Learning Objectives

By the end of this chapter, students will be able to:

1. Parse and analyze Software Bills of Materials (SBOMs) in both SPDX and CycloneDX formats.
2. Build dependency DAGs and compute transitive vulnerability propagation across firmware components.
3. Implement multi-signal counterfeit device detection using MAC OUI, firmware hash, TCP stack fingerprint, entropy analysis, and timing anomalies.
4. Design a 10-dimensional vendor trust scorecard with weighted scoring and grade computation.
5. Evaluate device compliance against the EU Cyber Resilience Act (CRA) 13 essential requirements.
6. Construct phylogenetic trees from firmware using function-level hashing and UPGMA clustering.
7. Verify software provenance using SLSA (Supply Chain Levels for Software Artifacts) attestations.

Agentic Lens

This chapter examines how to evaluate provenance and defects simultaneously. The main challenge is deciding whether the evidence is strong enough to trust a device, its firmware, or a supplier's claims. To help with this, the chapter introduces an analytic framework that uses a step-by-step decision process:

1. Define the evidence set: Collect all provenance artifacts, SBOM data, and defect or anomaly signals for the device or vendor claim.

2. Map the provenance chain: Trace component lineage from source through build, signing, and deployment, noting all observed and missing links.

3. Weigh defects vs. provenance: Compare defect signals (such as vulnerabilities, anomalies, counterfeit indicators) against provenance confidence, scoping any uncertainty.

4. Identify and record analytic conflicts: Document cases where signals contradict, and assign preliminary evidentiary weights to each.

5. Decide escalation or hold: Use a set of guardrails to determine whether to escalate, request more evidence, or place the claim in a hold state, avoiding single-signal overreliance.

6. Produce a bounded analytic output: For each claim, clearly state what the evidence supports, what it cannot yet support, and what additional steps are required for a stronger conclusion.

> **Caution:** The techniques described in this chapter have operational boundaries. Always verify assumptions against the specific deployment environment before relying on any automated output.

This framework helps doctoral students use a systematic approach to supply-chain analysis, making sure every recommendation is well-supported and based on solid evidence.

- **Agent role:** Assess supply-chain authenticity and determine which anomalies warrant escalation.
- **Observations:** SBOM structure, dependency paths, firmware hashes, MAC OUI, timing signals, provenance attestations, and vendor history.

- **Tools:** SBOM parsers, dependency DAG analysis, counterfeit detection modules, trust scorecards, phylogenetic clustering, and SLSA verification tools.
- **State:** Per-device provenance records, anomaly histories, confidence scores, and identification of missing evidence required to substantiate a strong claim.
- **Verifier: Employ** multi-signal corroboration, reference known-good baselines, and explicitly distinguish between documentation gaps and evidence of malicious activity.
- **Guardrails:** A single anomaly is typically insufficient to establish that a device is counterfeit. It is essential to distinguish between missing documentation and genuine indicators of tampering.
- **Watch out** for relying too much on a single signal. This can lead you to wrongly label complex yet legitimate supply chains as threats, which can undermine the credibility of your assessment.

Threat Model and Assumptions

This chapter treats the supply chain as a dynamic environment in which the quality and clarity of evidence can vary from case to case.

- **Threat model.** The attacker may compromise source code, build systems, package registries, update channels, firmware images, or device provenance records.
- **Threat model.** The defender also faces non-malicious ambiguity: incomplete SBOMs, sloppy vendor metadata, inconsistent manufacturing records, and reused components across product lines.
- **Assumption:** SBOMs and attestations can vary a lot in quality. Missing provenance is a warning sign, but it does not always indicate a compromise.
- **Assumption:** Counterfeit scores come from several weak or moderate signals. These scores help guide investigations, but they are not a replacement for legal or forensic decisions.
- **Assumption:** Vendor trust scores are based on facts you can check and how open the vendor is. They do not try to guess motives. This chapter distinguishes supplier mistakes, undocumented component reuse, and intentional tampering as distinct issues. Keeping these differences clear is important for solid analysis. **The 2020 SolarWinds breach showed that supply chain attacks can get past many defenses. Attackers put a backdoor in the Orion build process, which led 18,000 organizations, including US government agencies, to install the compromised update through normal patching. The XZ Utils backdoor (CVE-2024-3094), found in March 2024, also showed that even trust in open-source maintainers can be undermined by long-term social engineering.**

In the context of IoT and operational technology, supply chain threats affect both software and hardware. Counterfeit hardware—such as cameras running copied firmware, PLCs with modified bootloaders, or access points hiding backdoors—can put critical infrastructure at risk. For example, a surveillance camera might look real and have the right MAC prefix, but its firmware could add hidden network features that slip past **no**rmal defenses. Five independent detectors (MAC OUI, firmware hash, TCP stack, entropy, timing) produce an ensemble confidence score.

1. **Vendor trust scoring** – a 10-dimension scorecard rates vendors on CVE density, patch response, SBOM quality, and more.
2. **EU CRA compliance** – evaluates devices against the 13 essential security requirements of Regulation 2024/2847.
3. **Firmware phylogenetics** – builds evolutionary trees from firmware binary similarity to detect unauthorized forks.

4. **SLSA provenance** – verifies build attestations against SLSA Level 1-4 requirements.

Evidence label: Illustrative. **Method note:** The analyst readout shows how to apply the chapter's evidence-based workflow. **Assumptions:** Flagged devices, inherited CVEs, and compliance failures come from the modeled evidence set. **Boundary:** This example is not meant as a universal benchmark. Doctoral students should use it as a template for building their own analytic claims, focusing on careful evaluation of evidence boundaries and making sure their workflow can stand up in different real-world situations.

3 devices flagged as potential counterfeits with confidence above 0.85. 47 transitive CVEs inherited from shared OpenSSL 1.1.1. One vendor cohort scores poorly on patch response and SBOM quality. 8 devices fail the screened CRA checks used in this chapter."

9.2 SBOM Analysis

9.2.1 Formats and Parsing

The sbom_analyzer.py module supports two industry-standard SBOM formats:

- **SPDX** (ISO/IEC 5962:2021) – the Linux Foundation's format, widely used in open-source ecosystems.
- **CycloneDX** (OWASP) – designed for security use cases, with native vulnerability and license tracking.

```
# From apps/api/app/scanning/supply_chain/sbom_analyzer.py

def analyze_sbom( host: dict[str, Any], sbom_data: dict[str, Any] | None =
None, ) -> SBOMAnalysis: """ Full SBOM analysis for a single device."""

    if sbom_data is None:

        sbom_data = extract_sbom_from_firmware(host.get("firmware") or {})

    fmt = _detect_format(sbom_data)

    components = parse_sbom(sbom_data, format=fmt.value)

    dag = build_dependency_dag(components)

    max_depth = _compute_max_depth(dag)

    diamonds = detect_diamond_dependencies(dag)

    transitive = resolve_transitive_cves(dag)

    return SBOMAnalysis(

        device_ip=ip,

        format=fmt,

        component_count=len(components),

        max_depth=max_depth,

        diamond_deps=diamonds,
```

```
        transitive_cves=transitive,
    )
```

9.2.2 Dependency DAG Construction

The dependency DAG represents the transitive closure of all software components. Each node is a DependencyNode with a package name, version, and optional CPE. Edges represent "depends on" relationships.

```
graph TD A[Camera Firmware v3.1.2] --> B[BusyBox 1.31.1] A --> C[OpenSSL 1.1.1k] A --> D[curl 7.74.0] D --> C B --> E[musl libc 1.2.2] C --> E style A fill:#e1f5fe style C fill:#ffcdd2 style D fill:#fff9c4
```

9.2.3 Diamond Dependencies and Transitive CVEs

A **diamond dependency** occurs when two separate paths in the dependency graph converge on a single component. For example, both curl and camera firmware may rely on OpenSSL. A vulnerability in that shared library can propagate, affecting every system connected to it.

The resolve_transitive_cves() function traverses the dependency graph from the leaves, letting CVEs cascade upward to parent components. For example, a vulnerability in OpenSSL 1.1.1k can propagate to curl, affect the camera firmware, and threaten every device running that firmware. A single weak link can compromise an entire fleet.

9.3 Multi-Signal Counterfeit Detection

9.3.1 Detection Architecture

The counterfeit_detector.py runs five independent sub-detectors and combines them via a weighted ensemble:

```
# From apps/api/app/scanning/supply_chain/counterfeit_detector.py
_INDICATOR_WEIGHTS: dict[CounterfeitIndicatorType, float] = {
    CounterfeitIndicatorType.mac_oui_mismatch: 0.30,
    CounterfeitIndicatorType.firmware_hash_mismatch: 0.25,
    CounterfeitIndicatorType.stack_deviation: 0.20,
    CounterfeitIndicatorType.weak_rng: 0.10,
    CounterfeitIndicatorType.timing_anomaly: 0.15,
}
def detect_counterfeit( host: dict[str, Any], known_hashes: dict[str, Any] | None = None, ) -> CounterfeitResult: """Run all counterfeit sub-detectors and produce an aggregated result."""
    indicators: list[CounterfeitIndicator] = []
```

```
    detectors = [
        lambda: check_mac_oui(host),
        lambda: check_firmware_hash(host, known_hashes=known_hashes),
        lambda: check_tcp_stack(host),
        lambda: check_entropy(host),
        lambda: check_timing(host),
    ]
    for detector in detectors:
        indicator = detector()
        if indicator is not None:
            indicators.append(indicator)
    score = compute_counterfeit_score(indicators)
    verdict = classify_verdict(score)
    return CounterfeitResult(device_ip=ip, indicators=indicators, score=score,
verdict=verdict)
```

9.3.2 The Five Detection Vectors

```
graph LR A[Device Under Test] --> B[MAC OUI Check] A --> C[Firmware Hash Check] A -->
D[TCP Stack Fingerprint] A --> E[Entropy / RNG Analysis] A --> F[Timing Analysis] B
--> G[Weighted Ensemble] C --> G D --> G E --> G F --> G G --> H{Verdict} H -->|Score
> 0.7| I[Counterfeit] H -->|0.4 < Score < 0.7| J[Suspicious] H -->|Score < 0.4|
K[Authentic]
```

1. MAC OUI Mismatch (30%): The first three bytes of a MAC address identify the manufacturer (OUI). A device claiming to be a Hikvision camera should have an OUI registered to Hikvision. The module maintains a database of known vendor OUI prefixes:

```
_VENDOR_OUI_MAP: dict[str, list[str]] = {
    "hikvision": ["c0:56:e3", "44:19:b6", "c4:2f:90", "54:c4:15", "bc:ad:28"],
    "dahua": ["3c:ef:8c", "a0:bd:1d", "e0:50:8b", "40:2c:76"],
    "axis": ["00:40:8c", "ac:cc:8e", "b8:a4:4f"],
    ...
}
```

If the vendor found in HTTP banners or SSDP doesn't match the MAC OUI, this mismatch is a strong sign that the device could be counterfeit.

2. Firmware Hash Mismatch (25%): This check compares the device's firmware hash to a list of known-good hashes. If they don't match, it could mean tampering or that the device is counterfeit.

3. TCP Stack Fingerprint Deviation (20%): The stack_fingerprinter.py module checks how the device behaves on the network and compares it to what is expected for its claimed operating system. For example, if a device says it runs Embedded Linux but acts like Windows on the network, this difference should be investigated.

4. Entropy / RNG Analysis (10%): The entropy_analyzer.py module checks how random things like TLS nonces, TCP sequence numbers, and session IDs are. Counterfeit devices often exhibit predictable patterns, which can occur when the hardware is of low quality.

5. Timing Anomaly (15%): This test looks at jitter and clock drift. Real devices from a single manufacturer usually have consistent timing, but counterfeits made from mixed hardware often exhibit irregular timing.

9.4 Vendor Trust Scoring

The vendor_scorecard.py computes a 10-dimensional trust scorecard:

```
# From apps/api/app/scanning/supply_chain/vendor_scorecard.py

def compute_vendor_scorecard( vendor_name: str, devices: list[dict[str, Any]],
cve_data: list[dict[str, Any]] | None = None, ) -> VendorScorecard:
    dimensions: list[VendorDimension] = [
        score_cve_density(vendor_name, cve_data),
        score_patch_response(vendor_name),
        score_sbom_quality(vendor_name),
        score_firmware_signing(devices),
        score_eol_policy(vendor_name),
        score_disclosure_program(vendor_name),
        score_cert_management(devices),
        score_supply_chain_transparency(vendor_name),
        score_default_security(devices),
        score_compliance_certs(vendor_name),
    ]
    overall_score = _weighted_average(dimensions)
    overall_grade = compute_grade(overall_score)
    return VendorScorecard(
        vendor_name=vendor_name,
        overall_grade=overall_grade,
```

```
        overall_score=round(overall_score, 1),
        dimensions=dimensions,
    )
```

The ten dimensions are:

Dimension	Weight	Description
CVE Density	15%	CVEs per device, normalized by fleet size
Patch Response	15%	Average time from CVE disclosure to patch
SBOM Quality	10%	Completeness and format compliance of SBOMs
Firmware Signing	10%	Percentage of devices with signed firmware
EOL Policy	10%	Published end-of-life support timeline
Disclosure Program	10%	Bug bounty or coordinated disclosure program
Certificate Management	10%	Certificate rotation practices
Supply Chain Transparency	5%	Published supply chain documentation
Default Security	10%	Percentage of devices with non-default credentials
Compliance Certs	5%	Industry certifications (ISO 27001, IEC 62443)

Grades range from A+ (90-100) to F (0-39).

Evidence label: Illustrative. Method note: The vendor-dimension weights are set by chapter policy for comparison, not based on data. Assumptions: The ten dimensions are sufficiently important to include in a single scorecard. Boundary: Adjust the weights before using them for procurement or regulatory decisions.

9.5 EU Cyber Resilience Act Compliance

9.5.1 The CRA Framework

The EU Cyber Resilience Act (Regulation 2024/2847) establishes 13 essential security requirements for products with digital elements. The cra_compliance.py module evaluates each requirement:

```
# From apps/api/app/scanning/supply_chain/cra_compliance.py

def check_vuln_handling(device: dict[str, Any]) -> CRARequirement: """ CRA Art. 6(1)(a) -- Products shall have no known exploitable vulnerabilities."""
    cves = device.get("cves") or device.get("vulnerabilities") or []
```

```
    critical_cves = [ c for c in cves if str(c.get("severity", "")).lower() in
("critical", "high") ]
    if not cves:
        return CRARequirement(
            req_id="CRA-01",
            status=CRAStatus.passed,
            evidence="No CVEs associated with this device",
        )
    if critical_cves:
        return CRARequirement(
            req_id="CRA-01",
            status=CRAStatus.failed,
            evidence=f"{len(critical_cves)} critical/high CVEs found",
            remediation="Apply vendor patches for all critical/high CVEs",
        )
```

9.5.2 The 13 Essential Requirements

1. No known exploitable vulnerabilities (Art. 6(1)(a))
2. SBOM provision (Art. 6(1)(b))
3. Secure default configuration (Art. 6(1)(c))
4. Protection against unauthorised access (Art. 6(1)(d))
5. Confidentiality of stored/transmitted data (Art. 6(1)(e))
6. Data integrity protection (Art. 6(1)(f))
7. Data minimisation (Art. 6(1)(g))
8. Availability and resilience (Art. 6(1)(h))
9. Secure update mechanism (Art. 6(1)(i))
10. Vulnerability handling process (Art. 6(1)(j))
11. Security event logging (Art. 6(1)(k))
12. Incident notification capability (Art. 6(1)(l))
13. Secure disposal/transfer (Art. 6(1)(m))

Each check returns a CRARequirement result (pass, fail, or partial), an evidence note, and advice for fixing any problems.

9.6 Firmware Phylogenetics

9.6.1 UPGMA Clustering

The phylogenetic.py module builds evolutionary trees from firmware binary similarity using function-level hashing and UPGMA (Unweighted Pair Group Method with Arithmetic Mean) hierarchical clustering:

```
# From apps/api/app/scanning/supply_chain/phylogenetic.py

def build_phylogenetic_tree(firmwares: list[dict[str, Any]]) -> dict[str,
Any]: """Build a phylogenetic tree from firmware binary similarity."""

    hash_sets: dict[str, set[str]] = {}

    for fw in firmwares:

        label = _make_label(fw)

        fn_hashes = fw.get("function_hashes") or compute_function_hashes(fw)

        hash_sets[label] = set(fn_hashes)

    # Build pairwise distance matrix (1 - Jaccard similarity)

    distance_matrix = [[0.0] * n for _ in range(n)]

    for i in range(n):

        for j in range(i + 1, n):

            sim = compute_similarity(hash_sets[labels[i]],
hash_sets[labels[j]])

            distance_matrix[i][j] = 1.0 - sim

    # UPGMA clustering

    tree = upgma_cluster(distance_matrix, labels)

    return tree
```

Similarity is computed using the **Jaccard index** over function hash sets:

```
J(A, B) = |A intersection B| / |A union B|
```

If two firmware images share 90% of their function hashes, they probably come from the same SDK or codebase, even if they are from different vendors. This can show:

- **Shared SDK lineage** – multiple vendors using the same Realtek or MediaTek SDK
- **Unauthorized forks** – cloned firmware with minor modifications
- **Counterfeit ancestry** – firmware that is a direct copy of another vendor's binary

9.6.2 Interpreting the Tree

```
graph TD R[Root] --> A[Cluster A] R --> B[Cluster B] A --> C[Hikvision DS-2CD2185
v5.6.3] A --> D[Unknown Camera v1.0] B --> E[Dahua IPC-HFW v2.800] B --> F[Amcrest
IP4M v2.620] style D fill:#ffcdd2
```

In this example, the unknown camera is closely grouped with Hikvision firmware, sharing 95% of its function hashes. This close match suggests it could be a rebrand or a counterfeit using copied code.

9.7 SLSA Provenance Verification

The slsa_verifier.py evaluates firmware provenance against SLSA Level 1-4 requirements:

```
# From apps/api/app/scanning/supply_chain/slsa_verifier.py

SLSA_LEVEL_REQUIREMENTS: dict[int, dict[str, Any]] = {

    1: { "label": "SLSA Level 1 -- Documented build process", "requirements":
{ "provenance_available": "Provenance document exists", "build_logged": "Build
steps are logged/recorded", }, },

    2: { "label": "SLSA Level 2 -- Hosted build service", "requirements": {
"version_controlled_source": "Source in version control",
"hosted_build_service": "Build on hosted CI/CD",
"build_service_signed_provenance": "Signed provenance", }, },

    3: { "label": "SLSA Level 3 -- Hardened build platform", "requirements": {
"isolated_build_environment": "Isolated build env",
"non_falsifiable_provenance": "Non-falsifiable provenance",
"signed_provenance": "Digitally signed provenance", }, },

    4: { "label": "SLSA Level 4 -- Two-party review + hermetic builds",
"requirements": { "two_party_review": "Code reviewed by two parties",
"hermetic_build": "Hermetic build (no network)", "reproducible_build": "Build
is reproducible", }, },

}
```

The verifier also checks the **in-toto link metadata,** which is a chain of signed attestations for each step from the source commit to the deployed binary. If there is a break in this chain, it may mean the software was tampered with.

9.8 Monte Carlo Blast Radius Simulation

The attack_sim.py module runs Monte Carlo simulations to estimate the blast radius of a supply chain compromise. Given a compromised component (e.g., a backdoored SDK), the simulation:

1. Identifies all devices using firmware built on that component.
2. For each device, compute the probability of exploitation based on reachability and exposure.
3. Simulates lateral movement from compromised devices using the attack graph (Phase 4).
4. It shows the expected blast radius as a probability curve, making clear how many assets could be at risk.

This method provides procurement teams with risk-based insight. For example, if vendor X's SDK is breached, you can expect about 23 devices to be affected, with high confidence that the range will be 18-31.

---

9. B Deep Technical Expansion

9.B1 Provenance is a graph, not a logo

A supply-chain claim has edges. Component to package. Package to build. Build to signer. Signer to vendor. Vendor to reseller. Reseller to deployment. Break one edge, and the claim weakens. A logo on a web interface does not repair the graph.

The graph has to survive hostile questions. Who built this image? From which source? With which dependency lockfile? On which builder? Under which identity? Who signed it? Who shipped it? Who installed it? Which deployed devices actually run it? If one edge is missing, the analyst should shrink the claim rather than fill the gap with vendor confidence.

This matters because supply-chain failures rarely arrive as a clean confession. They arrive as a certificate that looks real, a firmware string that looks plausible, a reseller invoice that looks ordinary, and one stubborn inconsistency. Cyber analytics earns its place by preserving the inconsistency long enough to test it.

9.B2 Counterfeit conflict handling

Conflicting evidence should produce a conflict record, not an immediate verdict. A matching OUI and mismatching firmware lineage can coexist. A valid certificate and suspicious timing fingerprint can coexist. The analyst should record the conflict, isolate high-consequence use cases, and request the next artifact.

The conflict record should name each signal and its evidentiary weight. OUI is useful but weak against relabeling and recycled network interfaces. Firmware hash strength increases as the known-good corpus matures. Timing and TCP fingerprints can be noisy, but they catch devices that lie at the application layer. Entropy anomalies may identify packing, encryption, or harmless build differences. The analyst should not collapse those signals too early.

A high-consequence site should use conflict to choose a hold state. A police evidence facility might keep the camera off the evidence network. An airport might allow a sensor to report to a quarantine collector. A residential provider might block cloud callbacks until lineage is settled. The point is not to punish uncertainty. The point is to keep uncertainty from becoming silent trust.

9.B3 Patch lag as vendor behavior

Patch lag measures behavior over time. It is harder to fake than a brochure. A vendor that repeatedly ships late fixes is teaching the buyer how future incidents will feel. The score should age evidence, separate critical from low-severity lag, and preserve the difference between no disclosure and no vulnerability.

Patch lag should be read as a survival signal. Some vendors disclose quickly but patch slowly. Some patches were quietly disclosed poorly. Some patch the current branch but abandon fielded devices. Some publish advisories that customers cannot map to installed firmware. Each pattern creates a different operational risk.

The metric should keep dates separate: disclosure date, vendor acknowledgment date, patch release date, customer availability date, and field deployment date. A single “days to patch” number hides where the supply chain failed. Doctoral work should avoid that compression unless the underlying data remain available.

9.B4 SBOM scope interrogation

Every SBOM needs a scope statement. Does it cover the web UI bundle, bootloader, vendor SDK blobs, kernel modules, or cloud connector libraries? If unknown, the SBOM is still useful, but the claim must be scaled back. Scope is the difference between inventory and theater.

The hardest part of SBOM review is the negative space. The document lists components but rarely what it failed to see. Firmware may include a Linux userland, a bootloader, a radio module, a vendor SDK, a web front end, and a cloud client. If the SBOM covers only one layer, the missing layers are not harmless. They are an unscored attack surface.

An expert review should ask for evidence extraction. Which image was unpacked? Which filesystem was mounted? Which package manager databases were parsed? Which binaries were fingerprinted without package metadata? Which blobs were left unidentified? A precise SBOM with a narrow scope is useful. A broad SBOM with no extraction trail is a theater of the absurd.

9.B5 Transparency logs for firmware reality

Firmware integrity improves when releases become publicly auditable events. A transparency log can record firmware hashes, signing identities, release times, device-family mappings, and revocation events. The log does not prove the firmware is safe. It proves that the release claim is visible, append-only, and difficult to rewrite quietly.

That distinction is valuable in IoT and OT procurement. A factory may find a controller with a firmware hash absent from the vendor's release log. A police facility may find two cameras with the same model string but different hashes. An airport may learn that a maintenance contractor has installed a build that exists only in a private channel. These facts do not, by themselves, prove compromise. They justify a hold, vendor challenge, and stronger collection action.

The log should tie to deployment evidence. A hash in a vendor portal is not enough. The organization must connect the advertised release to the observed device, the reseller chain, the installation record, and the current runtime measurement. That connection turns supply-chain security from paperwork into cyber analytics.

9.B6 Vendor dispute workflow

A mature buyer needs a dispute workflow before the first suspicious device arrives. The workflow should specify what evidence the buyer will share, what the vendor must provide, how long the vendor has to respond, what temporary controls apply, and who can release the device from hold. Without it, every counterfeit investigation becomes a negotiation under pressure.

The vendor may be right: the device could be a regional build, a replacement part, a beta hotfix, or a manufacturing-line variant. The buyer may be right: the device could be counterfeit, tampered, or gray-market. The workflow should support both possibilities without surrendering control. It should preserve the observed hash, network behavior, photographs, procurement documents, serial number, boot logs, and any vendor response.

For doctoral students, the key is adjudication under incomplete evidence. A binary verdict may not be available. The right output may be "not admitted to safety-critical segment," "allowed only behind broker," "accepted after vendor attestation and runtime measurement," or "rejected and reported." The answer is operational, not rhetorical.

9.B7 Procurement clauses that analytics can enforce

Procurement language often fails because it asks for virtue rather than evidence. “Vendor shall maintain secure development practices” sounds serious, but it is hard to test at the receiving end. Better language names artifacts: SBOM format, signing requirement, release-log entry, vulnerability-disclosure contact, patch notification time, attestation format, device identity fields, and right-to-audit evidence.

Analytics should be designed around those clauses. If the contract requires a firmware hash in a transparency log, the receiving workflow checks the log. If it requires SBOM coverage for bootloader and userland, the SBOM parser records the scope. If it requires a vulnerability notice within a defined time, the vendor scorecard tracks dates. The procurement clause becomes executable.

This is where Chapter 9 should feel difficult for experts. It does not ask them to memorize SBOM acronyms. It asks them to design a chain where a legal promise becomes a measurable control and a measurable control becomes a defensible operational decision. That is the difference between buying security and proving it at the loading dock.

9. A Advanced Practitioner Fieldbook: What Experts Still Miss

This fieldbook expands Chapter 9 for doctoral students and senior cybersecurity practitioners. It assumes the reader knows the vocabulary. The goal is harder. The reader must reason under pressure, preserve evidence boundaries, and explain why the analysis should be trusted when the environment fights back.

The text uses short sentences on purpose. They help reveal weak claims and make the material easier to teach in a live seminar. As you prepare for peer teaching or seminar facilitation, use these concise statements to prompt quick critique, encourage debate, and highlight ambiguous claims. In analytic drills, invite students to revise each short sentence, challenge their assumptions, or restate it as an evidence-based conclusion. These techniques train doctoral students to spot unsupported logic and explain their reasoning clearly under pressure.

9.A1 The fake camera with a real certificate

A police facility buys cameras through a reseller—the certificate chains to a familiar vendor. The firmware hash does not match the known release family. The device streams video, but its timing fingerprint is wrong.

Figure 9.5: Counterfeit Detection Pipeline

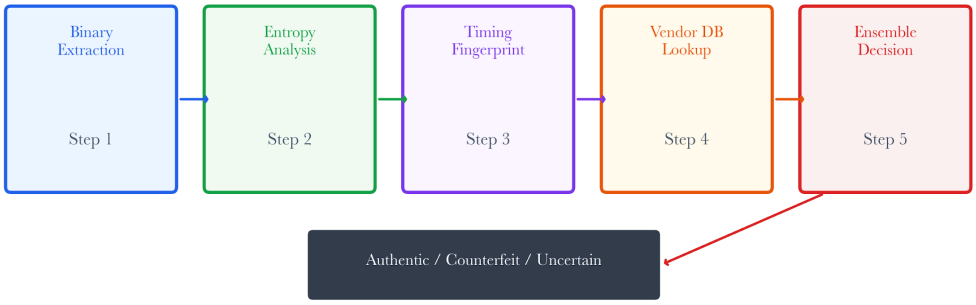

Figure 9.A1: ch09 fig05 counterfeit detection pipeline.

Figure note: The detection pipeline belongs to the fake-camera case. It shows why certificate, firmware, network, entropy, and timing evidence must be fused.

For the fake camera with a real certificate, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger of the fake camera with a real certificate is the false sense of closure it creates. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for the fake camera with a real certificate produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A2 SBOM as sworn inventory

An SBOM is a sworn inventory. It names components. It can omit, misname, or flatten dependencies. The analyst must test how it was produced.

Figure 9.2: SPDX vs CycloneDX Feature Comparison

Feature	SPDX 2.3	CycloneDX 1.5
License tracking	Excellent	Good
Vulnerability refs	Good	Excellent
Dependency graph	Basic	Rich
Composition analysis	Limited	Full
ISO standard	ISO/IEC 5962	ECMA-424
Serialization	TV/JSON/RDF/XML	JSON/XML/Protobuf

Figure 9.A2: ch09 fig02 spdx vs cyclonedx.

Figure note: The SPDX/CycloneDX comparison makes the SBOM scope. A sworn inventory must say what format, layer, and artifact boundary it covers.

For an SBoM as a sworn inventory, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in SBOMs as sworn-in inventories is false closure. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for SBOM as sworn inventory produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A3 Diamond dependencies and hidden blast radius

Two vendors may ship different products that converge on the same vulnerable library. A diamond dependency turns one component weakness into fleet risk.

Figure 9.4: Diamond Dependency Problem

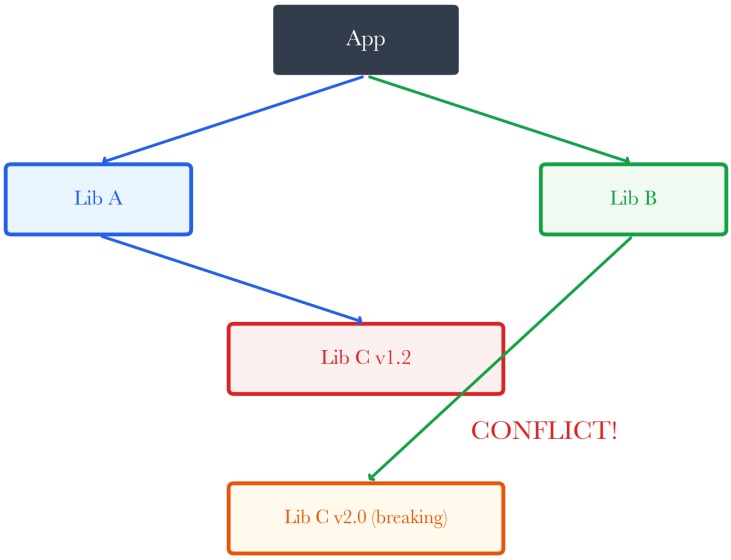

Figure 9.A3: ch09 fig04 diamond dependency.

Figure note: The diamond dependency diagram makes the hidden blast radius visible. One shared component can quietly bind otherwise separate device families.

For diamond dependencies and hidden blast radius, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in diamond dependencies and hidden blast radius is false closure. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for diamond dependencies and hidden blast radius produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A4 SLSA provenance under procurement pressure

Procurement teams want yes or no. Provenance gives levels of build discipline. The analyst translates level gaps into operational risk.

Figure 9.10: SLSA Provenance Levels

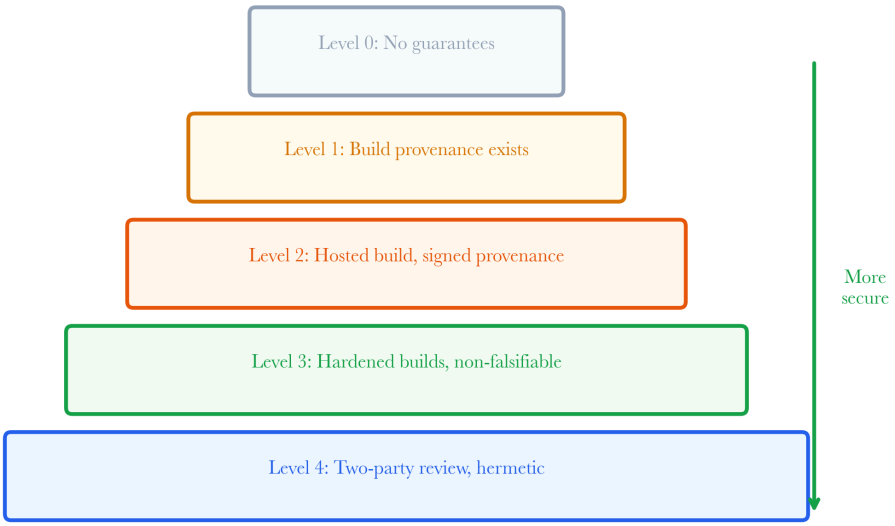

Figure 9.A4: ch09 fig10 slsa provenance levels.

Figure note: The SLSA figure belongs with provenance pressure. It turns build claims into reviewable requirements instead of procurement theater.

For slsa provenance under procurement pressure, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in slsa provenance under procurement pressure is false closure. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for slsa provenance under procurement pressure produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A5 Counterfeit detection as evidence fusion

No single signal convicts a counterfeit. Entropy, timing, ancestry, component markings, certificates, and vendor history must be combined.

Figure 9.6: Five-Signal Ensemble for Counterfeit Detection

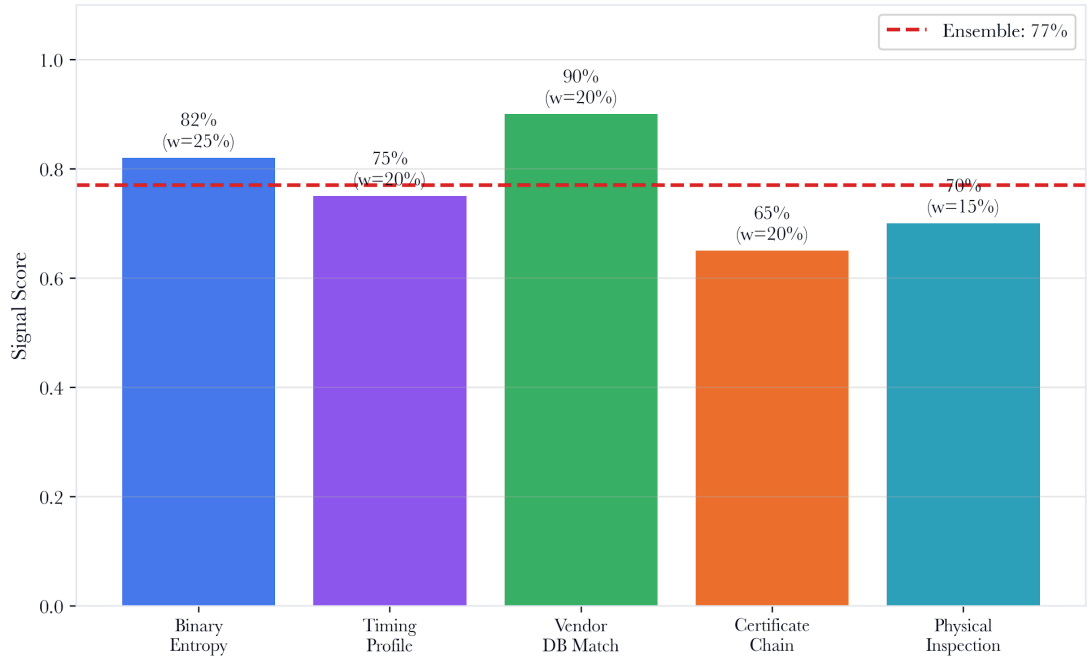

Figure 9.A5: ch09 fig06 five signal ensemble.

Figure note: The ensemble figure shows counterfeit adjudication as weighted evidence. It prevents a single comfortable signal from overruling contradictions.

For counterfeit detection as evidence fusion, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in counterfeit detection, as in evidence fusion, is false closure. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for counterfeit detection as evidence fusion produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A6 Firmware phylogenetics for adults

Firmware trees uncover hidden families, unexpected forks, copied SDKs, suspicious gaps, and abrupt leaps that defy official release stories.

Figure 9.9: Firmware Phylogenetic Tree

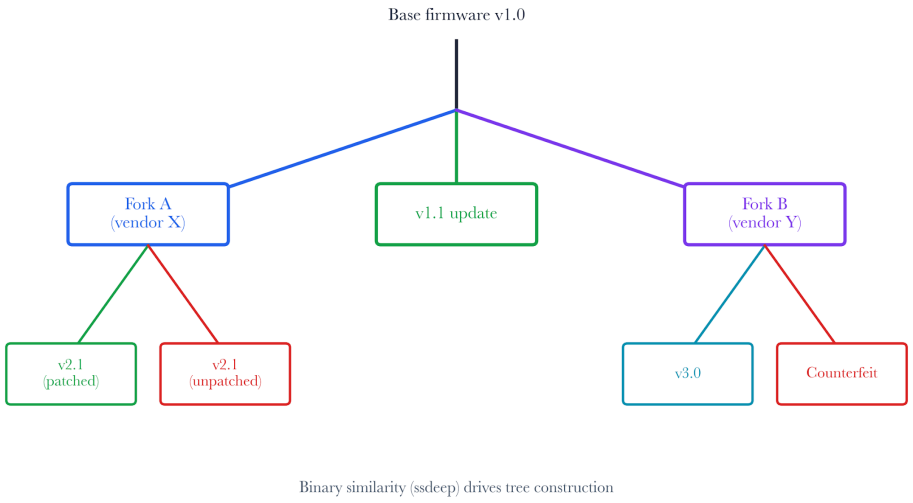

Figure 9.A6: ch09 fig09 firmware phylogenetic tree.

Figure note: The phylogenetic tree matches firmware lineage. It helps detect a device with the correct label, but the wrong family history.

For firmware phylogenetics for adults, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in firmware phylogenetics for adults is the risk of false closure. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for firmware phylogenetics for adults produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A7 Vendor trust as a perishable asset

Vendor trust is a perishable asset. A vendor swift with patches last year might now deliver only opaque binaries.

Figure 9.7: Vendor Trust Scorecard

Vendor	SBOM	Patch SLA	CVE History	Trust Score
Cisco	Full	24h	Low	92/100
Hikvision	Partial	72h	Medium	58/100
TP-Link	None	90d	High	31/100
Siemens	Full	48h	Low	87/100
Ubiquiti	Partial	48h	Medium	65/100

Figure 9.A7: ch09 fig07 vendor trust scorecard.

Figure note: The trust scorecard belongs with vendor behavior. Patch lag, disclosure quality, and support history become evidence, not reputation.

For vendor trust as a perishable asset, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in vendor trust as a perishable asset is the risk of false closure. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for vendor trust as a perishable asset produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A8 CRA compliance without theater

Compliance mapping only gains value when it is anchored in real evidence: vulnerability handling, secure defaults, timely updates, and transparent SBOM disclosure.

Figure 9.8: EU Cyber Resilience Act Requirements

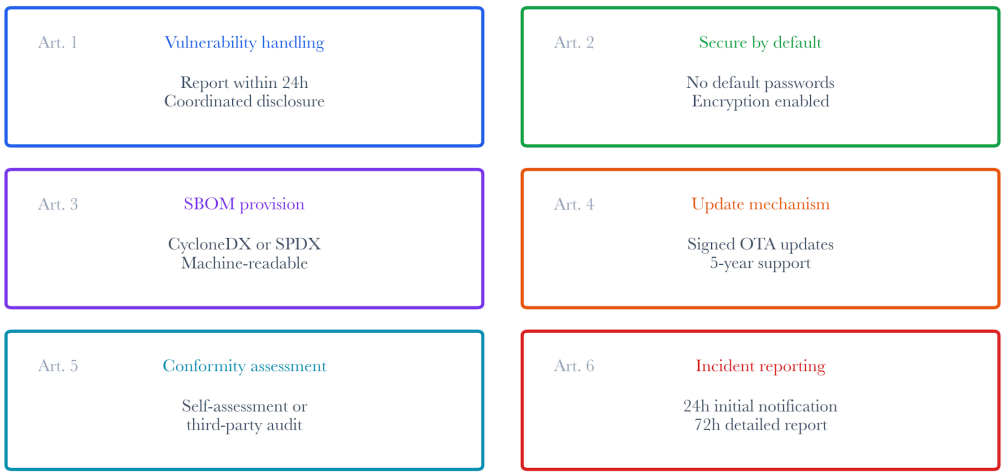

Figure 9.A8: ch09 fig08 eu cra requirements.

Figure note: The CRA requirements figure anchors compliance in specific obligations. It keeps the subsection tied to enforceable controls rather than slogans.

For CRA compliance without theater, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in CRA compliance without theater is the false sense of closure. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for CRA compliance without theater produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A9 The reseller fog

Resellers blur the lines—altering labels, bundling firmware, and muddying responsibility. The analyst's job is to trace a clear path from manufacturer to deployment.

For the reseller fog, evidence quality is not evenly distributed. The airport operations center may provide a precise artifact lineage, but it still hides the condition that matters. That is why chapter-level mastery requires evidence-based boundaries, not vocabulary recall.

The review question is practical: where can an adversary make the system overconfident? The answer usually sits between vendor response, provenance graph, and quarantine hold. That gap is where expert students should work.

The section should end with an action. Quarantine, shadow-test, narrow the claim, collect one more artifact, or reject the conclusion. A vague recommendation is a failed analytic output.

For the reseller fog, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in the reseller fog is false closure. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for the reseller fog produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A10 Component health as fleet intelligence

Component health is a living record: maintainer activity, release rhythm, patch delays, transitive exposure, and the maturity of exploits all shape its pulse.

Component health as fleet intelligence should change how the analyst briefs on risk. In the police evidence facility, leadership does not need a tour of the algorithm. They need to know which artifact lineage is admissible, which assumption is fragile, and which decision can be taken now.

The system should resist false closure. When vendor response and quarantine hold disagree, the correct state may be unresolved. That is a professional answer, provided the unresolved state has an owner, expiry, and next evidence request.

The highest-grade student will keep model behavior, governance, and mission consequence separate. Merging them into one confidence score destroys the audit trail that makes cyber analytics defensible.

For component health as fleet intelligence, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in component health as fleet intelligence is false closure. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for component health as fleet intelligence produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A11 When a clean SBOM lies by omission

A spotless SBOM can be treacherous if it quietly omits blobs, bootloaders, web UI bundles, or vendor SDK code.

In the factory cell, when a clean SBOM is omitted, it becomes a claim-control problem within supply-chain analytics. The analyst starts by naming the observed artifact lineage, the transformation that touched it, and the smallest defensible conclusion. Anything stronger has to wait for evidence.

The failure path is subtle. The dashboard combines vendor response, provenance graph, and quarantine hold into a single status. A senior reviewer should split them apart and preserve the disagreement as a first-class record.

The stop rule matters. The subsection should leave the reader able to say what blocks the claim, what promotes it, and what keeps it in review. That discipline is the difference between analytics and decorative scoring.

For when a clean SBOM lies by omission, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger when a clean SBOM lies by omission is false closure. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for when a clean SBOM lies by omission produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A12 The counterfeit that is safer today

A counterfeit device might boast fewer known CVEs than its genuine counterpart, yet hidden maintenance lapses and provenance gaps still cast the longest shadow over risk.

Treat the safer counterfeit today as an adversarial hearing, not a feature description. The municipal water site gives the system partial data, delayed data, and political pressure. The answer must still bind the artifact lineage to a named decision.

A strong implementation keeps the uncomfortable edge visible. If the vendor response says proceed while the quarantine hold says wait, the system should not average the conflict into the confidence. It should record the conflict and assign ownership.

The doctoral move is to ask for the counterfactual. What observation would make the original claim false? If the workflow cannot answer, it has built a belief engine rather than a cyber-analytic control.

For the counterfeit that is safer today, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in the counterfeit that is safer today is the false sense of closure it creates. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for the counterfeit that is safer today produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A13 Supply-chain blast radius simulation

A component compromise moves through procurement, firmware reuse, shared credentials, cloud services, and update channels.

The operational test for supply-chain blast radius simulation is whether a second expert can replay the reasoning. In a residential safety platform, this means the input record, transformation, exception handling, and decision authority must survive handoff.

The dangerous shortcut is to trust the most convenient signal. The provenance graph may look stable, while quarantine hold carries the real warning. Vendor response may look mathematically clean, while the mission context changes the cost of error.

A useful report should be modest and sharp. It should say what the analyst saw, what it inferred, what it refused to infer, and which collection step would move the case forward.

For supply-chain blast radius simulation, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in supply-chain blast radius simulation is false closure. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for supply-chain blast radius simulation produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A14 Adjudicating vendor claims

A vendor claim should survive artifact checks, observed behavior, provenance evidence, and timely correction of defects.

In adjudicating vendor claims, evidence quality is unevenly distributed. The regional hospital may provide a precise artifact lineage and still hide the condition that matters. That is why chapter-level mastery requires evidence boundaries, not vocabulary recall.

The review question is practical: where can an adversary make the system overconfident? The answer usually sits between vendor response, provenance graph, and quarantine hold. That gap is where expert students should work.

The section should end with an action. Quarantine, shadow-test, narrow the claim, collect one more artifact, or reject the conclusion. A vague recommendation is an analytic failure.

For adjudicating vendor claims, the review starts with the firmware artifact and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in adjudicating vendor claims is the risk of false closure. Vendor response may support one interpretation while the provenance graph points toward another. The case should remain open until the hold state explains why the stronger claim is admissible or why it has been refused.

A senior answer for adjudicating vendor claims produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof. ### 9.A15 Doctoral Mastery Check

A student has mastered Chapter 9 when they can adjudicate a device, firmware image, or vendor claim as a chain of evidence rather than a brand impression. The answer must connect the SBOM scope, release hash, build provenance, reseller history, runtime fingerprint, and deployment consequence.

Perturb the case with five supply-chain shocks: make the certificate valid but the firmware lineage incorrect, omit the bootloader from the SBOM, make the reseller path gray-market, make the vendor patch date disputed, and make the vulnerable component shared by a safety device. The student should choose a hold state and justify it.

9.A16 Expert Failure Review Drill

The final drill begins with a comforting procurement story. A vendor provides an SBOM. The reseller provides a certificate of origin. The device passes functional testing. No critical CVEs appear in the declared components. The purchasing office wants approval. Then the analyst notices that the firmware lineage does not fit the vendor release family. The web interface resembles the vendor's product, but the bootloader strings, entropy profile, and timing fingerprint point elsewhere.

The correct response is not an immediate accusation but evidence preservation. The analyst should mark the claim as contested, preserve the artifact hash, record the SBOM scope, request provenance from the vendor, compare firmware ancestry against known families, and isolate the device from high-consequence zones until

the conflict is resolved. This is a hard answer because it slows business without pretending to know more than the evidence supports.

The advanced reporting sentence should say: “The product identity is not rejected, but it is no longer accepted as a single-source vendor claim.” This is the language of mature supply-chain analytics. It avoids drama and false certainty. It gives procurement, operations, and security a defensible control path. The strongest students should also propose a control. A contested-origin hold prevents the device from entering sensitive zones. A provenance request clock starts. A firmware-family comparison is repeated when the vendor responds. A purchasing exception is documented if operations accepts the risk. This small discipline turns supply-chain suspicion into auditable governance.

9.D Seminar War Rooms

This section is intentionally demanding. Each war room asks the reader to work like a senior analyst responsible for supply-chain graph decisions in live IoT or OT environments. The task is not to recite the chapter but to make a bounded claim under pressure and defend the evidence boundary when another expert attacks it.

9.D1 Gray-Market Camera

War Room 1 begins with the gray-market camera. The first analyst wants to accept the firmware artifact because it fits the expected story. The second analyst refuses to move until the vendor attestation is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should interrogate the case by changing one fact a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

For the gray-market camera, the analytical output is a compact evidence packet. This evidence packet typically contains these elements:

- Input record: the observed artifact, device details, and all relevant evidence logs or SBOM entries

- Transformation: a summary of the analytic process, such as firmware unpacking, hash calculation, anomaly detection, or scope clarification

- Unresolved conflict: a clear statement of any conflicting evidence or uncertainties that require attention

- Decision authority: who owns the next decision and what authority or role can release the device or escalate the case

- Hold state: the current operational disposition, such as quarantine, shadow-test, or restriction

- Leadership sentence: a concise, operationally-focused statement that is cautious, precise, and avoids claims not supported by observed evidence

Example template:

Evidence Packet for [Device/Claim]:

- Input record: [Summarize key observed artifacts]

- Transformation: [Describe analytic steps applied]

- Unresolved conflict: [State conflicting or missing evidence]

- Decision authority: [Name or role responsible]

- Hold state: [Current status; e.g., "Quarantine pending vendor response"]

- Leadership sentence: ["Device remains in hold due to unresolved firmware provenance; operational deployment is paused until supporting evidence is collected."]

That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For a gray-market camera, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the gray-market camera is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove. ### 9.D2 Baggage Scanner Firmware

War Room 2 begins with the baggage scanner firmware. The first analyst wants to accept the firmware artifact because it fits the expected story. The second analyst refuses to move until the vendor attestation is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should challenge the case by changing one fact a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

The final artifact for the baggage scanner firmware is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For baggage scanner firmware, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the baggage scanner firmware is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also includes one leadership sentence weaker than the analyst wants but stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

War Room 3 begins with the PLC clone. The first analyst wants to accept the firmware artifact because it fits the expected story. The second analyst refuses to move until the vendor attestation is tied to an observed

artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should bound the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

The final artifact for the PLC clone is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For PLC clone, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the PLC clone is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove. ### 9.D4 Infusion Gateway Sdk

War Room 4 begins with the infusion gateway SDK. The first analyst wants to accept the firmware artifact because it fits the expected story. The second analyst refuses to move until the vendor attestation is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should quarantine the case by changing one fact a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

The final artifact for the infusion gateway SDK is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For the infusion gateway SDK, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the infusion gateway SDK is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also includes one leadership sentence weaker than the analyst wants but stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

War Room 5 begins with the home router identity. The first analyst wants to accept the firmware artifact because it fits the expected story. The second analyst refuses to move until the vendor attestation is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should simulate the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must

state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow- test, collect, rollback, or reject.

The final artifact for the home router identity is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For home router identity, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for home router identity is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove. ### 9.D6 Water Sensor Reseller

War Room 6 begins with the water sensor reseller. The first analyst wants to accept the firmware artifact because it fits the expected story. The second analyst refuses to move until the vendor attestation is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should replay the case by changing one fact a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

The final artifact for the water sensor reseller is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For a water sensor reseller, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the water sensor reseller is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also includes one leadership sentence weaker than the analyst wants but stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

War Room 7 begins with the bootloader gap. The first analyst wants to accept the firmware artifact because it fits the expected story. The second analyst refuses to move until the vendor attestation is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should audit the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

The final artifact for the bootloader gap is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also contains one leadership

sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For the bootloader gap, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the bootloader gap is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove. ### 9.D8 Port Crane Component

War Room 8 begins with the port crane component. The first analyst wants to accept the firmware artifact because it fits the expected story. The second analyst refuses to move until the vendor attestation is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should stage the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

The final artifact for the port crane component is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For the port crane component, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the port crane component is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove. ### 9.D9 Patch-Lag Dispute

War Room 9 begins with the patch-lag dispute. The first analyst wants to accept the firmware artifact because it fits the expected story. The second analyst refuses to move until the vendor attestation is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should defer the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

The final artifact for the patch-lag dispute is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For a patch-lag dispute, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and

which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the patch-lag dispute is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove. ### 9.D10 Procurement Receiving Dock

War Room 10 begins with the procurement receiving dock. The first analyst wants to accept the firmware artifact because it fits the expected story. The second analyst refuses to move until the vendor attestation is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should escalate the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow- test, collect, rollback, or reject.

The final artifact for the procurement receiving dock is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For the procurement receiving dock, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the procurement receiving dock is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove. ### 9.D11 Counterfeit Safe-Today Paradox

War Room 11 begins with the counterfeit safe-today paradox. The first analyst wants to accept the firmware artifact because it fits the expected story. The second analyst refuses to move until the vendor attestation is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should rollback the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow- test, collect, rollback, or reject.

The final artifact for the counterfeit safe-today paradox is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For the counterfeit safe-today paradox, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the counterfeit safe-today paradox is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove. ### 9.D12 Binary Transparency Gap

War Room 12 begins with the binary transparency gap. The first analyst wants to accept the firmware artifact because it fits the expected story. The second analyst refuses to move until the vendor attestation is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should instrument the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow- test, collect, rollback, or reject.

The final artifact for the binary transparency gap is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hold state. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For the binary transparency gap, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the binary transparency gap is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove. ## 9. E Adversarial Oral Examination

These oral-exam prompts are designed to make expert students uncomfortable in the right way. Each prompt forces a precise claim about supply-chain adjudication, then changes one operational fact that could occur in a real IoT or OT environment.

9.E1 When the certificate is valid, but the hash is unknown

The examiner gives the student a clean initial narrative, then reveals that the certificate is valid but the hash is unknown. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for the certificate is valid. However, the hash is unknown and has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter. ### 9.E2 When Sbom covers userland but not bootloader

The examiner gives the student a clean initial narrative, then reveals that SBOM covers the userland but not the bootloader. The student must decide whether the original recommendation survives. A passing answer

does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for an SBOM that covers userland but not the bootloader has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter. ### 9.E3 When the reseller gives an invoice without a custody trail

The examiner gives the student a clean initial narrative, then reveals that the reseller gives an invoice without a custody trail. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for the reseller when an invoice is issued without a custody trail has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and the next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter. ### 9.E4 When Vendor claims private hotfix

The examiner gives the student a clean initial narrative, then reveals that the vendor claims a private hotfix. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for vendor claims private hotfix has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter. ### 9.E5 When Runtime fingerprint contradicts model label

The examiner gives the student a clean initial narrative, then reveals that the runtime fingerprint contradicts the model label. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an

incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for the runtime fingerprint contradicts the model label, which has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter. ### 9.E6 When Patch lag conflicts with active exploitation

The examiner gives the student a clean initial narrative, then reveals that patch lag conflicts with active exploitation. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for patch lag conflicts with active exploitation has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter. ### 9.E7 When Dependency appears in safety and non-safety assets

The examiner gives the student a clean initial narrative, then reveals that dependency appears in safety and non-safety assets. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for dependency appears in safety and non-safety assets, has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter. ### 9.E8 When a counterfeit device is operationally safer than an old genuine unit

The examiner gives the student a clean initial narrative, then reveals that the counterfeit device is operationally safer than the old genuine unit. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for a counterfeit device is operationally safer than the old genuine unit, which has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and

next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter. ## 9.9 Limitations

- SBOM quality is often poor in the field. Missing components, vague versions, and proprietary blobs can hide the very dependencies the analysis is trying to score. Practical mitigation strategies include: requesting supplementary SBOMs or extracting evidence from vendors; using automated tools to analyze the negative space (e.g., detecting unlisted binaries or modules); cross-verifying component claims through firmware unpacking and runtime inspection; and maintaining a list of required SBOM fields for procurement checklists. If gaps are found, analysts should clearly flag the scope, seek additional artifacts, and treat uncovered areas as untrusted surfaces until confirmed.
- Counterfeit detection in this chapter is a weighted screening ensemble. A high score warrants investigation, but it is not, by itself, forensic proof of counterfeiting. Vendor trust grades compress many dimensions into a single number. That is useful for triage, but it can hide which exact governance failure matters most for a specific deployment.
- The CRA checks here are screened technical signals aligned to the regulation's requirements. They are not a legal certification or a substitute for a full conformity assessment.
- Firmware phylogenetics can reveal lineage and suspicious forks, but similarity alone does not distinguish legitimate shared SDK use from malicious cloning without additional provenance evidence.

9.10 Summary

Phase 9 provides a comprehensive supply chain integrity assessment:

1. **SBOM Analysis** – parses SPDX/CycloneDX SBOMs, builds dependency DAGs, resolves transitive CVEs.
2. **Counterfeit Detection** – five-signal weighted ensemble (MAC OUI, firmware hash, TCP stack, entropy, timing).
3. **Vendor Trust Scoring** – 10-dimension scorecard with A+ through F grades.
4. **EU CRA Compliance** – automated evaluation of 13 essential security requirements.
5. **Firmware Phylogenetics** – UPGMA clustering over function hash Jaccard distance.
6. **SLSA Provenance** – Level 1-4 attestation verification with in-toto chain checking.

Review Questions

1. A camera's SBOM shows OpenSSL 1.1.1k as a transitive dependency through both curl and the RTSP library. Explain how a diamond dependency amplifies the blast radius of CVE-2022-0778 (OpenSSL infinite loop). How would you compute the blast radius across a fleet of 200 cameras?
2. Device X claims to be a "Hikvision DS-2CD2185" but has a MAC OUI of 00:11:22 (unregistered) and a firmware hash of abc123 (not in the known-good database). Its TCP stack shows Windows-like behavior. Compute the weighted counterfeit score and classify the verdict.
3. Compare SPDX and CycloneDX formats. What information does CycloneDX capture that SPDX does not? When would you prefer one format over the other?

4. A vendor scores 95 on CVE density but 20 on patch response time. Explain why the overall grade may still be low despite a few vulnerabilities. How would you weigh the dimensions differently for a healthcare deployment vs. a consumer IoT deployment?
5. Walk through the SLSA Level 3 requirements. Why is non-falsifiable provenance harder to achieve than signed provenance? Give an example of a build pipeline that satisfies Level 2 but fails Level 3.
6. Design a firmware phylogenetic analysis for detecting shared SDK lineage across 5 camera vendors. What would the tree look like if all 5 vendors used the same Realtek SDK? How would you distinguish legitimate SDK reuse from counterfeit firmware cloning?
7. Design an agentic provenance assessor for a device that has conflicting firmware, hardware, and paperwork signals. What evidence should it gather before using the word "counterfeit," and what verifier should separate supplier sloppiness from supply-chain compromise?

References

1. SPDX Specification v2.3, ISO/IEC 5962:2021. https://spdx.dev/specifications/
2. CycloneDX v1.5 Specification, OWASP Foundation. https://cyclonedx.org/specification/overview/
3. EU Cyber Resilience Act, Regulation (EU) 2024/2847. Official Journal of the European Union.
4. SLSA v1.0 Specification. https://slsa.dev/spec/v1.0/
5. in-toto Specification v0.9. https://github.com/in-toto/specification
6. Cui, A., & Stolfo, S. J. (2010). "A Quantitative Analysis of the Insecurity of Embedded Network Devices." *ACSAC 2010*.
7. Thompson, K. (1984). "Reflections on Trusting Trust." *Communications of the ACM*, 27(8), 761-763.
8. NIST SSDF (SP 800-218): Secure Software Development Framework.
9. Ohm, M., et al. (2020). "Backstabber's Knife Collection: A Review of Open Source Software Supply Chain Attacks." *DIMVA 2020*.
10. SolarWinds SEC Filing (2021). "Sunburst: Timeline and Attribution."

Public sources used for the added operational layer

- **Public-source-derived:** CISA Software Bill of Materials resources. URL: https://www.cisa.gov/sbom
- **Public-source-derived:** CISA 2025 Minimum Elements for a Software Bill of Materials. URL: https://www.cisa.gov/resources-tools/resources/2025-minimum-elements-software-bill-materials-sbom
- **Public-source-derived:** European Commission summary of Regulation (EU) 2024/2847, Cyber Resilience Act. URL: https://digital-strategy.ec.europa.eu/en/policies/cra-summary
- **Public-source-derived:** CISA Secure-by-Design guidance. URL: https://www.cisa.gov/resources-tools/resources/secure-by-design
- **Public-source-derived:** NIST SP 800-53 Rev. 5, Security and Privacy Controls. URL: https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final
- **Public-source-derived:** CISA Cross-Sector Cybersecurity Performance Goals. URL: https://www.cisa.gov/cybersecurity-performance-goals