SECURITY & COMPLIANCE REFERENCE — 2026

The Developer's Guide to Security Compliance

What Developer Teams Need to Know About End-of-Life Open Source Software across the Top 20 Global Security Standards, Frameworks, and Regulations

SECURITY & COMPLIANCE REFERENCE — 2026

The Developer's Guide to Security Compliance

What Developer Teams Need to Know About End-of-Life Open Source Software across the Top 20 Global Security Standards, Frameworks, and Regulations

By **Javeir Perez**

Published by HeroDevs, Inc. · herodevs.com

CONTENTS

1 EXECUTIVE SUMMARY

Compliance used to be someone else's problem. A Governance, Risk Management, and Compliance (GRC) team filled out questionnaires, an auditor showed up once a year, and developers kept shipping. That era is over. Across the world, regulators and standards bodies have converged on a set of security-related obligations that land squarely on the teams writing the code and shipping applications: engineering teams.

There are common principles that appear in virtually every major framework and regulation:

- Maintain an accurate inventory of software components, increasingly in the form of a Software Bill of Materials (SBOM).
- Remediate known vulnerabilities within defined timeframes, often 30 days or less for critical severity.
- Demonstrate a documented, repeatable process for identification, prioritization, and remediation of vulnerabilities.

End-of-life (EOL) software fails all three by definition. When a framework version, runtime, or library stops receiving security patches, every subsequent CVE becomes permanently unfixed through normal channels. No patch is coming. That single fact is what turns "we are running AngularJS" or "we are still on Node.js 16" from a technical debt conversation into a compliance conversation.

This whitepaper walks through each major standard, framework, directive, and regulation, focusing on four things developers actually control: vulnerability handling, open source software, supported versions, and end-of-life software. For each one, it provides concrete examples and practical recommendations.

2 UNITED STATES OF AMERICA

2.1 PCI DSS 4.x (Payment Card Industry Data Security Standard)

> **Applies to:** Every entity that stores, processes, or transmits payment card data. Contractually mandatory, enforced by card brands and acquiring banks.

What it requires. Requirement 6.3.1 requires organizations to identify new security vulnerabilities using industry-recognized sources and assign risk rankings. Requirement 6.3.2 requires an inventory of bespoke and custom software, including third-party and open source components incorporated into it. Requirement 6.3.3 requires patches for critical or high-severity vulnerabilities be installed within one month of release, with all other applicable patches installed within a timeframe the entity defines (commonly three months). Requirement 11.3.1 layers on internal vulnerability scans at least quarterly, and 11.3.1.1 requires a targeted risk analysis for vulnerabilities that are not critical or high. To round all those requirements up, Requirement 12.3.4 mandates an annual review of all hardware and software in use, confirming technologies still receive vendor security fixes, support ongoing compliance, and are tracked for end-of-life announcements, with a senior-management-approved plan to remediate outdated and end-of-life technologies.[1, 2]

Why EOL software breaks it. The one-month clock in 6.3.3 assumes a patch exists. When a critical CVE is published against an EOL framework, no vendor or open source project patch will ever ship, so the requirement becomes structurally impossible to satisfy. An assessor who finds an EOL component with a known critical CVE in the cardholder data environment has found non-compliance, not a finding to negotiate.[3]

Example. A payments platform runs its checkout front end on AngularJS 1.x, which reached end of life in January 2022.[4] A new critical CVE is published against AngularJS. Under 6.3.3, the organization has one month to install a patch that does not exist. Its realistic options are an emergency migration (rarely achievable in 30 days for a production checkout flow), documented compensating controls that an assessor must accept, or commercial extended support that delivers a backported patch and restores a defensible remediation path.

Recommendations for developers:

- Maintain the 6.3.2 component inventory as a build artifact, not a spreadsheet. Generate an SBOM in CI on every release so the inventory is always current.
- Track EOL dates for every framework and runtime in the production environment, and treat an approaching EOL date as a compliance deadline, not a backlog item.
- For components already past EOL, either isolate them outside the production environment, migrate them, or place them under a commercial extended support agreement so that critical CVEs can still be patched inside the 30-day window.

2.2 Health Insurance Portability and Accountability Act (HIPAA) Security Rule

> **Applies to:** Covered entities and business associates handling electronic protected health information (ePHI) in the United States. Mandatory federal law.

What it requires. The Security Rule requires an accurate and thorough assessment of risks and vulnerabilities to ePHI (45 CFR 164.308(a)(1)(ii)(A))[5] and security measures sufficient to reduce those risks to a reasonable and appropriate level. HHS guidance treats risk analysis and vulnerability management as foundational, and the Office for Civil Rights' (OCR) January 2026 cybersecurity newsletter explicitly states that risk analysis must identify vulnerabilities such as unpatched software and pair that identification with active remediation.[6] A December 2024 Notice of Proposed Rulemaking would go further, adding explicit vulnerability management and patch management standards, vulnerability scanning at least every six months, annual penetration testing, and proposed patch timelines as short as 15 days for critical vulnerabilities.[7, 8] As of mid-2026 the NPRM has not been finalized and final action has slipped, but OCR is actively enforcing the existing rule, and its enforcement posture already targets organizations that document risks and then fail to act on them.[9]

Why EOL software breaks it. HIPAA never says "no EOL software." It says safeguards must be reasonable and appropriate. After a breach, running an unsupported component with known, unpatchable CVEs in a system touching ePHI is extremely difficult to defend as reasonable. OCR investigations repeatedly find the same vulnerabilities appearing in risk assessments year after year, unmitigated, until exploited; that pattern is what draws willful-neglect findings.

Example. A patient portal runs on an EOL version of Drupal (v7). The annual risk analysis flags it, the finding is documented, and nothing happens for two years because the migration is expensive. A breach occurs through a known CVE. The documented-but-ignored finding becomes the centerpiece of the OCR investigation, converting a technical shortcut into evidence of neglect.

Recommendations for developers:

- Ensure every EOL component that touches ePHI appears in the formal risk analysis with a remediation plan and an owner. A risk analysis that omits unpatched software is now explicitly deficient under OCR guidance.
- Close the loop: for each identified vulnerability, record the remediation action taken (patch, upgrade, isolation, extended support), because OCR enforcement now focuses on what organizations did about identified risks.
- Do not wait for the NPRM to finalize. Its proposed controls (scanning cadence, patch timelines, asset inventory) describe what regulators already consider reasonable security in 2026.

2.3 FedRAMP and NIST SP 800-53 SI-2 (Flaw Remediation)

> **Applies to:** All cloud services sold to US federal agencies. Mandatory.

What it requires. FedRAMP baselines are built from NIST SP 800-53 controls. Control SI-2 requires organizations to identify, report, and correct system flaws and to install security-relevant software updates within defined timeframes.[10] FedRAMP operationalizes this with severity-based remediation timelines: 30 days for high, 90 days for moderate, and 180 days for low-severity findings, tracked through the monthly Plan of Action and Milestones (POA&M) process and continuous monitoring scans.[11]

Why EOL software breaks it. Security scanning tools flag CVEs; an EOL component with a high-severity CVE creates a POA&M item with a 30-day clock and no patch to install. Findings that age past their window without remediation put the Authority to Operate (ATO) at risk. Unlike a commercial audit, this is not an annual event: scans run monthly or more often, so EOL software generates a continuously regenerating stream of overdue findings.

Example. A FedRAMP-authorized SaaS provider discovers its reporting service depends on an EOL version of Express. Every security scan re-flags the component, each new CVE opens a new 30- or 90-day POA&M item, and the Third-Party Assessment Organization (3PAO) assessor lists the unsupported dependency as a persistent weakness. The provider must show either a credible migration milestone or vendor-backed support that restores patch availability.

Recommendations for developers:

- Treat the FedRAMP 30/90/180-day clocks as engineering SLAs and wire them into your issue tracker so vulnerability tickets carry due dates automatically.
- Eliminate or gain supported coverage for EOL components before the initial assessment. It is far cheaper than defending recurring POA&M items every month afterward.
- Keep deviation requests and risk adjustments honest and rare; assessors track patterns, and habitual deviation requests around the same unsupported component signal an unmanaged risk.

2.4 NIST Cybersecurity Framework (CSF) 2.0

> **Applies to:** Voluntary for most private organizations; mandatory for US federal agencies and widely required of their contractors. Also the de facto reference model for cyber insurance and board reporting.

What it requires. CSF 2.0 subcategory PR.PS-02 states that software must be "maintained, replaced, and removed commensurate with risk."[12] ID.RA subcategories require that vulnerabilities in assets be identified, validated, and recorded. The framework is outcome-based: it does not prescribe patch windows, but it establishes the expectation that every piece of software in the environment is either actively maintained or deliberately retired based on risk.

Why EOL software breaks it. EOL software is by definition no longer maintained. Under PR.PS-02, an organization has exactly three defensible states for such software: replace it, remove it, or restore maintenance for it. Silently continuing to run it is the one state the framework does not permit. This is also the framework where commercial extended support maps most cleanly, because vendor-backed patching of an EOL component literally constitutes the "maintained" outcome.

Example. A manufacturer aligning to CSF 2.0 for its cyber insurance renewal inventories its plant-floor applications and finds a scheduling tool built on an EOL Vue 2 front end and an EOL version of Node.js runtime. The insurer's questionnaire asks whether all software is supported and patched. Answering honestly requires either a migration plan with dates or a support contract; answering dishonestly voids coverage arguments after an incident.

Recommendations for developers:

- Map every application to one of three states: actively maintained by the upstream project, maintained through commercial extended support, or scheduled for replacement/removal with a date. Anything unmapped is a PR.PS-02 gap.
- Feed EOL and support-status data into the same asset inventory used for CSF profiling so risk owners see software lifecycle status alongside criticality.
- Record risk decisions. CSF is evidence-driven: the difference between "we accept this risk until Q3 migration" and "nobody looked" is documentation.

2.5 NIST SSDF (SP 800-218, Secure Software Development Framework)

> **Applies to:** Voluntary framework, OMB memorandum M-26-05[13] rescinded the mandatory requirement of self-attestation for software vendors selling to US federal agencies, previously required by Executive Order 14028 and OMB memoranda M-22-18/M-23-16. Agencies may still choose to require attestations or SBOMs based on their own risk assessments.[14, 15, 16]

What it requires. The SSDF serves as a flexible framework focused on shifting security to the left, meaning security is baked into the entire software creation process. It requires organizations to define and maintain secure development policies (PO practices), protect software (PS), produce well-secured software (PW), and respond to vulnerabilities (RV). Practice PW.4 covers reusing well-secured software, requiring organizations to evaluate and monitor third-party and open source components. The RV practice group requires identifying, assessing, and remediating vulnerabilities in released software on an ongoing basis, which presumes the components involved can still receive fixes.

Why EOL software breaks it. A vendor cannot attest to SSDF-conformant vulnerability response for a product built on components that no longer receive security updates. PW.4's expectation of acquiring components from maintained sources, and RV's expectation of ongoing remediation, both collapse when the underlying open source project has ended support.

Example. A software company selling to a federal agency voluntarily signs the CISA secure software development attestation form. Its flagship product still ships with jQuery 1.x and an EOL Bootstrap version. Legal asks engineering whether the attestation is accurate. Engineering either scrambles a remediation program or the company attests to something its dependency tree contradicts, which is a false-statement risk, not merely a security risk.

Recommendations for developers:

- Establish an open source intake policy that records the support status and EOL horizon of every new dependency before adoption, not after.
- Monitor dependencies for lifecycle events (maintenance mode, archived repositories, announced EOL dates), not only for CVEs. A project's end-of-life support is a security event.
- Before any federal attestation, run a dependency lifecycle audit and remediate or gain supported coverage for every EOL component the product ships.

2.6 NIST SP 800-171 and Cybersecurity Maturity Model Certification (CMMC)

> **Applies to:** DoD contractors and federal suppliers handling Controlled Unclassified Information (CUI). Mandatory, with CMMC assessments now phasing into DoD contracts.

What it requires. Requirement 3.14.1 (renumbered 03.14.01 in Rev 3) requires identifying, reporting, and correcting system flaws in a timely manner. The assessment guidance points directly to installed patches, service packs, and hotfixes as the evidence of correction.[17] Related requirements cover vulnerability scanning and remediation (3.11.2, 3.11.3).

Why EOL software breaks it. "Correct flaws in a timely manner" has no exception for software whose vendor or open source project has stopped issuing corrections. An EOL component with known CVEs inside a CUI enclave is a standing failure of 3.14.1 that a CMMC assessor will score against the contractor, and a single failed practice can cost certification and therefore contract eligibility.

Example. A defense supplier's engineering document portal runs on an EOL web framework. During CMMC Level 2 assessment, the assessor asks for evidence that flaws in the portal are corrected timely. The supplier can show scan reports identifying the CVEs but no corrections, because none exist upstream. The practice scores as not met, jeopardizing the certification the company needs to keep its DoD contracts.

Recommendations for developers:

- Scope aggressively: keep EOL software out of the CUI boundary entirely where possible, because everything inside the boundary is assessed.
- Where an EOL component must remain in scope, document compensating controls and pair them with a supported patch source so "timely correction" remains demonstrable.
- Keep patch evidence (tickets, deploy logs, before/after scans) organized by requirement; CMMC assessments are evidence exercises, and undocumented fixes score the same as no fixes.

2.7 SEC Cybersecurity Disclosure Rules

> **Applies to:** All publicly traded companies in the United States. Mandatory, effective December 2023.

What it requires. Registrants must disclose material cybersecurity incidents on Form 8-K within four business days of determining materiality, and must annually disclose (Regulation S-K Item 106) their processes for assessing, identifying, and managing material cybersecurity risks, including board oversight[18] with cybersecurity risk management, strategy, and governance in their annual reports (Form 10-K).

Why EOL software matters here. The rules do not mention patching or EOL software. Materiality does. A known, unpatched critical vulnerability in a component the business depends on can itself constitute a material risk requiring disclosure, and an incident traced to a known-EOL system is precisely the fact pattern that turns a breach into securities litigation, because plaintiffs will argue the risk was known, unremediated, and inadequately disclosed.

Example. A financial services enterprise suffers a breach through a Java-based app using EOL Spring framework. During incident response, counsel discovers internal tickets flagging the EOL status six months earlier. The four-day disclosure clock is now the easy part; the hard part is explaining in the 8-K and subsequent filings why a documented, known risk was never remediated, with the board's oversight processes under direct scrutiny.

Recommendations for developers:

- Understand that internal engineering records (tickets, Slack threads, risk register entries) about EOL software are discoverable and will be read against public disclosures after an incident.
- Give security and legal teams accurate lifecycle data: an honest inventory of EOL exposure lets the company describe its risk management process truthfully in Item 106 disclosures.
- Prioritize remediation of EOL components in revenue-critical paths, because that is where "material" and "unpatched" intersect.

3 CANADA

3.1 CCSPA (Critical Cyber Systems Protection Act, Bill C-8)

> **Applies to:** Designated operators in federally regulated critical infrastructure: telecommunications, banking, energy, and transportation. Mandatory. Bill C-8 received Royal Assent on June 15, 2026, and the CCSPA's obligations are coming into force on a phased basis by order in council.[19, 20]

What it requires. Designated operators must establish and implement a cybersecurity program, mitigate supply-chain and third-party risks, report cybersecurity incidents on a defined timeline, comply with cybersecurity directions, and keep records. Penalties reach up to CAD $15 million for organizations, and continuing violations can count separately for each day they continue.[21] Board directors and corporate officers can be held personally liable if they directed, authorized, or acquiesced in the organization's non-compliance.

Why EOL software breaks it. The supply-chain risk obligation reaches directly into the software stack. An unpatched open source library or EOL framework inside a critical cyber system is exactly the class of third-party risk the program must identify and mitigate, and with per-day penalty exposure, "we knew and did nothing" is an expensive posture. CCSPA requires programs that include routine vulnerability assessments, system scanning, and active mitigation plans to patch or isolate weaknesses before they can be exploited.

Example. A federally regulated energy operator's outage-management portal runs on an EOL Angular version. Once designated under the CCSPA, the operator's cybersecurity program must document supply-chain risks. The EOL front end appears in the assessment; the operator must show mitigation, which in practice means migration on a committed timeline, isolation, or restored vendor support with patch SLAs.

Recommendations for developers:

- If your employer operates in Canadian telecom, finance, energy, or transport industries, start the software lifecycle inventory now; designations and regulations are phasing in, and a program built after designation will be built in a hurry.
- Treat open source dependencies as suppliers for CCSPA purposes: record who maintains each critical component, its support status, and how security fixes reach you.
- Build incident-reporting readiness into engineering practice (log retention, component-level blast-radius mapping) so a report can name affected systems and components quickly.

4 EUROPEAN UNION

4.1 DORA (Digital Operational Resilience Act)

> **Applies to:** EU financial entities (banks, insurers, investment firms, payment institutions, crypto-asset providers) and critical ICT third-party providers. Mandatory, in application since January 17, 2025.

What it requires. DORA's ICT risk management framework requires financial entities to use ICT systems that are reliable, have sufficient capacity, and are technologically resilient, to maintain an updated inventory of all ICT assets (including tracking of assets approaching end of life), to implement patch management, and to identify and document legacy ICT systems with a view to phasing them out or managing them explicitly.[22]

Why EOL software breaks it. DORA is unusually direct: unsupported systems are treated as a fundamental flaw in operational resilience, and the asset-inventory obligation specifically anticipates EOL tracking. A financial entity running EOL software in payment or trading paths is carrying a documented resilience defect that supervisors and their auditors are instructed to look for.

Example. An EU payment institution's transaction dashboard runs on Vue 2, EOL since December 2023. The DORA asset inventory must record this, its criticality, and the plan. During a supervisory review, "no plan" is a finding; "migration scheduled for 2027 with commercial security patches covering the interim" is a managed legacy system.

Recommendations for developers:

- Add EOL dates and support status as mandatory fields in the ICT asset inventory, and automate their population from your dependency manifests.
- For each legacy or EOL component in a critical or important function, document one of three plans: migrate by a date, decommission by a date, or keep it patched until one of the first two happens.
- Test resilience assumptions: DORA expects digital operational resilience testing, and known-vulnerable EOL components are the first thing a threat-led penetration test will exploit.

4.2 GDPR (General Data Protection Regulation), Article 32

> **Applies to:** All organizations processing EU personal data, regardless of where the organization is located. Mandatory.

What it requires. Article 32 requires "appropriate technical and organisational measures" to ensure a level of security appropriate to the risk, explicitly taking into account "the state of the art."[23] It does not mandate patching, timelines, or specific technologies.

Why EOL software breaks it. The phrase that matters is "state of the art." After a personal data breach, the supervisory authority asks whether the measures in place were appropriate given available technology. Running software with a publicly disclosed CVE, no available patch, and a long-expired support window is very difficult to defend as state of the art, and European regulators have repeatedly fined organizations following breaches through known, unpatched vulnerabilities.

Example. A marketing SaaS processing EU customer data is breached through a known CVE in an EOL jQuery plugin chain. The Article 32 analysis is not about whether the company was unlucky; it is about whether continuing to run an unsupported component with public exploits was "appropriate." Fines under GDPR can reach 4% of global annual turnover where basic-principle violations are also found, and breach-through-known-vulnerability is a classic aggravating pattern (negligence and failure of technical measures).

Recommendations for developers:

- Prioritize vulnerability remediation on systems processing personal data, and be able to show the prioritization logic.
- When a component goes EOL, record the risk decision at the time (migrate, isolate, support contract), because contemporaneous documentation is the strongest Article 32 defense.
- Practice data minimization in legacy systems: the less personal data an EOL system touches, the smaller its Article 32 exposure while migration proceeds.

4.3 NIS2 Directive

> **Applies to:** Essential and important entities across 18 sectors in the EU (energy, transport, health, digital infrastructure, manufacturing of critical products, digital providers, and more). Mandatory; transposition into member-state law was due October 2024 and enforcement is ramping across member states on national timelines.

What it requires. Article 21(2) mandates ten minimum risk-management measures, including (e) "security in network and information systems acquisition, development and maintenance, including vulnerability handling and disclosure" and (d) supply chain security covering the relationships with direct suppliers and service providers. Article 21(3) requires entities to take into account the vulnerabilities specific to each supplier and the overall quality and secure development practices of suppliers' products.[24, 25] Article 23 sets incident reporting at 24 hours (early warning), 72 hours (notification), and one month (final report). Fines reach EUR 10 million or 2% of worldwide turnover for essential entities, and NIS2 adds personal accountability for management bodies.[26]

Why EOL software breaks it. Article 21(2)(e) makes vulnerability handling in development and maintenance a legal obligation, and EOL software is software for which vulnerability handling has structurally ended. Supply-chain measures under 21(2)(d) and 21(3) extend the same logic to open source components: an entity must account for the security practices behind the software it depends on, and a dead upstream project has none.

Example. A hospital group (essential entity) runs its staff scheduling system on an EOL Nuxt 2 application. A ransomware incident enters through a known CVE in that stack. Beyond the 24-hour reporting clock, the national authority's investigation examines whether Article 21 measures were in place; a known-EOL, known-vulnerable system with no documented handling is a direct 21(2)(e) failure, and under NIS2 the management board carries personal responsibility for approving the risk-management measures.

Recommendations for developers:

- Implement a vulnerability handling process that explicitly covers open source dependencies: identification (SCA scanning), triage, remediation SLAs, and disclosure handling.
- Maintain SBOMs for the systems supporting essential or important services so supply-chain questions can be answered with data.
- Escalate EOL exposure to management in writing. NIS2's personal-accountability provisions mean leadership must know; giving them accurate lifecycle data is part of the engineering job.

4.4 Cyber Resilience Act (CRA)

> **Applies to:** Manufacturers, importers, and distributors of "products with digital elements" placed on the EU market, hardware and software alike, regardless of where the manufacturer is based. Mandatory. Reporting obligations for actively exploited vulnerabilities and severe incidents apply from September 11, 2026; the full essential requirements, conformity assessment, and CE marking apply from December 11, 2027. Penalties reach EUR 15 million or 2.5% of global turnover.[27, 28]

What it requires. This is the most consequential law on this list for developers, because it regulates the product itself across its lifecycle. Annex I Part I requires products to be delivered without known exploitable vulnerabilities. Annex I Part II sets vulnerability handling requirements: manufacturers must identify and document vulnerabilities and components, including drawing up an SBOM in a machine-readable format covering at least top-level dependencies; address vulnerabilities without delay through free security updates; apply effective and regular security testing; and publicly disclose fixed vulnerabilities. These obligations run for the support period, which must be at least five years in most cases.[29] Article 14 requires reporting actively exploited vulnerabilities to ENISA and the national CSIRT within 24 hours (early warning), 72 hours (full notification), and 14 days after a fix is available (final report). Critically, the September 2026 reporting obligation applies to products already on the market, including legacy products shipped years ago.[30, 31]

Why open source and EOL software matter enormously here. A commercial vendor that ships open source components is legally responsible for vulnerabilities in those components. If an actively exploited vulnerability surfaces in an open source library inside your shipped product, the reporting clocks apply to you, and the obligation to remediate "without delay" applies to you, regardless of whether the upstream project still exists. Shipping a product built on an EOL framework means committing, under law, to handle vulnerabilities in a component whose upstream will never handle them again. The CRA converts "ignored CVE in an old dependency" from technical debt into legal liability with the largest fine ceilings in technology regulation.

Example. An ISV sells a document management product into the EU. The product embeds an EOL Bootstrap version and an EOL Express server. In October 2026, a CVE in that Express version appears in active exploitation. Under Article 14 the vendor has 24 hours to file an early warning, 72 hours for full notification, and 14 days after a fix exists to file the final report. There is no upstream fix. The vendor must either produce its own backported patch, buy one through extended commercial support, or explain to regulators why a fix is not coming for a product it is still selling.

Recommendations for developers:

- Generate and maintain machine-readable SBOMs (CycloneDX or SPDX) for every shipped product now; you cannot meet a 24-hour reporting clock in September 2026 without already knowing what is in each product version.
- Audit every product line for EOL components and resolve each one before December 2027: upgrade, remove, or secure a support arrangement that can deliver patches during your declared support period.
- Define and publish a realistic support period per product and build the patching cost of that commitment (including third-party components) into product planning; the support period is now a legal commitment, not a marketing statement.
- Stand up a coordinated vulnerability disclosure policy and an intake-triage-report workflow rehearsed against the 24/72-hour clocks.

5 UNITED KINGDOM

5.1 UK GDPR

> **Applies to:** Organizations processing UK personal data. Mandatory, enforced by the Information Commissioner's Office (ICO).

What it requires. Retained from the EU GDPR after Brexit, UK GDPR carries the same Article 32 obligation of "appropriate technical and organisational measures." The ICO's enforcement record makes the interpretation concrete: failing to patch a known vulnerability is treated as a failure of appropriate security, and breaches traced to unsupported or long-unpatched software are viewed as negligence rather than misfortune.[32]

Example. The ICO found that the Chartered Institute for Securities & Investment (CISI) breached Article 32 by running website software that had reached EOL; in addition to that, attackers exploited a critical vulnerability for which a patch had been available since 2017 but was never applied.[33]

Recommendations for developers:

- Apply the same discipline as EU GDPR: inventory, prioritized patching for systems touching personal data, and contemporaneous documentation of risk decisions on EOL components.
- Where a legacy system cannot be migrated quickly, reduce its personal-data footprint and isolate it, and keep evidence of both.
- Track ICO guidance and enforcement notices; they function as an ongoing case-law record of what "appropriate" means in practice.

5.2 UK NIS Regulations 2018

> **Applies to:** Operators of Essential Services (water, energy, transport, health) and Relevant Digital Service Providers (cloud, online marketplaces, search). Mandatory, legally binding.

What it requires. Operators must take appropriate and proportionate measures to manage risks to network and information systems and to prevent and minimize the impact of incidents, and must report significant incidents within 72 hours.[34] The NCSC's Cyber Assessment Framework (CAF), used by UK regulators to assess compliance, includes objectives on managing vulnerabilities and keeping systems supported; in practice, patch-timeliness expectations often align with the Cyber Essentials benchmark of applying critical updates within 14 days.[35, 36]

Why EOL software breaks it. If an incident that disrupts an essential service traces back to an unpatched EOL system, the regulator will likely find a breach of the duty to maintain service continuity, and the 72-hour report becomes the opening document of an enforcement case. A 14-day patch expectation is unachievable for software with no patch source.

Example. A UK water utility's telemetry dashboard runs on an EOL AngularJS build. A known CVE is exploited and operational visibility is lost for a day. The incident is reportable within 72 hours, and the CAF assessment that follows examines vulnerability management; an unsupported component with a public CVE and no patch pathway scores poorly across multiple CAF outcomes.

Recommendations for developers:

- Map essential-service systems against the CAF and flag every unsupported component; the CAF explicitly examines whether systems are supported and vulnerabilities managed.
- Establish a patch source (upstream or commercial) capable of meeting a 14-day critical-patch expectation for everything in the essential-service estate.
- Rehearse the 72-hour reporting flow with engineering involved, so component-level facts (what was vulnerable, since when, what has been done) are available fast.

6 ASIA-PACIFIC

6.1 APPI (Japan, Act on the Protection of Personal Information)

> **Applies to:** Organizations handling personal information of individuals in Japan. Mandatory.

What it requires. The APPI requires business operators to take necessary and appropriate action for the security management of personal data, preventing leakage, loss, or damage. The Personal Information Protection Commission's guidelines elaborate this into organizational, human, physical, and technical safeguards; the technical safeguards encompass keeping software maintained, applying security patches regularly, and monitoring for vulnerabilities.[37]

Why EOL software breaks it. As with GDPR, the statute does not name EOL software, but regulators clearly expect the use of supported and maintained software. A leak of personal data through a known vulnerability in unsupported software is difficult to square with "necessary and appropriate" security management, and mandatory breach notification (introduced in the 2022 amendments) ensures such incidents reach the regulator.

Example. A Japanese e-commerce operator suffers a data leak through an EOL version of a JavaScript framework in its account portal. The mandatory report to the PPC triggers scrutiny of technical safeguards, and "the component had been unsupported for three years" becomes the central finding.

Recommendations for developers:

- Treat PPC guidance on technical safeguards as the operational baseline: regular patching, vulnerability monitoring, and supported software for any system holding personal data of individuals in Japan.
- Include Japanese-market systems in the same lifecycle inventory and remediation SLAs used for EU/US compliance rather than maintaining a separate, weaker standard.
- Document safeguard decisions in a form that can support a breach report, since notification is now mandatory and the follow-up questions are predictable.

6.2 Active Cyber Defense Law (Japan, "ACD Law" or "ACDA")

> **Applies to:** Specified essential infrastructure providers designated under Japan's Economic Security Promotion Act, plus the IT vendors that supply them. Mandatory. Enacted May 16, 2025, with phased implementation reaching full effect by 2027.

What it requires. The ACDA restructures Japan's cyber defense around four pillars: public-private collaboration, use of communications data for threat detection, government neutralization of attacker infrastructure, and organizational reform. Designated providers must report cybersecurity incidents to the competent authorities, an obligation taking effect on October 1, 2026 (not yet formally fixed). For vulnerabilities affecting Critical Systems, the government may notify IT vendors and the competent minister may request corrective actions; the requests are not binding, but vendors must make reasonable efforts to respond. The framework also enables the government to promptly request that operators address zero-day vulnerabilities.[38, 39]

Why EOL software breaks it. The ACDA creates a direct channel between regulators and the software inside critical infrastructure, and EOL software fails on both ends. Advance notification makes the composition of critical systems visible to regulators before deployment, so an unsupported framework enters that review as a documented weakness. The remediation mechanism presumes correction is possible: when the minister requests corrective action on a CVE in an EOL component, no upstream fix exists, and "reasonable efforts" collapse into producing a patch independently or admitting none is coming. Mandatory incident reporting then guarantees that a breach through a known CVE in unsupported software reaches the authorities on a defined timeline.

Example. A designated electric power operator runs its outage-reporting portal on an EOL .NET version. The NCO identifies active exploitation of a CVE in that .NET version and the competent minister requests corrective action from the operator's IT vendor. No upstream patch exists. The vendor's options are an emergency migration or commercial extended support that restores a patch source; "the component is end-of-life and cannot be corrected" is now on the record with the ministry ahead of any future incident report.

Recommendations for developers:

- If your systems serve designated Japanese infrastructure operators, inventory every framework and runtime now and resolve EOL components before the notification and reporting obligations bite in November 2026.
- Maintain a patch source for every component in a Critical System, whether upstream, in-house, or commercial extended support, so government remediation requests can be answered on short notice.
- Treat the ACDA and APPI as complementary exposure: APPI governs personal data leaks, the ACDA governs critical service disruption, and one incident in a Japanese-market system can trigger both.

6.3 SOCI Act (Australia, Security of Critical Infrastructure Act 2018)

> **Applies to:** Owners and operators of critical infrastructure assets across 11 sectors in Australia. Mandatory.

What it requires. The SOCI Act and its Critical Infrastructure Risk Management Program (CIRMP) rules require responsible entities to identify material risks (including cyber and supply-chain hazards) and implement controls to minimize or mitigate them, with annual board-approved reporting.[40] Many entities satisfy the cyber framework obligation by adopting the ACSC Essential Eight, whose patching maturity levels expect critical patches within short, defined windows and which explicitly treats unsupported software as a gap to be removed or replaced.[41]

Why EOL software breaks it. EOL software is not named in the Act, but it is clearly in scope through the risk-management obligation: an unsupported component in a critical system is a material cyber risk requiring documented controls or remediation, and boards must attest annually to the program that addresses it.

Example. An Australian port operator's logistics application runs on an EOL Node.js version. The CIRMP annual report, signed by the board, must describe material risks and mitigations. The EOL runtime either appears with a credible mitigation (migration schedule, isolation, or extended support with patch delivery) or its omission becomes the finding when an incident or audit surfaces it.

Recommendations for developers:

- Align patching and lifecycle practices with the Essential Eight maturity model, which most Australian regulators and boards use as shorthand for "reasonable."
- Surface EOL exposure into the CIRMP risk register with proposed treatments, because board attestation makes unreported engineering risk a governance problem.
- For operational-technology-adjacent systems that genuinely cannot be upgraded, document segmentation and monitoring controls explicitly as risk treatments.

7 GLOBAL STANDARDS

7.1 Center for Internet Security (CIS) Controls v8.1

> **Applies to:** Voluntary best-practice framework, widely adopted, referenced by regulators and insurers, and mandatory in some contractual contexts.

What it requires. Control 2 (Inventory and Control of Software Assets) requires organizations to actively manage all software so that only authorized, supported software is installed and executed; the safeguards under Control 2 explicitly call for ensuring software is supported and for addressing unsupported software. Control 7 (Continuous Vulnerability Management) requires continuously assessing and tracking vulnerabilities and establishing remediation processes with defined cadence.[42]

Why EOL software breaks it. CIS is one of the few frameworks that names the problem directly: unsupported software is treated as inherently vulnerable and must be documented as an exception with mitigating controls or removed. Organizations that benchmark against CIS (or whose customers and insurers do) will fail internal audits on any unmanaged EOL component.

Example. A mid-market SaaS company adopts CIS Controls to answer enterprise security questionnaires. Its Control 2 software inventory reveals four applications on EOL frameworks. Each must be listed as a documented exception with compensating controls and a sunset date, or moved to supported status; leaving them unlisted fails the control and, worse, falsifies the questionnaire answers built on it.

Recommendations for developers:

- Automate the Control 2 inventory from package manifests and deployment tooling, and include a "supported until" field for every framework and runtime.
- Set remediation SLAs by severity under Control 7 and measure against them; the control expects a process with cadence, not ad hoc patching.
- Treat every EOL component as a formal exception requiring documentation, mitigation, and an expiry date, exactly as the control intends.

7.2 SOC 2 Trust Services Criteria

> **Applies to:** Voluntary attestation, but effectively mandatory for B2B SaaS and cloud vendors because enterprise customers demand the report contractually.

What it requires. The Trust Services Criteria (notably CC7.1) require monitoring for new vulnerabilities, periodic scanning, timely remediation, and a functioning patch and change management process. Auditors test whether vulnerabilities identified during the audit period were remediated within the organization's own stated SLAs and whether the process operated consistently.[43]

Why EOL software breaks it. A SOC 2 Type II report covers a period of months. An EOL component with open CVEs sits in every scan across that period with no remediation possible, producing either an exception in the report or an increasingly creative narrative around it. Enterprise customers read exceptions; a vulnerability-management exception in a SaaS vendor's report is a sales problem as much as a security one.

Example. A SaaS vendor's Type II audit period runs January through June. In February, a high-severity CVE is published against the EOL open source tooling in its build pipeline and against an EOL Express dependency in an internal service. The vendor's own policy says high-severity findings are remediated in 30 days. The auditor samples those tickets in July. Without an upstream or commercial patch, the tickets are open at day 120, and the report ships with a noted exception.

Recommendations for developers:

- Write remediation SLAs you can actually meet, then meet them; auditors test against your own policy, and heroic policies with missed deadlines are worse than modest kept ones.
- Ensure every component in scope has a patch source before the audit period opens; EOL software discovered mid-period becomes a months-long documented failure.
- Keep vulnerability tickets clean and complete (identified date, severity, remediation, closure evidence), because they are the audit population.

7.3 ISO/IEC 27001:2022

> **Applies to:** Voluntary international certification for information security management systems, frequently made mandatory through customer contracts and procurement requirements.

What it requires. Annex A Control 8.8 (Management of Technical Vulnerabilities) requires obtaining information about technical vulnerabilities in a timely fashion, evaluating exposure, and taking appropriate measures. Control 8.9 (Configuration Management) requires establishing, documenting, implementing, and monitoring configurations, including baseline templates for hardware and software. Control 5.9 requires an inventory of information and associated assets.[44]

Why EOL software breaks it. Under 8.8, "appropriate measures" for a known vulnerability in an EOL component cannot include waiting for a patch that will not arrive; the organization must upgrade, isolate, mitigate, or restore support, and must show the decision trail. Under 8.9, EOL software is by definition outside any defensible security baseline, since no hardening template can compensate for a component that accumulates unfixable CVEs. Certification audits recur (surveillance audits annually, recertification every three years), so an unmanaged EOL component is a nonconformity waiting for each visit.

Example. A software consultancy holds ISO 27001 certification that its government clients require. During a surveillance audit, the auditor samples the vulnerability management records and finds recurring scanner findings against an EOL jQuery version in the client portal, each closed as "risk accepted" with no mitigation, no owner, and no end date. The auditor raises a nonconformity against 8.8: risk acceptance without evaluation, treatment, or time-bound review is not an "appropriate measure."

Recommendations for developers:

- Connect the asset inventory (5.9), configuration baselines (8.9), and vulnerability process (8.8) so that lifecycle status flows through all three; auditors increasingly test the seams between controls.
- Make risk acceptances for EOL components time-bound, owned, and paired with real mitigations, and revisit them at defined intervals.
- Feed SCA and EOL scanning output directly into the ISMS's corrective-action process so evidence of "timely" identification and response accumulates automatically.

8 WHAT THIS ALL MEANS FOR DEVELOPERS: A CONSOLIDATED PLAYBOOK

Reading twenty frameworks or regulations side by side, the pattern is unmistakable. The specific clocks differ (30 days under PCI DSS 6.3.3 and FedRAMP high findings, 14 days under UK Cyber Essentials expectations, 24/72 hours for CRA and NIS2 reporting), but the underlying demands are identical, and all of them are engineering demands:

1. **Know what you ship and run.** SBOMs are no longer optional. The CRA makes them a legal requirement for products sold into the EU, PCI DSS 6.3.2 requires the component inventory, DORA requires the ICT asset inventory with EOL tracking, and CIS Control 2, ISO 27001 5.9, and NIST CSF all presume one. Generate SBOMs in CI, per release, in CycloneDX or SPDX, and keep them queryable.[45] When the next Log4Shell-class event happens, every regime on this list expects you to answer "are we affected, where, and since when" in hours.

2. **Treat EOL dates as compliance deadlines.** An open source project's announced end-of-life date is the date after which every new CVE in it becomes permanently unremediable through normal channels. Track EOL horizons for every runtime, framework, and major library (Node.js, Angular, AngularJS, Vue, React tooling, Express, Nuxt, Bootstrap, jQuery, Drupal, .NET, Spring, and the rest of the stack) with the same seriousness as certificate expiry.[46, 47, 48] Alert at 12 months out, plan at 9, act before the EOL day.

3. **Give every component exactly one of three statuses.** Actively maintained upstream; maintained through a commercial extended support arrangement that delivers security patches (which is what satisfies "maintained" under NIST CSF PR.PS-02, keeps PCI's 30-day clock meetable, and preserves FedRAMP POA&M timelines); or scheduled for replacement/removal with a committed date. Anything without one of these three statuses is the gap every auditor, assessor, and regulator on this list is trained to find.

4. **Put SLAs on remediation and instrument them.** Wire severity-based due dates into the issue tracker, measure mean time to remediate, and keep the closure evidence. SOC 2, ISO 27001, CMMC, and FedRAMP are all, at bottom, evidence exercises: the organizations that pass are the ones whose ordinary engineering workflow produces the audit trail as a byproduct.

5. **Prepare for reporting clocks that assume you already know.** The CRA's 24-hour early warning (from September 11, 2026, including for products already on the market), NIS2's 24-hour incident warning, and the SEC's four-business-day materiality clock all presuppose that component-level knowledge of your systems exists before the incident. Rehearse the flow: detection, component identification via SBOM, impact assessment, draft notification.

6. **Escalate lifecycle risk in writing.** NIS2 makes management bodies personally accountable, SEC rules put boards on the hook for oversight disclosures, and Australia's CIRMP requires board attestation. Leadership can only govern risks it can see. An honest, current picture of EOL exposure, with options (migrate, isolate, extend support), is one of the highest-leverage documents an engineering team can produce.

The direction of travel is one-way. The CRA's full application in December 2027, NIS2 enforcement maturing across member states, CMMC phasing into DoD contracts, Canada's CCSPA phasing in following Royal Assent, and the proposed HIPAA Security Rule overhaul all tighten the same three screws: inventory, timely remediation, documented process. Developers who build lifecycle awareness into their pipelines now will experience these regimes as paperwork. Those who do not will experience them as emergencies.

9 HOW HERODEVS NEVER-ENDING SUPPORT (NES) CLOSES THE EOL COMPLIANCE GAP

NES covers a portfolio of end-of-life open source frameworks, runtimes, and libraries that account for a large share of unresolved EOL exposure across organizations of all sizes, including AngularJS, Angular, Vue 2, Node.js, Spring, .NET, jQuery, Bootstrap, Express, Nuxt, and others. NES delivers continuous CVE remediation through a private, secure registry and a complete compliance artifact set: a signed support agreement, a published CVE remediation log, release notes, and releases suitable for SBOM inclusion.

Mapped against the obligations described in the preceding sections, NES converts the structurally impossible into demonstrable coverage:

- **Remediation clocks become meetable again.** NES restores a commercial patch source for components whose upstream will never ship another fix, which keeps those clocks satisfiable and keeps findings closeable.
- **"Maintained" becomes a truthful answer.** Under NIST CSF 2.0 PR.PS-02, software must be maintained, replaced, or removed commensurate with risk. Vendor-backed patching of an EOL component constitutes the maintained outcome.
- **Legacy systems become managed systems.** DORA and NIS2 do not prohibit legacy software; they prohibit unmanaged legacy software. An EOL component under an NES contract, with a committed patch SLA and a documented remediation log, makes legacy software secured and under commercial support.
- **Product-lifecycle commitments become sustainable.** Under the Cyber Resilience Act, a manufacturer shipping open source components is legally responsible for remediating vulnerabilities in them throughout the declared support period. NES provides the backported patches that make support commitment credible for products built on frameworks whose upstream support ended years earlier.

NES is not a substitute for migration where migration is the right answer. It is the mechanism that makes migration timelines honest. An organization can commit to a realistic modernization schedule, in writing, to its board, its auditors, and its regulators, while every intervening CVE is remediated on time and on the record.

The frequency of AI-discovered vulnerabilities are multiplying the risk of EOL software. The operational result of having NES is that an enterprise running EOL open source software in production can convert its answer to the auditor's question from a disqualifying disclosure into a documented and defensible control.

For current technology coverage, release notes and CVE remediation documentation, visit herodevs.com and https://docs.herodevs.com/guide/getting-started

REFERENCES

Sources are numbered in order of first appearance in the text. URLs are provided for reference.

1. PCI Security Standards Council, Payment Card Industry Data Security Standard (PCI DSS) v4.0.1, Requirements 6.3.1, 6.3.2, 6.3.3, and 11.3.1, June 2024. https://www.pcisecuritystandards.org/document_library/https://blog.basistheory.com/pci-dss-requirement-6
2. Risk Associates, How PCI DSS v4.0.1 Shifts the Rules on Identifying and Fixing Vulnerabilities, April 2026. https://riskassociates.com/blogs/how-pci-dss-v4-0-1-shifts-the-rules-on-identifying-and-fixing-vulnerabilities/
3. Endor Labs (with Schellman), An Auditor's Perspective on Addressing OSS Vulnerabilities for PCI DSS v4, January 2026. https://www.endorlabs.com/learn/an-auditors-perspective-on-addressing-oss-vulnerabilities-for-pci-dss-v4
4. Angular Team, Discontinued Long-Term Support for AngularJS, Google, January 2022. https://blog.angular.io/discontinued-long-term-support-for-angularjs-cc066b82e65a
5. U.S. Code of Federal Regulations, 45 CFR Part 164, Subpart C (HIPAA Security Rule). https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164
6. Medcurity, 2026 HIPAA Security Rule Update: New Requirements to Prepare For (summarizing the OCR January 2026 Cybersecurity Newsletter on risk analysis and unpatched software), June 2026. https://medcurity.com/hipaa-security-rule-2026-update/
7. U.S. Department of Health and Human Services, Office for Civil Rights, HIPAA Security Rule Notice of Proposed Rulemaking to Strengthen Cybersecurity for Electronic Protected Health Information. https://www.hhs.gov/hipaa/for-professionals/security/hipaa-security-rule-nprm/index.html
8. Federal Register, HIPAA Security Rule To Strengthen the Cybersecurity of Electronic Protected Health Information, 90 FR 898, January 6, 2025. https://www.federalregister.gov/public-inspection/2024-30983/health-insurance-portability-and-accountability-act-security-rule-to-strengthen-the-cybersecurity-of
9. Clearwater Security, HIPAA Security Rule Enforcement: Where Things Stand in 2026, July 2026. https://clearwatersecurity.com/blog/hipaa-security-rule-enforcement-2026/
10. National Institute of Standards and Technology, NIST SP 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations, Control SI-2 (Flaw Remediation). https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf
11. FedRAMP Continuous Monitoring Playbook. https://www.fedramp.gov/resources/documents/Continuous_Monitoring_Playbook.pdf
12. National Institute of Standards and Technology, The NIST Cybersecurity Framework (CSF) 2.0, NIST CSWP 29, Subcategory PR.PS-02, February 2024. https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf
13. Office of Management and Budget, United States, Memorandum to the heads of executive departments and agencies, M-26-05, January 23,2026,. https://www.whitehouse.gov/wp-content/uploads/2026/01/M-26-05-Adopting-a-Risk-based-Approach-to-Software-and-Hardware-Security.pdf
14. National Institute of Standards and Technology, NIST SP 800-218: Secure Software Development Framework (SSDF) Version 1.1, Practices PW.4 and RV.1–RV.3. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218.pdf
15. Cybersecurity and Infrastructure Security Agency, Secure Software Development Attestation Form. https://www.cisa.gov/secure-software-attestation-form

16. Federal Register, Executive Order 14028: Improving the Nation's Cybersecurity, May 12, 2021. https://www.federalregister.gov/documents/2021/05/17/2021-10460/improving-the-nations-cybersecurity
17. National Institute of Standards and Technology, NIST SP 800-171 Rev. 3: Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations, Requirement 3.14.1. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-171r3.pdf
18. U.S. Securities and Exchange Commission, Cybersecurity Risk Management, Strategy, Governance, and Incident Disclosure, Release Nos. 33-11216; 34-97989, July 2023. https://www.sec.gov/files/rules/final/2023/33-11216.pdf
19. Parliament of Canada, Bill C-8: An Act respecting cyber security, amending the Telecommunications Act and making consequential amendments to other Acts, Statutes of Canada 2026, c. 9, LEGISinfo. https://www.parl.ca/legisinfo/en/bill/45-1/c-8
20. Public Safety Canada, Government of Canada Strengthens Cyber Security and Critical Infrastructure with Royal Assent of Bill C-8, June 2026. https://www.canada.ca/en/public-safety-canada/news/2026/06/government-of-canada-strengthens-cyber-security-and-critical-infrastructure-with-royal-assent-of-bill-c8.html
21. Borden Ladner Gervais LLP, Bill C-8 Adopted: Critical Cyber Systems Protection Act, June 2026. https://www.blg.com/en/insights/2025/07/bill-c8-revives-canadian-cyber-security-reform-what-critical-infrastructure-sectors-need-to-know
22. European Union, Regulation (EU) 2022/2554 on digital operational resilience for the financial sector (DORA), Articles 5–15, EUR-Lex. https://eur-lex.europa.eu/eli/reg/2022/2554/oj
23. European Union, Regulation (EU) 2016/679 (General Data Protection Regulation), Article 32, EUR-Lex. https://eur-lex.europa.eu/eli/reg/2016/679/oj
24. European Union, Directive (EU) 2022/2555 (NIS2 Directive), Articles 21 and 23, EUR-Lex. https://eur-lex.europa.eu/eli/dir/2022/2555/oj
25. NIS-2-Directive.com, NIS 2 Directive, Article 21: Cybersecurity Risk-Management Measures (full article text). https://www.nis-2-directive.com/NIS_2_Directive_Article_21.html
26. Glocert International, NIS2 Article 21 Risk Management Measures Explained: All 10 Controls, December 2025. https://www.glocertinternational.com/resources/guides/nis2-article-21-risk-management-measures-explained/
27. European Union, Regulation (EU) 2024/2847 (Cyber Resilience Act), Article 14 and Annex I, EUR-Lex. https://eur-lex.europa.eu/eli/reg/2024/2847/oj
28. European Commission, The Cyber Resilience Act: Summary of the Legislative Text, Shaping Europe's Digital Future. https://digital-strategy.ec.europa.eu/en/policies/cra-summary
29. CyberResilienceAct.eu, The Cyber Resilience Act Explained: Scope, Classes and Deadlines, June 2026. https://www.cyberresilienceact.eu/explained.html
30. Keysight Technologies, One Year Countdown to EU CRA Compliance: September 11, 2026, Changes Everything, September 2025. https://www.keysight.com/blogs/en/tech/nwvs/2025/09/11/one-year-countdown-to-eu-cra-compliance-september-11-2026-changes-everything
31. Mend (via Security Boulevard), The EU Cyber Resilience Act: A Complete Compliance Guide for 2026 and Beyond, May 2026. https://securityboulevard.com/2026/05/the-eu-cyber-resilience-act-a-complete-compliance-guide-for-2026-and-beyond/
32. Information Commissioner's Office, Case Reference Number INV/0158/2020. https://ico.org.uk/media2/1kip5gw2/chartered-institiute-for-securities-and-investment-reprimand.pdf

33. Information Commissioner's Office, A Guide to Data Security (UK GDPR security outcomes). https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/security/a-guide-to-data-security/
34. UK Government, The Network and Information Systems Regulations 2018 (SI 2018/506). https://www.legislation.gov.uk/uksi/2018/506/contents
35. National Cyber Security Centre (UK), Cyber Assessment Framework (CAF). https://www.ncsc.gov.uk/collection/cyber-assessment-framework
36. National Cyber Security Centre (UK), Cyber Essentials: Requirements for IT Infrastructure (14-day update requirement). https://www.ncsc.gov.uk/cyberessentials/overview
37. Personal Information Protection Commission (Japan), Act on the Protection of Personal Information (APPI) and PPC Guidelines (English resources). https://www.ppc.go.jp/en/legal/
38. Japan - Cybersecurity Laws and Regulations 2026. https://iclg.com/practice-areas/cybersecurity-laws-and-regulations/japan
39. Japan's New Active Cyber Defense Law: Impact on Businesses. https://connectontech.bakermckenzie.com/japans-new-active-cyber-defense-law-impact-on-businesses/
40. Australian Government, Security of Critical Infrastructure Act 2018 (including Critical Infrastructure Risk Management Program rules). https://www.legislation.gov.au/C2018A00029/latest
41. Australian Cyber Security Centre, Essential Eight Maturity Model (patching applications and operating systems; removal of unsupported software). https://www.cyber.gov.au/resources-business-and-government/essential-cyber-security/essential-eight
42. Center for Internet Security, CIS Critical Security Controls v8.1, Control 2 (Inventory and Control of Software Assets) and Control 7 (Continuous Vulnerability Management). https://www.cisecurity.org/controls
43. AICPA, Trust Services Criteria (2017, with Revised Points of Focus, 2022), Criterion CC7.1. https://www.aicpa-cima.com/resources/download/2017-trust-services-criteria-with-revised-points-of-focus-2022
44. International Organization for Standardization, ISO/IEC 27001:2022 Information Security Management Systems, Annex A Controls 5.9, 8.8, 8.9. https://www.iso.org/standard/27001
45. CyberResilienceAct.eu, The Cyber Resilience Act Explained: Scope, Classes and Deadlines, June 2026. https://www.cyberresilienceact.eu/explained.html
46. endoflife.date, End-of-Life Dates for Frameworks, Runtimes, and Databases (community-maintained reference). https://endoflife.date/
47. Node.js Project (OpenJS Foundation), Previous Releases and End-of-Life Schedule. https://nodejs.org/en/about/previous-releases
48. Vue.js Team, Vue 2 End of Life (December 31, 2023). https://v2.vuejs.org/eol/