrial robots, collaborative robots, mobile robots, medical robots, and cloud/fleet robotics

Audience: robotics engineers, security engineers, OT/ICS practitioners, product teams, and technical leadership

Front /> Code

Feb 2026

Executive Summary

Robotics has shifted from isolated electromechanical equipment to highly connected cyber-physical systems (CPS). Modern robots embed general-purpose operating systems, third-party libraries, middleware (often publish/subscribe), remote service channels, and cloud-based fleet management. Connectivity delivers productivity gains, but it also means the robot is now part of the organization's cyber attack surface and part of its operational risk calculus.

Cybersecurity and robotics are linked by a single hard reality: a successful cyberattack on a robot is not just "data loss." It can become physical disruption, safety hazards, operational downtime, quality defects, IP theft, or loss of regulatory compliance. The relationship is therefore bidirectional: robotics increases cyber risk exposure, and cybersecurity constraints increasingly shape robotics architectures (network topology, update mechanisms, safety functions, and even mechanical design choices).

Recent industry data highlights why the topic is urgent. OT/ICS survey findings summarized by Dragos for the 2025 SANS State of ICS/OT Cybersecurity Survey report that nearly 50% of incidents were detected within 24 hours and 60% were contained within 48 hours of detection [4]. Yet the same summary reports that only 12.6% of organizations claim full visibility across the full ICS cyber kill chain—from initial IT compromise through potential impacts on control systems [4]. These gaps matter because adversaries do not stop at the IT/OT boundary; once a foothold exists in IT, attackers can pivot into OT environments where robots and control systems operate.

At the enterprise level, breach reporting shows that supply chain and third-party relationships are now major drivers of risk. Verizon's 2025 DBIR Executive Summary reports analysis of 22,052 security incidents and 12,195 confirmed breaches [5]. The 2025 DBIR also highlights a sharp increase in third-party involvement in breaches (reported as 30%) [5]. In robotics, the dependency graph is often unusually large—robot OEM firmware, PLCs, safety controllers, edge gateways, vision stacks, AI models, and integrator code—so supply chain and third-party risk management must be treated as a design requirement, not an afterthought.

This article provides a deep technical survey of the robotics attack surface, threat models, and defensive controls; it then connects them to OT/ICS standards (ISA/IEC 62443), guidance (NIST SP 800-82), and product regulations (EU Cyber Resilience Act). It concludes with a practical roadmap that helps teams move from ad-hoc controls to a repeatable, auditable security program without compromising real-time performance or safety requirements.

1. Why Cybersecurity Is Now a Core Robotics Discipline

For much of industrial history, robots were engineered under a "closed world" assumption. Controllers were deployed on segmented factory networks, programmed by specialists on-site, and rarely connected to enterprise IT—let alone the public internet. That assumption no longer holds.

Three macro-shifts explain why cybersecurity has become integral to robotics engineering:

(1) IT/OT convergence and remote operations. Manufacturing and logistics organizations want central visibility, predictive maintenance, remote diagnostics, and rapid reconfiguration. This drives connectivity between robot networks and enterprise services, often through gateways, historians, MES/ERP integrations, VPNs, and remote vendor access.

(2) Software-defined robotics. The capabilities of robots are increasingly differentiated by software—perception, planning, optimization, and fleet behavior—rather than by mechanical design alone. That "software gravity" increases dependency on open-source stacks and third-party components.

(3) Expansion of robotics beyond factories. Robots now operate in hospitals, warehouses, public spaces, and homes. These environments introduce new adversaries, weaker physical controls, and higher exposure to consumer and internet-facing networks.

A useful framing is that a robot is "a computer that can move," and movement changes the stakes. When confidentiality is compromised, organizations worry about IP theft or privacy. When integrity is compromised, a robot may produce incorrect motion, incorrect manipulation, or incorrect decisions. When availability is compromised, production or logistics may stop, causing measurable downtime and safety knock-on effects. In short: CIA impacts become CPS impacts.

OT security survey data shows that organizations are making progress in detection and containment, but visibility gaps remain deepest where physical consequences become most severe. Dragos' summary reports that while many organizations have OT-specific detection capabilities, visibility drops sharply as one moves deeper into industrial control layers, and only 12.6% report full visibility across the ICS cyber kill chain [4]. Robotics systems often sit at supervisory and control layers (Purdue Levels 2–3) and bridge to enterprise IT through gateways; weak visibility and segmentation at those layers directly increases operational risk.

From a governance standpoint, this is pushing robotics cybersecurity from "nice-to-have" to "duty-of-care." Safety standards are beginning to acknowledge cybersecurity as safety-relevant, and product regulations are increasingly mandating secure-by-design practices for connected products.

2. Robotics as a Cyber-Physical System: Architectural Foundations

To understand the relationship between robotics and cybersecurity, it helps to decompose a robot system into layers and interfaces. While robot types differ (industrial arms, AMRs, drones, surgical robots), many share a common architecture:

• Hardware layer: sensors (cameras, LiDAR, force/torque, encoders), actuators (motors, servos, grippers), power systems.

• Real-time control layer: motor control loops, safety PLCs, motion control firmware, deterministic fieldbuses.

• Compute layer: embedded computers (x86/ARM), GPU/accelerators, RTOS or Linux, container runtimes.

• Middleware layer: message passing, discovery, pub/sub topics, time synchronization, distributed coordination.

• Application layer: perception, SLAM, navigation, manipulation, task planning, HMI, diagnostics.

• Management layer: updates, telemetry, fleet management, logging, remote support, identity and access management.

• Integration layer: MES/WMS/ERP, SCADA, building management, identity providers, SIEM/SOC tooling, cloud services.

Each layer introduces security objectives and constraints. The control loop layer needs determinism and low latency; encryption and deep inspection must be engineered so they do not break real-time requirements. The management layer demands identity, auditing, and safe update processes. The integration layer introduces trust boundaries and supply chain risk.

A key observation is that robotics architecture is distributed. Multiple computers and controllers cooperate across multiple networks: the robot's internal bus, the cell network, the OT network, and sometimes an enterprise or cloud network. The security model must therefore be explicit about trust boundaries and failure modes: what happens if the robot loses access to the cloud? What if identity services are unavailable? What if time synchronization is attacked? What if a safety controller receives inconsistent commands from a compromised HMI? The design must specify safe degradation modes, not just "normal operation."

NIST describes operational technology (OT) as programmable devices that interact with the physical environment, including industrial control systems, building automation, transportation systems, and other systems that detect or cause direct physical changes [1]. Robots fit squarely in this definition. That means robotics cybersecurity inherits the OT "design triangle": security must be balanced against safety, reliability, and availability constraints.

3. The Robotics Attack Surface: A Structured Map

Robotics attack surface analysis benefits from being explicit and exhaustive. In practice, most robot compromises occur through the same broad classes of weaknesses seen in IT and OT—credential exposure, insecure remote access, unpatched components, weak segmentation—yet the downstream consequences may be physical.

Below is a structured map of common robotics attack surfaces and why they matter.

Robotics surface area	Typical weaknesses	Operational impact if exploited
Robot controller & firmware	Insecure services; weak auth; outdated components; unsafe default configs; inadequate signing/secure boot	Loss of control, unsafe motion, cell downtime, lateral movement in OT network
Programming workstations / offline programming tools	File parsing flaws; path traversal; macro/plug-in abuse; poor code signing	Malicious programs delivered into controller; propagation across robot fleet
Middleware & message bus (e.g., pub/sub)	Unauthenticated discovery; plaintext topics; weak authorization model; insecure time sync	Spoofed sensor/command messages; degraded autonomy; hidden manipulation
Edge gateways / integration servers	Bridging IT and OT; exposed APIs; default passwords; remote access misconfiguration	Pivot from IT to OT; data exfiltration; ransomware propagation
Fleet management & cloud services	Over-privileged tokens; multi-tenant isolation failures; API abuse; third-party dependencies	Fleet-wide disruption; coordinated misbehavior; mass downtime
Safety systems & interlocks	Assumed-trusted signals; inadequate integrity checks; engineering workstations unprotected	Bypass of safety function; unsafe continuation; regulatory exposure
Sensors & perception pipelines	Spoofing; calibration tampering; compromised drivers; weak integrity on sensor streams	Navigation/manipulation errors; quality defects; safety near-misses
Update channels & software supply chain	Unsigned updates; incomplete SBOM; vulnerable libraries; compromised vendors	Persistent compromise; delayed remediation; systemic risk across deployments
Human-machine interfaces (HMI) & mobile apps	Weak auth; shared accounts; insecure Wi-Fi; phishing	Unauthorized commands; loss of accountability; misconfiguration

Attack surface is not just “what ports are open.” It includes operational workflows: USB sticks used for program transfer, laptops that connect to both plant networks and public Wi-Fi, vendor remote support sessions, and ad-hoc credential sharing to keep production moving.

One robotics-specific multiplier is heterogeneity. A typical robot cell may include robot controllers from one vendor, safety PLCs from another, vision cameras from a third, and custom integration code written by a system integrator. Each component has different patch/support cycles, different security capabilities, and different logging outputs. That creates security “hand-offs” where responsibility is unclear—and attackers exploit unclear ownership.

A second multiplier is the fleet effect. Connectivity that enables centralized fleet management also enables fleet-wide blast radius. Compromise of a single fleet management credential can translate into control over hundreds or thousands of robots, especially when fleet services are designed for convenience rather than least privilege.

A third multiplier is physical access. Robots deployed in semi-public spaces (retail backrooms, hospitals, warehouses with contractors) are exposed to stronger insider risk and to attacks involving physical ports, removable media, or direct access to maintenance interfaces.

4. Threat Actors, Motives, and the OT Reality

Robotics cybersecurity must be threat-model driven. The relevant adversaries depend on deployment context, but several classes recur:

Cybercrime groups (ransomware, extortion). In industrial settings, cybercrime actors target availability because downtime is monetizable. Mandiant’s M-Trends 2024 reports that in 70% of ransomware-related cases, organizations learned about the intrusion from external sources [13]. That pattern is consistent with disruptive attacks where defenders do not detect compromise until the attacker announces it via a ransom note.

Nation-state and strategic actors. These actors often target industrial sectors for espionage, sabotage options, or strategic positioning. Their goals may include persistent access to industrial environments and the ability to disrupt or degrade critical processes.

Insiders and contractors. Robotics deployments often rely on vendors and integrators with elevated access. When credential hygiene is weak or shared accounts are common, insider risk increases and attribution becomes difficult.

Competitive and IP-driven actors. Robots embed valuable know-how: control parameters, optimized trajectories, product recipes, calibration data, and proprietary ML models. Theft of these artifacts can be as damaging as theft of traditional documents.

For defenders, mapping robotics threat behavior to known frameworks improves clarity. MITRE ATT&CK for ICS provides a matrix of adversary tactics and techniques observed in industrial environments, which helps robotics/OT teams translate “threat intelligence” into concrete hardening and detection priorities [10].

5. Safety Meets Security: The Coupling in Robotics

In robotics, safety and cybersecurity are coupled because both can affect whether the system behaves within acceptable risk bounds. Historically, safety engineering assumed failures were accidental (random faults, wear-out). Cybersecurity adds the possibility of intentional, adaptive failure induced by an attacker.

This coupling is increasingly reflected in standards. The ISO 10218-1:2025 preview notes that the updated standard includes “adding requirements for cybersecurity to the extent that it applies to industrial robot safety” [12]. This is a meaningful milestone: cybersecurity is being treated as a factor that can undermine functional safety if left unmanaged.

From an engineering standpoint, the safety-security relationship creates design patterns:

• Safety independence: Safety functions should remain effective even if non-safety compute is compromised.

• Integrity of safety-relevant data: When safety depends on sensor data or configuration, those inputs need integrity protection and auditable change control.

• Fail-safe vs fail-operational decisions: A secure design must define what happens under partial compromise (e.g., loss of network, loss of time sync, corrupted perception). Not all failures should trigger emergency stop—but the decision logic must be explicit, tested, and documented.

• Security updates as safety events: Patching a robot cell can change timing, performance, or certified behavior. Secure update mechanisms must therefore integrate with safety validation and change management processes.

A common failure mode is “security as friction.” If security controls make maintenance hard, operations teams bypass them. Robotics security must balance strong controls with usability, clear procedures, and operational reality.

6. Standards, Guidance, and Regulation: The Compliance Tail That Shapes Robotics

Robotics security is shaped by OT/ICS cybersecurity standards, enterprise risk frameworks, and product regulations—particularly for robots that are sold as connected products or deployed in regulated sectors.

6.1 ISA/IEC 62443: the most influential OT security standard family

The ISA/IEC 62443 series defines requirements and processes for implementing and maintaining electronically secure industrial automation and control systems (IACS) [8]. The series is structured across roles—asset owners, service providers/integrators, and product suppliers—enabling shared responsibility across the lifecycle.

6.2 NIST guidance for OT and enterprise frameworks

NIST SP 800-82 Rev.3 provides an overview of OT security, threats and vulnerabilities, and recommended countermeasures, emphasizing that OT systems have unique performance, reliability, and safety requirements [1]. NIST CSF 2.0 provides a cross-sector risk-management framework and emphasizes governance and supply chain risk management [2].

6.3 EU Cyber Resilience Act: secure-by-design becomes mandatory for “products with digital elements”

For robotics vendors selling connected robots into the EU, the Cyber Resilience Act (CRA) is a major regulatory driver. The European Commission notes that the CRA introduces mandatory cybersecurity requirements for manufacturers across planning, design, development, and maintenance, and requires vulnerability handling across the product lifecycle [7]. The CRA entered into force on 10 December 2024; the main obligations apply from 11 December 2027, with reporting obligations applying as of 11 September 2026 [7].

6.4 Supply chain artifacts: SBOMs and evidence

Robotics systems depend heavily on third-party software and firmware. The security community is increasingly using Software Bills of Materials (SBOMs) to improve transparency and response speed when vulnerabilities are disclosed. For robot vendors and integrators, SBOM practices link directly to vulnerability management, customer trust, and regulatory expectations.

7. Security Engineering for Robots: From Principles to Concrete Controls

Security engineering for robotics should be treated as an end-to-end lifecycle discipline, not a set of one-off hardening tasks. A mature approach aligns product development, deployment architecture, and operational monitoring.

Threat modeling in robotics must cover both cyber and physical consequences. Useful approaches include system decomposition (interfaces, data flows, trust boundaries), misuse cases (unsafe motion, degraded quality, production stoppage), and mapping to industrial threat behavior using ATT&CK for ICS [10].

Identity and access control must be fleet-grade. Robots often ship with permissive defaults designed for integration convenience; in production, robots need unique device identities, strong

authentication for every interface, least-privilege authorization, and credential rotation/revocation processes.

Network segmentation is a first-order control in OT. In robotics, a practical pattern includes a robot cell zone, an OT operations zone, a DMZ for integration services, and enterprise/cloud zones. Conduits (allowed communications) should be minimal, authenticated, logged, and resilient to failure.

Secure update and rollback is essential because robots are long-lived assets while vulnerabilities are discovered continuously. Secure update design requires signed firmware/software updates, protected key storage, staged rollout with health monitoring, and rollback plans aligned with downtime procedures.

Finally, logging and incident response must include robotics-specific telemetry. OT survey data suggests faster detection and containment are achievable with OT-specific monitoring and segmentation [4]. Robotics deployments should prioritize passive monitoring at cell boundaries, centralized collection of auth/config changes, and recovery playbooks that include safe-stop, isolation, program restoration, and validation steps.

8. Middleware Security: ROS, DDS, and the 'Network of Networks' Problem

Robotics middleware is where cybersecurity and robotics intersect most directly. Middleware defines how sensor data, control commands, and state estimates move between processes and computers. If middleware is insecure, the robot's "nervous system" is insecure.

ROS 1 achieved massive adoption in research and prototyping, but the original design did not prioritize network security. Open Robotics community discussion explicitly notes the lack of encryption, authentication, and access control in the original design, and recommends shielding ROS 1 deployments behind VPNs when public networks are involved [14].

ROS 2 moved to DDS as the default middleware and gained access to DDS-Security mechanisms such as authenticated discovery, encryption, and access control policies. However, these capabilities require configuration and operational discipline.

Research on SROS2 (Secure ROS 2) introduces tooling intended to make graph security usable and proposes DevSecOps-style workflows for generating and applying security artifacts [15]. A key argument is that security must be usable by robotics engineers; otherwise it will be bypassed or misconfigured.

In real deployments, middleware security fails due to certificate lifecycle issues, mixed-vendor compatibility, performance concerns that lead teams to disable encryption, incomplete authorization models, and debugging workflows that reintroduce insecure channels. Organizations should treat middleware security as a product feature with test coverage, operational procedures, and clear ownership.

9. Vulnerabilities in Robotics: Patterns and a Concrete Example

Robot vulnerabilities often resemble classic software vulnerabilities, but the presence of proprietary programming environments and factory integration patterns introduces unique pathways for compromise.

A concrete example involves offline programming (OLP) and controller ecosystems. TXOne Networks discusses a path traversal vulnerability (CVE-2023-1864) affecting FANUC robot offline programming [11]. The same analysis cites Trend Micro research noting that proprietary programming languages provided by industrial robot manufacturers sometimes include dynamic code loading functionality [11]. In a worst-case scenario, dynamic code loading can enable delivery of malicious logic into a controller, persistence on the robot cell network, and lateral movement to other robots or adjacent industrial assets—especially when engineering workstations are insufficiently segmented.

The defensive takeaway is not “robots are doomed.” It is that robotics security must include secure software supply chains for programming tools, validation of program artifacts before deployment, segmentation between engineering workstations and control networks, integrity monitoring and logging around program deployment workflows, and vendor coordination for timely patching and mitigation guidance.

In parallel, the growth of robotics-focused vulnerability disclosure ecosystems (e.g., robot vulnerability databases) reflects increasing attention to robotics as a distinct security domain [17].

10. Cloud and Fleet Robotics: Security at Scale

Fleet robotics is reshaping operations. Instead of treating each robot as an isolated asset, organizations increasingly manage fleets via cloud platforms that handle telemetry, task assignment, map distribution, software updates, and analytics.

This provides operational leverage but increases cybersecurity coupling: a compromised fleet management credential can affect every robot in scope, APIs expand the integration surface, and multi-tenant and third-party dependencies enlarge the dependency graph.

Enterprise breach data suggests third-party involvement is a growing driver of compromise. Verizon’s 2025 DBIR Executive Summary reports analysis of 22,052 security incidents and 12,195 confirmed breaches [5], and the DBIR highlights a sharp increase in third-party involvement (reported as 30%) [5]. For robotics, this is amplified by reliance on integrators, OEM remote support, and cloud vendors.

Security patterns that scale in fleet robotics include mutual TLS, fine-grained authorization, segmented blast radius by site/fleet/function, monitoring of API usage and token anomalies, and secure update channels with staged rollouts and automated health checks.

11. AI, Perception, and the New Attack Surface of Model-Centric Robotics

Robotics is increasingly AI-driven: deep learning for perception, reinforcement learning for control policies, and generative models for planning and human interaction. This creates security issues that differ from classic software vulnerabilities and increases the importance of governance for models and model-connected services.

IBM's Cost of a Data Breach Report 2025 highlights an "AI oversight gap." It reports that 13% of organizations experienced breaches involving their AI models or applications; among those, 97% lacked proper AI access controls [6]. It also reports that the most common incidents occurred through the AI supply chain (compromised apps, APIs, or plug-ins) and that ungoverned "shadow AI" usage increases exposure [6]. In robotics, perception and planning stacks often depend on multiple third-party models, plug-ins, and cloud APIs, so the AI supply chain becomes part of the robot's effective attack surface.

Robustness vs security: adversarial ML issues (e.g., spoofing) sit at the intersection of safety and security. Even when they are not "cyberattacks" in the traditional sense, they can cause integrity failures in perception, translating into unsafe or inefficient behavior. Robotics validation workflows should therefore include testing against realistic sensor spoofing and data-integrity scenarios.

Protecting models as assets: robotics models embody expensive IP and data. Model theft can reveal proprietary know-how and can enable more effective attacks. Controls include model access control, encryption at rest, protected model distribution, and monitoring for exfiltration.

The key step is to treat AI components as first-class assets: they require identity, access control, change control, and vulnerability management like any other software component.

12. A Practical Robotics Cybersecurity Roadmap

Security programs fail when they are abstract. Robotics teams need an actionable roadmap that respects operational constraints and aligns with OT reality.

12.1 The first 30–90 days: reduce obvious exposure

• Inventory: build an asset inventory of robots, controllers, gateways, HMIs, and engineering workstations.

• Access: eliminate shared accounts; enforce MFA for remote access; rotate credentials; disable unused services.

• Segmentation quick wins: restrict inbound access to robot cell networks; implement a DMZ for integrations.

• Backup and restore: create tested backups of controller programs, configuration, and key data.

• Logging baseline: centralize authentication and program deployment logs where feasible.

• Vendor engagement: document vendor patch channels and support terms for each component.

12.2 The next 3–12 months: make it repeatable

• Threat modeling: perform system-level threat modeling for representative robot deployments.

• Vulnerability management: track advisories and patches; use SBOMs where possible.

• Secure update pipeline: adopt signed updates, staged rollout, and rollback procedures.

• Middleware hardening: enable ROS 2/DDS security features where applicable; define key management procedures.

• Monitoring and response: deploy passive OT monitoring and robotics-specific playbooks.

• Training: cross-train robotics engineers in security basics and security teams in robotics/OT constraints.

12.3 12+ months: mature to an auditable program

• Align to standards: map controls to IEC 62443 requirements (asset owner, integrator, supplier).

• Product security: for vendors, build a secure development lifecycle aligned to IEC 62443-4-1.

• Red teaming: perform adversary simulations with safety safeguards; validate segmentation and recovery.

• Metrics and governance: adopt CSF profiles/tiers, define KPIs (patch latency, credential hygiene, dwell time).

• Regulatory readiness: if applicable, prepare for CRA obligations (secure-by-design evidence, reporting workflows).

The goal is to reduce the probability and impact of realistic attacks while keeping robots safe, reliable, and maintainable.

Conclusion

Robotics and cybersecurity are converging because robotics has become software-defined, connected, and scaled. The robot is now a node in a distributed system whose failure modes include physical consequences. This forces robotics teams to treat cybersecurity as an engineering requirement—on par with latency, accuracy, uptime, and safety.

The defensible path is clear: adopt OT-aware security architecture (segmentation, identity, monitoring), use standards and frameworks to structure responsibilities across asset owners, integrators, and suppliers, and build secure-by-design product capabilities that can be maintained for the multi-decade lifetimes common in industrial robotics.

Organizations that internalize this coupling will ship and operate robots that are not just capable—but trustworthy.

Appendix A: Robotics Security Checklist (Deployment)

This checklist is intended for teams deploying robots in production environments. It is not exhaustive, but it covers common failure points.

Identity & Access

□ Unique credentials per device and per human operator (no shared accounts)

□ MFA for remote access and administrative interfaces

□ Role-based access control: operator vs maintenance vs integrator vs vendor

□ Credential rotation/revocation process for contractors

Network & Segmentation

□ Robot cell networks separated from enterprise networks; explicit conduits only

□ Remote access terminates in a controlled jump host/broker (not direct to controllers)

□ Management interfaces not exposed to public networks

□ Time synchronization secured and monitored

Software & Updates

□ Signed firmware/software updates; verified on-device

□ Documented update window and rollback plan

□ Inventory of third-party components; SBOM where feasible

□ Vulnerability intake process and patch tracking

Monitoring & Recovery

□ Centralized logging for auth events, configuration changes, program deployments

□ Passive OT monitoring at cell boundary (where feasible)

□ Backups of controller programs and configuration; restore drills performed

□ Incident playbooks include safe-stop, isolation, validation, and return-to-service steps

Physical & Operational Controls

□ Maintenance ports and removable media controlled

□ Engineering workstations hardened and dedicated where possible

□ Change management for safety-relevant configuration and calibration

Appendix B: Glossary of Key Terms

Availability: ensuring systems remain operational and accessible when needed.

CIA triad: confidentiality, integrity, and availability—core security objectives.

CPS: cyber-physical systems that link computation, networking, and physical processes.

DDS: Data Distribution Service; middleware used by ROS 2 for publish/subscribe communication.

DMZ: demilitarized zone; a network segment that mediates communication between trust zones.

ICS/OT: industrial control systems / operational technology.

ISA/IEC 62443: a family of standards for cybersecurity of industrial automation and control systems.

Least privilege: granting only the permissions required to perform a task.

Purdue model: a reference architecture for segmenting industrial networks by levels.

ROS: Robot Operating System; open-source middleware and tools used widely in robotics.

SBOM: Software Bill of Materials; inventory of components in a software product.

Secure boot: boot process that verifies integrity/authenticity of software before execution.

SROS2: tooling ecosystem for enabling and managing ROS 2 security artifacts.

Threat modeling: structured analysis of how systems could be attacked and how to mitigate risks.

References

1. [1] NIST, Special Publication 800-82 Rev. 3: Guide to Operational Technology (OT) Security, September 2023. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r3.pdf
2. [2] NIST, The Cybersecurity Framework (CSF) 2.0, February 26, 2024. https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf
3. [3] ENISA, ENISA Threat Landscape 2024 (report). https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024
4. [4] Dragos, “SANS State of OT Security 2025: What the Data Tells Us,” November 24, 2025. https://www.dragos.com/blog/sans-state-of-ot-security-2025-what-the-data-tells-us
5. [5] Verizon, 2025 Data Breach Investigations Report (DBIR) Executive Summary, May 2025. https://www.verizon.com/business/resources/reports/2025-dbir-executive-summary.pdf
6. [6] IBM & Ponemon Institute, Cost of a Data Breach Report 2025 (PDF mirror), August 2025. https://www.bakerdonelson.com/webfiles/Publications/20250822_Cost-of-a-Data-Breach-Report-2025.pdf
7. [7] European Commission, Cyber Resilience Act (CRA) policy page (updated Dec 3, 2025). https://digital-strategy.ec.europa.eu/en/policies/cyber-resilience-act
8. [8] International Society of Automation (ISA), ISA/IEC 62443 Series of Standards overview. https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards
9. [9] IEC, IEC 62443-2-1:2024 (preview) Security program requirements for IACS asset owners. https://webstore.iec.ch/en/iec_catalog/product/preview/?id=L3B1Yi9wZGYvcHJldmlldy9pbmZvX2llYzYyNDQzLTItMXtlZDIuMH1iLnBkZg%3D%3D
10. [10] MITRE, ATT&CK for ICS Matrix. https://attack.mitre.org/matrices/ics/
11. [11] TXOne Networks, “FANUC Robot Off-Line Programming Path Traversal Vulnerability (CVE-2023-1864),” citing Trend Micro research. https://www.txone.com/blog/fanuc-robot-off-line-programming-path-traversal-vulnerability-cve20231864/
12. [12] ISO 10218-1:2025 preview (industrial robots safety requirements; includes cybersecurity mention). https://cdn.standards.iteh.ai/samples/73933/b5387b20934848a48b4518c9b2e5455d/ISO-10218-1-2025.pdf
13. [13] Mandiant, Special Report: M-Trends 2024 (PDF). https://services.google.com/fh/files/misc/m-trends-2024.pdf
14. [14] Open Robotics Discourse, “System security, ROS and its security” (ROS 1 design discussion). https://discourse.openrobotics.org/t/system-security-ros-and-its-security/468
15. [15] Mayoral-Vilches et al., “SROS2: Usable Cyber Security Tools for ROS 2.” https://www.researchgate.net/publication/362469001_SROS2_Usable_Cyber_Security_Tools_for_ROS_2
16. [16] Claroty, “State of CPS Security: OT Exposures 2025.” https://claroty.com/resources/reports/state-of-cps-security-ot-exposures-2025
17. [17] Alias Robotics, Robot Vulnerability Database (RVD) repository. https://github.com/aliasrobotics/RVD