Computer Science and Engineering 2026, 16(2): 29-48
DOI: 10.5923/j.computer.20261602.01

Architectural Anti-Patterns in Critical Digital Infrastructure: A Catalogue of Recurring Structural Weaknesses and Their Mitigation Through Security Patterns

Cornelius Chipasha*, Simon Tembo, Mulundumina Shimaponda

Department of Electrical and Electronics, University of Zambia, Lusaka, Zambia, University of Zambia

Abstract Critical Digital Infrastructure (CDI), the power grids, water utilities, pipelines, and industrial control environments on which modern society depends, is undergoing rapid Information Technology / Operational Technology (IT/OT) convergence. This convergence has expanded the attack surface of systems that were never designed for external connectivity, and repeated incidents, including the 2000 Maroochy Water Services breach, the 2015-2016 Ukrainian grid attacks, the 2017 TRISIS/TRITON safety-system compromise, the 2020 SolarWinds supply-chain intrusion, and the 2021 Colonial Pipeline ransomware event, demonstrate that architecture, not merely control gaps, is a persistent source of systemic cyber risk, meaning risk capable of cascading beyond a single asset or organisation to affect an entire sector. Existing standards such as IEC 62443, NIST SP 800-82, and NIST CSF 2.0, adversary-behaviour catalogues such as MITRE ATT&CK for ICS, and enterprise architecture frameworks such as TOGAF, SABSA, Zero Trust, and the Purdue model, describe controls, risk-assessment procedures, adversary techniques, and design principles, but none catalogues the recurring architectural mistakes, or anti-patterns, that make those techniques and control gaps possible in the first place. This distinction matters: a technique catalogue explains how an intrusion unfolds once inside a system, and a control catalogue lists what safeguards exist, while neither explains why the same structural conditions keep recreating the same opportunity across unrelated organisations and sectors. This paper addresses that gap. Drawing on software anti-pattern theory, security pattern literature, architectural technical debt research, and forensic analysis of major CDI incidents, we identify, define, and classify twelve recurring architectural anti-patterns spanning governance, identity, trust, segmentation, visibility, operations, and resilience domains. Each anti-pattern is documented using a structured template covering context, problem, symptoms, root cause, security consequences, detection indicators, and a recommended pattern; each is corroborated by at least two independent sources spanning at least two CDI sectors, and mapped to one or more corresponding architectural security patterns. Contributions include a cross-sector catalogue of architecture-level anti-patterns for CDI, explicitly distinguished from existing adversary-technique catalogues and single-incident lessons-learned advisories; a reusable identification methodology; an anti-pattern dependency analysis; and a direct anti-pattern-to-pattern mapping with partial-remediation pathways. Practical implications, including implementation barriers and prioritisation guidance, are discussed for architects, operators, auditors, regulators, and researchers. The catalogue provides a practical, architecture-first foundation for proactively eliminating structural weaknesses in Critical Digital Infrastructure before they evolve into systemic cyber risk.

Keywords Critical digital infrastructure, Architectural anti-patterns, Security patterns, IT/OT convergence, Zero trust, Industrial control systems, Cyber-physical security

1. Introduction

1.1. Background

Critical Digital Infrastructure encompasses the energy, water, transportation, communications, and industrial systems that CISA formally recognises across sixteen critical infrastructure sectors, whose incapacitation would have a debilitating effect on national security, public health, or economic stability [24]. Over the past decade the architecture of these systems has grown markedly more complex. Distributed control systems that were once isolated by air gaps and proprietary protocols are now routinely bridged to enterprise networks, cloud analytics platforms, and remote-access services in pursuit of

* Corresponding author:
corneliuschipasha41@gmail.com (Cornelius Chipasha)
Received: Jul. 6, 2026; Accepted: Jul. 25, 2026; Published: Jul. 31, 2026
Published online at http://journal.sapub.org/computer

operational efficiency, predictive maintenance, and remote workforce enablement.

This shift is best understood as IT/OT convergence: the progressive integration of information technology networks, which prioritise confidentiality and rapid patching, with operational technology networks, which prioritise availability, safety, and long equipment lifecycles. Murray, Johnstone, and Valli describe this convergence as a structural transformation rather than a purely technical one, noting that it introduces IT-originated attack paths into environments whose protocols and devices were never designed to resist them [17]. Kapoor, Kumar, and Vardhan similarly observe that OT protocols such as Modbus and DNP3 lack native authentication or encryption, so any new connectivity path created by convergence becomes an exploitable seam rather than a monitored boundary [18].

Architectural decisions made early in a system's life, such as how networks are segmented, how trust is established between components, and where administrative authority is concentrated, tend to persist for decades in CDI because control systems are replaced far less frequently than enterprise IT assets. This durability means architectural choices function as long-term security determinants: a decision that appears economical or expedient at design time can silently accumulate risk for the operational life of the plant. Verdecchia, Kruchten, Lago, and Malavolta formalise this phenomenon as architectural technical debt (ATD), a class of design decisions that are locally rational when made but that degrade a system's structural integrity as the gap between intended and actual architecture widens over time [22]. In safety- and availability-critical environments, ATD does not merely raise maintenance cost, as it does in conventional software; it raises the probability and blast radius, meaning the set of systems reachable or credentials usable once a single point is compromised, of a cyber-physical failure.

The cost of poor architectural choices in CDI is not hypothetical. In the December 2015 Ukrainian grid intrusion, attackers who had gained a foothold in corporate IT networks moved laterally into SCADA dispatch systems over many months before disconnecting substations, an escalation the E-ISAC/SANS joint analysis attributes directly to insufficient segmentation between administrative and operational domains [13]. In 2017, the TRISIS/TRITON malware reached a Schneider Electric Triconex safety instrumented system at a Saudi petrochemical facility, an event CISA's malware analysis report treats as evidence that safety systems were reachable from a compromised engineering workstation with no independent trust boundary [15], [16]. In 2021, Colonial Pipeline shut down 5,500 miles of pipeline not because operational technology was directly compromised, but because the absence of a clean architectural boundary between IT and OT made it impossible to be confident that ransomware in the IT estate had not reached the pipeline control network [11]. Each incident traces back to an architectural condition that existed long before the attacker did.

A persistent assumption underlying much OT security practice is that critical systems remain protected by an air gap: a physical absence of any network connection to less trusted domains. This assumption rarely held even before IT/OT convergence became formal policy. The Maroochy Water Services incident is instructive precisely because it predates modern IT/OT convergence: in 2000, a disgruntled former contractor used a stolen radio transmitter and a laptop to issue unauthorised commands to Maroochy Shire's sewage pumping stations from outside any wired network, releasing roughly one million litres of untreated sewage into local waterways over several months before the cause was identified [26]. What Maroochy demonstrates, and what two subsequent decades of ICS incidents confirm, is that most so-called air gaps were never absolute; they were what practitioners now often call paper gaps, bridged by maintenance laptops, vendor modems, removable media, or, as in Maroochy's case, a radio link that fell outside the scope of any documented architecture at all. IT/OT convergence therefore does not introduce risk into a previously sealed environment so much as it formalises and multiplies a connectivity pathway that informal, undocumented bridges already provided, which is precisely why this paper treats architecture, including its undocumented and informal elements, rather than convergence alone, as the primary unit of analysis.

1.2. Problem Statement

Standards bodies and vendors have produced an extensive body of guidance for securing CDI. IEC 62443 defines foundational requirements and security levels for industrial automation and control systems [9]; NIST SP 800-82 Revision 3 provides a comprehensive guide to operational technology security [7]; the NIST Cybersecurity Framework 2.0 offers a taxonomy of cybersecurity outcomes applicable across sectors [8]; and CISA's Secure by Design initiative asks manufacturers to treat security as a design-time property rather than a bolt-on feature [20]. This is not to say that standards are silent on risk identification. IEC 62443-3-2 in particular defines a rigorous, seven-stage process for partitioning a system under consideration into zones and conduits and assessing risk for each one, and a careful, disciplined application of that process can surface many of the individual conditions this paper catalogues [27]. The distinction this paper draws is that IEC 62443-3-2 is a bottom-up procedure applied anew to each system under consideration, with no requirement to name, generalise, or compare the structural conditions it repeatedly uncovers across different organisations, whereas this paper's contribution is the top-down, cross-sector generalisation itself: a named, citable catalogue that a risk assessor can consult before beginning a zone-and-conduit exercise, rather than one they must independently rediscover each time. What existing standards, adversary-technique catalogues, and risk-assessment procedures have in common is that they describe controls, principles, adversary behaviour, and case-by-case risk procedures, that is, what a secure architecture should

contain and how a given system's risk should be assessed, without systematically describing which recurring architectural mistakes, made independently by unrelated organisations, reliably produce the conditions those safeguards exist to compensate for. This is the gap this paper addresses: existing literature rarely identifies recurring architectural anti-patterns that lead to systemic cyber risk as named, cross-sector, independently citable artefacts, even though incident forensics from Maroochy, Ukraine, TRISIS, SolarWinds, and Colonial Pipeline show the same handful of structural mistakes recurring across sectors, decades, and attacker types.

1.3. Research Objectives

1. Identify recurring architectural anti-patterns that generate systemic cyber risk across Critical Digital Infrastructure sectors.
2. Classify these anti-patterns according to the architectural domain in which they originate, and analyse how they enable or reinforce one another.
3. Explain the security consequences of each anti-pattern in terms of concrete, observable failure modes.
4. Provide mitigation guidance, including partial-remediation pathways and implementation barriers that is technology-independent and applicable across sector-specific architectures.
5. Map each anti-pattern to one or more corresponding architectural security patterns capable of eliminating it.

1.4. Research Contributions

These contributions are not uniform in kind. Some are substantive, contributing new, citable knowledge about the structure of CDI risk; others are methodological, contributing a reusable process and documentation discipline that this paper applies but that is not itself specific to the twelve anti-patterns catalogued here. Both kinds of contribution are listed together below, labelled accordingly, because a reader evaluating this paper's novelty should be able to distinguish claims about CDI architecture from claims about how to study it.

1. A cross-sector catalogue of twelve architecture-level anti-patterns specific to Critical Digital Infrastructure, explicitly distinguished in Section 2.4 from adversary -technique catalogues such as MITRE ATT&CK for ICS and from single-incident lessons-learned advisories such as individual CISA reports. (Substantive)
2. A repeatable, criteria-based methodology, including an operationalised recurrence threshold and a documented candidate-extraction process, for identifying and evidentially validating architectural anti-patterns from incident forensics and literature. (Methodological)
3. A structured, nine-field template that standardises how architectural anti-patterns are documented and compared. (Methodological)
4. A seven-layer classification taxonomy, spanning governance, identity, trust, segmentation, visibility, operations, and recovery, that organises the catalogue and identifies cross-cutting anti-patterns that span more than one layer. (Substantive)
5. An anti-pattern dependency analysis identifying which structural weaknesses enable or reinforce others, with direct implications for remediation sequencing. (Substantive)
6. A direct mapping from each anti-pattern to one or more corresponding architectural security patterns, including partial-remediation pathways for organisations that cannot implement the primary pattern in full. (Substantive)
7. An analysis of practical implications, including implementation barriers and prioritisation guidance, for five distinct CDI stakeholder groups: architects, operators, auditors, regulators, and researchers. (Substantive)
8. A research agenda identifying six concrete, sequenced directions for validating, automating, and extending this catalogue. (Methodological and forward-looking)

2. Related Work

2.1. Software Anti-Patterns

The concept of an anti-pattern originates in software engineering. Brown, Malveau, McCormick, and Mowbray defined an anti-pattern as a commonly occurring solution to a recurring problem that, despite appearing beneficial at the time it is adopted, produces consequences that outweigh its advantages, and they catalogued such patterns at the level of software design, software architecture, and project management [1]. Their central contribution was not any single anti-pattern but the discipline of documenting recurring failure modes with the same rigour traditionally reserved for successful design patterns, including a named problem, its root cause, and a refactored solution. This documentation discipline is the direct methodological ancestor of the anti-pattern template used in Section 4 of this paper. However, Brown et al.'s catalogue is scoped to general-purpose software systems; it does not address the cyber-physical, safety-critical, and availability-dominated concerns that distinguish Critical Digital Infrastructure from conventional IT.

2.2. Security Anti-Patterns

Security-specific extensions of the anti-pattern concept remain comparatively underdeveloped. Yoder and Barcalow's early work on architectural patterns for application security implicitly catalogued failure modes, such as trusting input validation to a single layer or granting a single session unchecked authority, by describing the positive pattern that corrects them [3]. Schumacher, Fernandez-Buglioni, Hybertson, Buschmann, and Sommerlad's Security Patterns

extended this line of work into a comprehensive catalogue spanning enterprise, architectural, and operational layers, integrating security engineering into the broader systems engineering process [2]. Both works are oriented toward enumerating good solutions rather than enumerating recurring mistakes; the anti-pattern, where it appears, is implicit in the problem statement that motivates a given pattern rather than treated as an independent, reusable artefact in its own right. Neither addresses the operational technology environments, legacy protocol constraints, or physical-consequence failure modes that characterise CDI. Adjacent secure-by-design literature approaches the same underlying problem from the software engineering side without closing this gap: SAFECode's Fundamental Practices for Secure Software Development catalogues secure-by-design development practices based on member organisations' real-world experience, providing a template for design-time security integration that this paper's Security-by-Afterthought anti-pattern (AP12) draws on directly, but its scope remains conventional software development rather than cyber-physical, safety-critical architecture [28].

2.3. Enterprise Architecture Weaknesses

Enterprise architecture and security architecture frameworks provide the structural vocabulary this paper builds on, but each has a scope that stops short of an anti-pattern catalogue. The TOGAF Standard, tenth edition, supplies an Architecture Development Method for producing and governing enterprise architectures but is domain-agnostic and does not address cyber-physical or safety concerns [4]. The Sherwood Applied Business Security Architecture (SABSA) offers a risk-driven, business-attribute-led method for deriving security architecture from business requirements across six layers of abstraction, from business context down to physical implementation, but, like TOGAF, it specifies a process for producing good architecture rather than a catalogue of the structural mistakes practitioners repeatedly make while following that process [5]. NIST SP 800-207 formalises Zero Trust Architecture as a set of tenets, chief among them that no implicit trust should be granted based on network location or asset ownership, and it does discuss several deployment-relevant failure modes directly, including subject and asset compromise, denial-of-service against policy decision points, and visibility gaps introduced by encrypted traffic; the tenet that no implicit trust be granted based on location is itself the corrective pattern for what this paper terms Implicit Trust Architecture in Section 5 [6]. What NIST SP 800-207 does not provide, and what this paper contributes, is a catalogue that generalises this and eleven other structural conditions across sectors and across the broader set of enterprise and security architecture frameworks discussed in this section, rather than one confined to zero-trust deployment specifically. The Purdue Enterprise Reference Architecture, originally developed by Williams for computer-integrated manufacturing, established the hierarchical, layered model of ICS network segmentation that IEC 62443 and most OT security guidance still reference, but it describes an intended reference topology rather than the ways real deployments deviate from it over time [10].

2.4. Critical Infrastructure Security

Sector-specific standards for Critical Digital Infrastructure concentrate almost entirely on control catalogues and process guidance. The ISA/IEC 62443 series defines foundational requirements, security levels, and a lifecycle for securing industrial automation and control systems, organised around policies, system requirements, and component requirements [9]. NIST SP 800-82 Revision 3 provides an extensive guide to operational technology security, including typical OT topologies, threats, vulnerabilities, and recommended safeguards [7]. The NIST Cybersecurity Framework 2.0 supplies a cross-sector taxonomy of cybersecurity outcomes, organised around Govern, Identify, Protect, Detect, Respond, and Recover functions [8]. CISA's Secure by Design whitepaper asks manufacturers to shift the burden of security from customers to producers by embedding security decisions at the design stage [20], while CISA's post-incident advisories, including its analyses of the DarkSide ransomware attack on Colonial Pipeline [11], the SolarWinds supply-chain compromise [12], the Ukrainian grid intrusion [14], and the HatMan/TRITON safety-system malware [15], provide detailed forensic accounts of how specific incidents unfolded. These advisories are an invaluable source of empirical evidence, and this paper draws on them extensively in Sections 3 and 5, but none of them, individually or collectively, synthesises the recurring architectural conditions across incidents into a named, reusable catalogue.

Two further bodies of work sit adjacent to sector standards without overlapping this paper's contribution. MITRE ATT&CK for ICS catalogues adversary tactics and techniques observed in real ICS intrusions, organised around post-compromise adversary behaviour such as lateral movement or manipulation of control [29]. It is a valuable and complementary resource: where ATT&CK for ICS documents how an intruder moves once inside a system, this paper's catalogue documents why the architecture allowed that movement to succeed at all; Flat Network Architecture (AP1) in Section 5, for instance, is precisely what makes ATT&CK for ICS's Lateral Movement tactic tractable in practice. Falco, Viswanathan, Caldera, and Shrobe's automated attack-planning methodology for cyber-physical systems generates candidate attack trees for a given system model, occupying similarly complementary ground: their approach discovers exploitable paths within one known architecture, while this paper names the recurring architectural conditions that make broad classes of such paths possible across many architectures at once [30].

NIST IR 8183 supplies a manufacturing-sector profile that maps the CSF outcomes in [8] to specific manufacturing risks and controls, illustrating how the general CSF

taxonomy is adapted to a single sector [31], while ENISA's recommendations for protecting industrial control systems provide a European-policy analogue to the CISA advisories cited throughout this paper, reinforcing that the architectural concerns catalogued here are not specific to a single regulatory jurisdiction [32]. Finally, Slay and Miller's forensic account of the 2000 Maroochy Water Services breach, discussed in Section 1.1, remains among the earliest documented cases showing that a control system's vulnerability was architectural rather than a missing patch, predating both formal IT/OT convergence and most of the standards discussed in this section [26].

2.5. Research Gap

Table 1 summarises this gap. Software anti-pattern literature offers a rigorous documentation discipline but is scoped to conventional software rather than architecture-level, cyber-physical concerns. Security pattern literature and enterprise architecture frameworks describe good solutions and sound processes but do not catalogue the recurring mistakes those solutions correct. Sector standards and post-incident advisories describe controls, risk-assessment procedures, and individual forensic timelines but stop short of an architecture-level anti-pattern catalogue that generalises across incidents and sectors. Adversary-technique catalogues explain how a compromise unfolds inside a given architecture, not why the same architectural conditions recur across unrelated ones. This paper's catalogue is positioned precisely in that gap: it treats architectural anti-patterns in CDI as first-class, independently documented artefacts, each traceable to forensic evidence and each mapped to an existing, standards-aligned corrective pattern.

Table 1. Literature Gap

Existing Work	Focus	Limitation
Software Anti-Patterns (e.g., Brown et al. [1])	General-purpose software design, architecture, and process	Not scoped to cyber-physical or architecture-level CDI concerns
Security Patterns (e.g., Schumacher et al. [2], Yoder & Barcalow [3])	Cataloguing good security solutions	Recurring mistakes are implicit, not independently catalogued
Enterprise / Security Architecture Frameworks (TOGAF [4], SABSA [5], Zero Trust [6], Purdue [10])	Process, tenets, and reference topology for producing sound architecture	Describe intended architecture and select deployment failure modes, not a cross-framework anti-pattern catalogue
Sector Standards, Risk Procedures, and Advisories (IEC 62443 [9], [27], NIST SP 800-82 [7], NIST CSF 2.0 [8], CISA advisories [11], [12], [14], [15])	Controls, per-system risk assessment, and single-incident forensics	No cross-incident, architecture-level anti-pattern catalogue
Adversary Technique / Attack-Path Catalogues (MITRE ATT&CK for ICS [29]; Falco et al. [30])	Post-compromise adversary tactics, techniques, and attack-tree paths	Explain how attacks execute within a given architecture, not why the same architectural conditions recur across architectures

3. Methodology

This paper identifies architectural anti-patterns through a triangulated, evidence-based process rather than through expert opinion alone. Five source categories were consulted. First, the software and security anti-pattern literature [1], [2], [3] supplied the documentation discipline and an initial set of candidate structural themes, such as concentrated authority and implicit trust that recur across engineering domains. Second, forensic reporting on major CDI incidents, including the Maroochy Water Services breach [26], the Ukrainian grid intrusion [13], [14], the TRISIS/TRITON safety-system compromise [15], [16], the SolarWinds supply-chain compromise [12], the Colonial Pipeline ransomware event [11], and the 2023 Okta identity-provider breach [21], [34], supplied empirical evidence that a given structural condition was causally implicated in a real cyber-physical or systemic outcome, not merely theoretically risky. Third, architecture-first research, including the architectural technical debt literature [22], provided a vocabulary for describing how sound decisions decay into structural weaknesses over an asset's operational life. Fourth, the security pattern literature [2], [3], [6] and OT-specific technical literature [18], [19] provided candidate corrective patterns against which each anti-pattern could be paired. Fifth, sector standards, including IEC 62443 and IEC 62443-3-2 [9], [27], NIST SP 800-82r3 [7], NIST CSF 2.0 [8], and the Purdue model [10], were used to verify that a proposed anti-pattern corresponded to a condition those standards already treat as a control or risk-assessment gap, ensuring the catalogue is standards-aligned rather than idiosyncratic.

Candidate extraction was performed as a directed content analysis of the five source categories: each source was read in full and coded for recurring structural themes, such as trust concentration, boundary absence, or telemetry absence, using the vocabulary of the source itself wherever possible rather than a predetermined coding scheme, in order to avoid forcing sources toward a conclusion decided in advance. This process yielded nineteen candidate structural themes. Each candidate was then cross-referenced against the other four source categories to determine whether

independent corroboration existed, and against the five selection criteria below. Twelve candidates satisfied all requirements; seven were set aside, either because they were evidenced in only one sector, because they described a configuration error rather than an architecture-level condition, or because no standards-aligned corrective pattern could be identified. Two of the seven set-aside candidates, a sector-specific SCADA-historian misconfiguration pattern and a building-automation-specific vendor lock-in condition, are noted here explicitly as candidates for future, sector-focused extensions of the catalogue rather than omitted silently. This extraction was conducted by the author as a single-analyst synthesis rather than a multi-coder systematic review, a limitation addressed directly under Validation Approach below.

Candidate anti-patterns drawn from these five sources were then filtered against five selection criteria, each of which a candidate had to satisfy before inclusion in the catalogue. It must recur across sectors, a criterion operationalised precisely rather than left to intuition: a candidate was retained only if it was independently evidenced by at least two primary sources, comprising incident forensics, standards text, or peer-reviewed literature, drawn from at least two of the sixteen CISA-recognised critical infrastructure sectors [24]. A condition evidenced by only one incident or one sector, however serious, was set aside as a case study rather than admitted as a catalogued anti-pattern. It must amplify cyber risk, meaning there is documented or standards-recognised causal evidence that the condition increases the likelihood, impact, or blast radius of a cyber-physical incident, rather than being a purely stylistic or aesthetic architectural preference. It must be architecture-level, meaning the condition is a property of how components, trust boundaries, and administrative domains are structured, rather than a configuration error, missing patch, or single misconfigured device. It must be technology-independent, meaning the anti-pattern can be described without reference to a specific vendor product or protocol, so that it remains valid as vendor technology changes. Finally, it must be remediable, meaning a corresponding architectural pattern exists, or can be reasonably specified, that eliminates or substantially mitigates the condition without requiring a complete system replacement.

Twelve candidate anti-patterns satisfied all five criteria after cross-referencing against the source material; these form the catalogue in Section 5. Their interdependencies, several of which surfaced during extraction but required the full catalogue to be assembled before they could be analysed systematically, are examined separately in Section 6.3. Figure 1 summarises this identification methodology as a four-stage pipeline: source identification, candidate extraction, criteria-based filtering, and formalisation using the structured template defined in Section 4.

3.1. Validation Approach

This paper's validation is evidential rather than experimental, and it is important to state plainly what that does and does not establish. Each of the twelve catalogued anti-patterns is validated internally through the multi-source corroboration required by the recurrence criterion above: every anti-pattern in Section 5 is traceable to at least two independent primary sources spanning at least two sectors, and its recommended pattern is cross-checked against at least one standards document or peer-reviewed empirical study, as shown in the citations accompanying each template field. This establishes that each anti-pattern is a real, recurring, evidenced structural condition rather than a speculative or singular one.

It does not establish the relative frequency, severity, or prevalence of each anti-pattern across the population of operating CDI environments, and it does not constitute prospective validation by independent practitioners who did not participate in constructing the catalogue. Producing that evidence would require either a structured expert-elicitation exercise, for example a Delphi panel of practising ICS security architects rating each anti-pattern's real-world prevalence and severity, or a case-study application of the full nine-field template against one or more live CDI architectures with practitioner sign-off. Neither was undertaken for this paper. Rather than overstate the catalogue's evidentiary standing, this paper identifies prospective, practitioner-led validation explicitly in Section 9 as the single most important direction for future work arising from it.

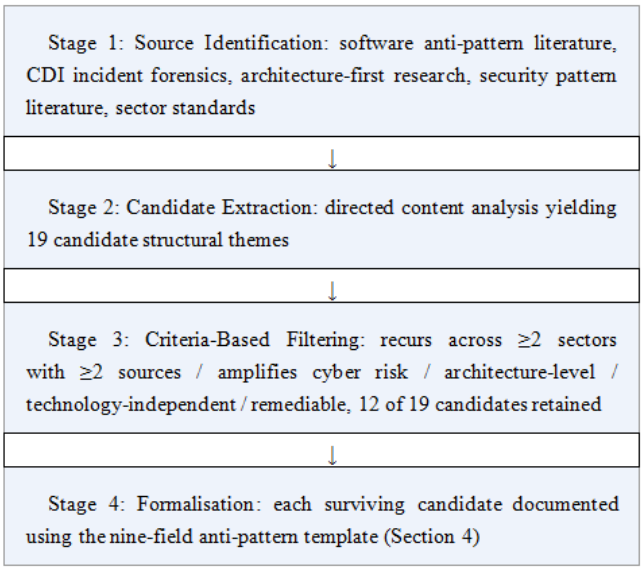

Figure 1. Anti-pattern identification methodology

4. Anti-Pattern Template

To ensure the catalogue in Section 5 is comparable, auditable, and directly actionable, every anti-pattern is documented using the same nine-field template, adapted from the documentation discipline established by Brown et al. [1] and extended with fields specific to cyber-physical CDI risk.

- Name: a short, memorable label used consistently throughout the paper and, ideally, in practitioner conversation.
- Context: the class of CDI environment in which the anti-pattern typically arises, such as brownfield OT networks or multi-site utility operations.
- Problem: a one-sentence statement of the structural condition itself.
- Symptoms: observable, practitioner-recognisable indicators that the anti-pattern is present, independent of any specific incident.
- Root Cause: the underlying architectural or organisational decision that produced the condition, distinguishing it from a superficial configuration issue.
- Security Consequences: the concrete failure modes the anti-pattern enables, expressed in terms of attacker capability, such as lateral movement or privilege escalation, rather than abstract risk scores. Where this paper describes an anti-pattern's blast radius, it means the set of systems reachable, or credentials usable, once a single point within that anti-pattern's scope is compromised.
- Detection Indicators: artefacts an architect, auditor, or automated tool could use to detect the anti-pattern's presence, such as flat VLAN topologies or absent authentication logs.
- Recommended Pattern: the named architectural security pattern, or patterns, that, if applied, eliminate or substantially mitigate the anti-pattern.
- Mitigation: a technology-independent description of how the recommended pattern is applied in practice, including, where relevant, concrete implementation classes and partial-remediation options.

5. Architectural Anti-Pattern Catalogue

This section presents the twelve anti-patterns that satisfied the selection criteria described in Section 3, each documented using the nine-field template defined in Section 4. Three pairs of anti-patterns in this catalogue address related but analytically distinct structural conditions, and the distinction is stated explicitly here to avoid ambiguity in later application. Flat Network Architecture (AP1) and Monolithic Security Perimeter (AP7) are both segmentation failures, but AP1 is a topological condition, the complete absence of any internal zone boundary, while AP7 is a control-placement condition, the presence of defined or de facto zones whose boundaries are enforced only at the outermost edge of the network. An environment can exhibit AP7 without AP1, for instance a network with clearly diagrammed internal zones that are never actually inspected or restricted at their internal boundaries, and the two anti-patterns therefore warrant independent detection and independent remediation even though both are corrected through complementary segmentation-family patterns. Implicit Trust Architecture (AP2) and Shared Administrative Domains (AP3) both concern trust, but along different dimensions: AP2 concerns the basis on which a single access decision is made, location versus verified identity, while AP3 concerns the scope of authority a single credential or administrative platform carries once trust has been granted. A system can correctly verify identity at every access decision, avoiding AP2, while still permitting one verified administrative identity to reach every criticality tier, exhibiting AP3. Shared Administrative Domains (AP3) and Dependency Concentration (AP11) both concern concentration of authority, but AP3 concerns concentration within the organisation's own administrative boundary, one internal team, directory, or credential set, while AP11 concerns concentration outside it, one external supplier, managed service, or shared platform on which the organisation depends. The corrective patterns differ accordingly: Distributed Policy Enforcement is an internal governance control, whereas Architectural Redundancy and Dependency Diversification is a sourcing and supply-chain control that internal policy redesign alone cannot substitute for.

AP1: Flat Network Architecture

Context: Brownfield OT networks that have grown organically over decades without a segmentation redesign.

Problem: Every device on the operational network can communicate with every other device, with no enforced boundary between zones of differing trust or criticality.

Symptoms: Single broad VLAN or physical LAN spanning engineering workstations, PLCs, HMIs, and historian servers; absence of internal firewalls or access control lists between control functions.

Root Cause: Segmentation was never designed in because the network predates routable IT connectivity, or was added incrementally without an architectural review.

Security Consequences: Once an attacker gains any foothold, lateral movement to safety-critical assets is unconstrained, directly enabling the credential harvesting and dispatch-station compromise observed in the Ukrainian grid intrusion [13] and the ransomware propagation patterns CISA warns against in its Colonial Pipeline advisory [11]. This is the specific architectural condition that makes MITRE ATT&CK for ICS's Lateral Movement tactic tractable in practice [29].

Detection Indicators: Network topology diagrams showing no internal boundaries; a single broadcast domain spanning multiple Purdue levels [10]; firewall rule sets with blanket allow rules between zones.

Recommended Pattern: Zero Trust Segmentation

Mitigation: Decompose the flat network into enforced zones aligned to the Purdue model [10] and NIST Zero Trust tenets [6], with explicit, logged, policy-based communication paths between zones rather than implicit connectivity. A durable remediation typically also requires the Identity-Centric Architecture pattern recommended for AP2, since enforcing zone boundaries without verifying what crosses them merely relocates implicit trust to the boundary rather than removing it; see Section 7 for this multi-pattern relationship.

AP2: Implicit Trust Architecture

Context: Any CDI environment where network location has historically been treated as a proxy for authorisation.

Problem: Systems and users are trusted because of where they sit on the network, such as being inside the OT perimeter, rather than because their identity and posture have been explicitly verified.

Symptoms: Authentication is weak or absent for devices already inside the OT boundary; a compromised perimeter device is automatically trusted by everything behind it.

Root Cause: Security architecture was designed around a castle-and-moat model that assumes the perimeter is impermeable, an assumption IT/OT convergence invalidates [17], [18]. Note this concerns the basis of a trust decision, location versus verified identity; it is distinct from Shared Administrative Domains (AP3), which concerns the scope a single verified credential is permitted to reach once trust is granted.

Security Consequences: A single breached perimeter device grants an attacker the same implicit trust as a legitimate internal system, collapsing the intended defence-in-depth into a single point of failure.

Detection Indicators: Absence of mutual authentication between internal components; access decisions logged as based on IP range or VLAN membership rather than verified identity.

Recommended Pattern: Identity-Centric Architecture

Mitigation: Apply NIST SP 800-207 Zero Trust Architecture tenets [6] so every access request is authenticated and authorised on its own merits, regardless of network location.

AP3: Shared Administrative Domains

Context: Organisations where a single IT or OT team, or a single privileged account, administers systems across multiple criticality tiers.

Problem: One administrator, credential set, or management platform holds control over systems that should be independently governed.

Symptoms: A single Active Directory forest, jump host, or remote-access tool spans corporate IT and safety-critical OT; one compromised administrative credential grants access everywhere.

Root Cause: Administrative consolidation is pursued for operational convenience and cost savings, often because too few staff understand both IT and OT domains to justify separate administration, without an accompanying blast-radius analysis. This internal concentration of authority tends to enable Single Trust Anchor (AP5) as a secondary effect, since the same operational convenience that motivates administrative consolidation typically motivates consolidating identity verification onto one platform at the same time; see Section 6.3.

Security Consequences: Compromise of a single privileged account, of the kind exploited in the 2023 Okta customer-support breach [21] and the SolarWinds supply-chain compromise [12], propagates across every system the shared domain touches.

Detection Indicators: Single sign-on or directory services spanning IT and OT; identical administrative credentials valid across safety and business systems.

Recommended Pattern: Distributed Policy Enforcement

Mitigation: Partition administrative domains so that privilege in one tier cannot be exercised in another without a separate, independently governed authorisation step. Where full partitioning is not immediately staffable, requiring distinct credentials, even under shared personnel, for OT-tier and IT-tier administration is a documented interim step that limits blast radius without requiring a larger team.

AP4: Legacy Protocol Exposure

Context: Industrial environments running Modbus, DNP3, or OPC Classic on networks reachable from IT or the internet.

Problem: Protocols designed without authentication, integrity checking, or encryption are exposed beyond the isolated segment they were designed for.

Symptoms: Modbus TCP or DNP3 traffic visible on IT-adjacent network segments; no protocol-aware inspection at the IT/OT boundary.

Root Cause: Legacy protocols were retained during IT/OT convergence for cost or compatibility reasons without adding a compensating security layer.

Security Consequences: East, Butts, Papa, and Shenoi catalogue at least twenty-three distinct attacks exploiting DNP3's absence of authentication and encryption alone [19], and Kapoor, Kumar, and Vardhan confirm the same absence in Modbus TCP, enabling command spoofing, man-in-the-middle modification, and replay attacks against process control [18].

Detection Indicators: Unencrypted, unauthenticated ICS protocol traffic crossing a routable boundary; absence of protocol allow-listing at the IT/OT demarcation.

Recommended Pattern: Secure Legacy Gateway

Mitigation: Interpose a protocol-aware gateway that terminates legacy protocols locally and re-establishes authenticated, encrypted communication for any traffic crossing a trust boundary, consistent with IEC 62443 zone and conduit requirements [9]. Concrete implementations of this pattern class include hardware-enforced unidirectional gateways and data diodes, which physically restrict traffic to a single direction and are well suited to telemetry replication, historian synchronisation, and patch-staging use cases where bidirectional legacy protocol traffic is not operationally required [33]; where bidirectional control traffic is unavoidable, a protocol-break application proxy that terminates and reconstructs each session, rather than merely forwarding packets, provides an analogous trust boundary without requiring field-device replacement. Where neither is immediately fundable, deploying the passive, protocol-aware monitoring recommended for AP6 as an interim compensating control converts an undetected

exposure into a detected and actively monitored one while the primary gateway deployment is planned.

AP5: Single Trust Anchor

Context: Organisations that consolidate identity, certificate, or time-synchronisation services into one provider or platform for operational simplicity.

Problem: A single identity provider, certificate authority, or authentication service is the sole root of trust for every downstream system.

Symptoms: One identity platform authenticates both corporate and industrial application access; no fallback authentication path exists if that platform is degraded or compromised.

Root Cause: Centralising identity management reduces operational overhead, but the resulting concentration is not offset by redundancy. A single trust anchor is itself frequently an externally hosted, single-supplier dependency, which is why this anti-pattern commonly co-occurs with, and reinforces, Dependency Concentration (AP11); see Section 6.3.

Security Consequences: The 2023 Okta breach demonstrated that compromise of a single identity provider can expose session tokens and support data affecting the entirety of its customer base. Okta's own root-cause disclosure attributes the initial access to a single compromised service -account credential, which is the specific failure mode a redundant trust-anchor design is intended to contain rather than merely detect after the fact [21], [34].

Detection Indicators: Absence of a secondary, independently operated authentication path; all downstream systems configured to trust exactly one issuing authority.

Recommended Pattern: Redundant Trust Anchors

Mitigation: Deploy independently operated, cross-validated trust anchors so that compromise or outage of one does not remove authentication assurance for the entire estate. Recovery infrastructure, specifically, should authenticate through a credential namespace distinct from production, which is also foundational to the Recovery Domain Isolation pattern recommended for AP9.

AP6: Monitoring Blind Spots

Context: OT networks where visibility instrumentation was never designed in, particularly below the SCADA/HMI layer.

Problem: Significant portions of the operational network, especially field-level PLC and RTU traffic, generate no telemetry that a defender can review.

Symptoms: No intrusion detection coverage below the control-centre layer; SPAN port limitations or absent sensors at PLC and RTU network segments; incident response reconstructing events entirely from IT-side logs.

Root Cause: OT protocols and legacy field devices were not designed to be observed, and retrofitting passive monitoring is frequently deprioritised against production continuity.

Security Consequences: In the Ukrainian grid intrusion, attackers retained undetected access for more than six months before executing the outage, a dwell time the E-ISAC/SANS analysis attributes in part to insufficient monitoring at the operational layer [13].

Detection Indicators: Absence of network taps or passive sensors below the HMI layer; security operations centres with no ingested OT telemetry; volumetric flow monitoring present without any protocol-level parsing.

Recommended Pattern: Layered Telemetry

Mitigation: Deploy passive, protocol-aware monitoring at each Purdue level [10], ensuring PLC and RTU segments generate telemetry without interfering with real-time control traffic [18]. Passive taps and SPAN-based collection are generally preferred at the field level because active probing risks introducing latency or malformed-packet responses into control loops with hard real-time constraints; where SPAN port packet loss under load is a concern, dedicated network TAP hardware avoids the switch-resource contention that otherwise causes passive visibility to silently degrade under exactly the traffic conditions an incident would produce. Protocol-depth inspection, parsing Modbus function codes or DNP3 object groups rather than treating traffic as opaque flows, is necessary because volumetric anomaly detection alone cannot distinguish a legitimate setpoint change from a malicious one; Layered Telemetry only closes the Monitoring Blind Spots gap when paired with this kind of protocol-aware analysis, not packet counting alone.

AP7: Monolithic Security Perimeter

Context: Organisations whose OT security investment is concentrated entirely on the IT/OT boundary.

Problem: Security controls are applied only at the outer perimeter, leaving the interior of the OT network without independent defence.

Symptoms: Firewalls and intrusion prevention exist only at the IT/OT demarcation; internal OT traffic is uninspected and unrestricted once past that boundary.

Root Cause: Perimeter-only defence reflects the same castle-and-moat assumption as Implicit Trust Architecture (AP2), applied specifically to security control placement rather than trust decisions. It is frequently entrenched by Static Security Policies (AP8): a policy set that is never revisited cannot extend enforcement inward as the network grows, which freezes protection at its original, perimeter-only scope; see Section 6.3.

Security Consequences: CISA's Colonial Pipeline advisory recommends robust IT/OT network segmentation precisely because a single perimeter breach otherwise grants unconstrained internal movement [11]; a monolithic perimeter converts one successful intrusion into total internal compromise.

Detection Indicators: Security control inventory showing investment concentrated at a single boundary; no internal choke points or inspection between OT sub-zones.

Recommended Pattern: Adaptive Microsegmentation

Mitigation: Extend enforcement into the interior of the

network using fine-grained, workload-aware microsegmentation, which recent zero-trust implementations show can isolate a compromised host from its microsegment in under a second in a cloud network context [23]; equivalent OT-specific validation, particularly within latency-sensitive control loops, remains an open evidence gap noted in Section 9.

AP8: Static Security Policies

Context: CDI environments where security policy is defined once during commissioning and rarely revisited.

Problem: Access control, segmentation, and monitoring policies do not adapt as the architecture, threat landscape, or operational context changes.

Symptoms: Firewall rule sets unchanged for years; access policies that do not distinguish between routine operations and elevated-risk states such as maintenance windows or active incidents.

Root Cause: Policy was treated as a one-time commissioning artefact rather than a continuously governed output of architecture, an omission the NIST Cybersecurity Framework's Govern function is designed to correct [8].

Security Consequences: Static policy cannot reflect newly connected assets, decommissioned systems, or emerging threat intelligence, so protection quietly decays relative to the current architecture even when no single control has failed.

Detection Indicators: Policy change logs showing no updates correlated with architecture changes; identical rule sets across audit cycles.

Recommended Pattern: Architecture-Aware Orchestration

Mitigation: Bind policy generation to a live architecture model so that segmentation and access rules are regenerated whenever the underlying architecture changes, aligned with CSF 2.0's Govern and Protect functions [8].

AP9: Recovery Monoculture

Context: Organisations that rely on a single backup platform, restoration procedure, or recovery site for all critical systems.

Problem: Every recovery path depends on the same underlying technology, credentials, or physical location as the production systems it is meant to restore.

Symptoms: Backup infrastructure reachable from the same administrative domain as production; a single disaster-recovery runbook covering both IT and OT with no independent validation.

Root Cause: Recovery architecture is commonly treated as an IT afterthought rather than a design-time requirement, mirroring Security-by-Afterthought (AP12) applied specifically to resilience.

Security Consequences: Colonial Pipeline's operators shut down the entire pipeline, rather than isolating and restoring only the affected segment, because no independently validated recovery path existed that they could trust was unaffected by the ransomware in the IT estate [11].

Detection Indicators: Backup and recovery systems sharing credentials, network paths, or administrative control with production; absence of tested, air-gapped recovery copies.

Recommended Pattern: Recovery Domain Isolation

Mitigation: Maintain independently governed, credential-isolated recovery domains for each criticality tier, tested through exercises that do not assume the primary domain is trustworthy. In practice this means recovery infrastructure authenticates through a credential namespace distinct from production, per the Redundant Trust Anchors pattern recommended for AP5; is connected to production only through a one-way or scheduled, explicitly authorised replication path rather than continuous bidirectional access; and is validated through periodic restoration drills that measure whether a segment can be recovered in isolation, without invoking any credential, service, or network path that could plausibly be compromised alongside the production system it is restoring. This is a materially different architectural commitment from simply operating a backup product: a backup that shares an administrative domain with production, as Recovery Monoculture describes, provides no assurance against the exact scenario it exists to address, namely that the primary domain itself is the compromised one.

AP10: Architecture Drift

Context: Long-lived CDI environments subject to continuous incremental change, such as vendor upgrades, new sensors, or acquired facilities.

Problem: The as-built architecture diverges from its documented or intended design over time, with no governance process to detect or correct the divergence.

Symptoms: Network diagrams that no longer match observed topology; undocumented remote-access points added for vendor support; segmentation exceptions granted temporarily and never revoked.

Root Cause: Verdecchia et al. characterise this as architectural technical debt: individually rational local decisions accumulate into a structural gap between intended and actual architecture that no single decision-maker is accountable for closing [22].

Security Consequences: Each undocumented deviation is a candidate anti-pattern in its own right; drift is the mechanism by which Flat Network Architecture (AP1), Monitoring Blind Spots (AP6), and Static Security Policies (AP8) silently re-emerge even in environments that were once correctly architected. This enabling relationship is analysed formally in Section 6.3.

Detection Indicators: Discrepancies between architecture documentation and automated network discovery; audit findings that recur across consecutive assessment cycles despite prior remediation.

Recommended Pattern: Continuous Architectural Assurance

Mitigation: Establish a continuous, automated comparison between intended and as-built architecture, with governance ownership assigned to close discrepancies rather than

merely document them [22].

AP11: Dependency Concentration

Context: Organisations that rely on a single supplier, shared service, or platform for a function common to many critical systems.

Problem: Critical operational or security functions depend on one external supplier or shared component with no substitute if that dependency is compromised or unavailable.

Symptoms: A single software vendor's update mechanism touches every managed endpoint; one managed service provider holds administrative access across multiple customer environments.

Root Cause: Vendor consolidation and shared-service adoption reduce cost and integration complexity but concentrate systemic risk in proportion to that consolidation. This is distinct from Shared Administrative Domains (AP3): AP3 concentrates authority inside the organisation's own administrative boundary, while Dependency Concentration concentrates it in an external supplier or shared service the organisation does not directly control, even though the two frequently co-occur, for instance when a single externally hosted identity platform is also used for internal administrative sign-on.

Security Consequences: The SolarWinds Orion compromise allowed a single trojanised software update to reach government agencies and critical infrastructure entities simultaneously, illustrating how dependency concentration converts a single supply-chain compromise into a multi-sector incident [12].

Detection Indicators: Software bill-of-materials analysis showing common upstream dependencies across otherwise independent systems; contractual review showing no substitute supplier for a critical function.

Recommended Pattern: Architectural Redundancy and Dependency Diversification

Mitigation: Diversify suppliers and shared services for critical functions where feasible, and where consolidation is unavoidable, isolate the blast radius of that dependency through segmentation and independent verification of updates before deployment.

AP12: Security-by-Afterthought

Context: Systems, including safety instrumented systems, where security controls are proposed only after deployment or after an incident.

Problem: Security is added to an existing architecture rather than designed into it from the outset, so protective controls must work around structural decisions already made.

Symptoms: Security requirements appear as a late change request rather than an initial design input; safety and control systems commissioned with no independent security review.

Root Cause: Security ownership is organisationally separated from architecture and engineering decision-making, so security is consulted after, rather than during, design. Because security is absent when structural decisions are actually made, the default outcomes of unconstrained design, namely flat topology (AP1), location-based trust (AP2), and perimeter-only defence (AP7), are what remain unless a later, harder retrofit corrects them; see Section 6.3.

Security Consequences: The TRISIS/TRITON malware reached a Triconex safety instrumented system because that system's engineering workstation had no independent security boundary from the broader control network, a condition CISA's analysis attributes to the safety system being designed for functional isolation rather than adversarial resistance [15], [16].

Detection Indicators: Security requirements documentation dated after system commissioning; architecture review records showing no security stakeholder present at design decision points.

Recommended Pattern: Architecture-First Secure Design

Mitigation: Require security architecture review as a mandatory, non-bypassable gate at each design decision point, consistent with CISA's Secure by Design principle that manufacturers, not customers, bear primary responsibility for security outcomes [20].

To provide a systematic understanding of architectural weaknesses, the identified anti-patterns are organised into a hierarchical set of architectural security domains. The hierarchy reflects the logical progression of architecture-first cybersecurity, beginning with governance as the strategic foundation and extending through identity, trust, segmentation, visibility, operational resilience, and recovery. This layered organisation illustrates how weaknesses in higher-level architectural domains can propagate to lower layers, thereby amplifying systemic cyber risk.

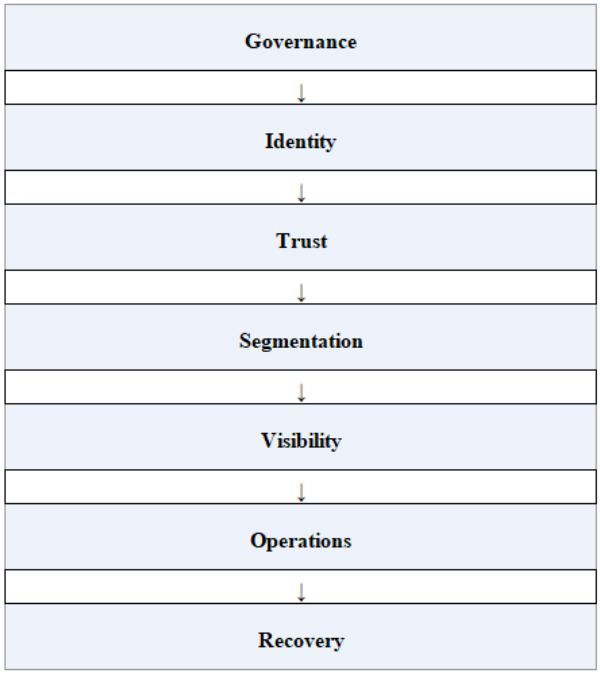

Figure 2. Anti-pattern classification stack

Table 2. Anti-Pattern Summary

Anti-Pattern	Symptoms	Consequences	Recommended Pattern
AP1 Flat Network Architecture	No internal network boundaries; single broadcast domain	Unconstrained lateral movement, ransomware propagation	Zero Trust Segmentation
AP2 Implicit Trust Architecture	Weak internal authentication; trust based on location	Perimeter breach grants full internal trust	Identity-Centric Architecture
AP3 Shared Administrative Domains	Single directory/credential spans IT and OT	One compromised credential grants cross-domain access	Distributed Policy Enforcement
AP4 Legacy Protocol Exposure	Unauthenticated Modbus/DNP3 on routable segments	Command spoofing, replay, man-in-the-middle	Secure Legacy Gateway
AP5 Single Trust Anchor	One identity/certificate provider for all systems	Provider compromise cascades to entire estate	Redundant Trust Anchors
AP6 Monitoring Blind Spots	No telemetry below HMI/SCADA layer	Long attacker dwell time, delayed detection	Layered Telemetry
AP7 Monolithic Security Perimeter	Controls only at IT/OT boundary	Single breach yields total internal compromise	Adaptive Microsegmentation
AP8 Static Security Policies	Unchanged rule sets over multiple audit cycles	Protection decays as architecture evolves	Architecture-Aware Orchestration
AP9 Recovery Monoculture	Backup shares credentials/ paths with production	Forced full shutdown instead of isolated recovery	Recovery Domain Isolation
AP10 Architecture Drift	Documentation diverges from as-built topology	Re-emergence of other anti-patterns undetected	Continuous Architectural Assurance
AP11 Dependency Concentration	Single supplier/service touches many systems	Supply-chain compromise cascades across sectors	Architectural Redundancy and Dependency Diversification
AP12 Security-by-Afterthought	Security review absent at design stage	Safety/control systems reachable without adversarial resistance	Architecture-First Secure Design

6. Classification

6.1. Domain Classification

The twelve anti-patterns cluster into seven architectural domains, arranged in Figure 2 from the most abstract, governance, to the most operational, recovery. This ordering is intentional: governance failures create the conditions in which identity, trust, and segmentation failures are tolerated, while visibility, operations, and recovery failures determine how much damage those upstream failures ultimately cause.

Governance anti-patterns concern how architectural decisions are made and audited over time rather than any single technical control. Architecture Drift (AP10) and Security-by-Afterthought (AP12) both originate in organisational process: the former in the absence of continuous architectural governance, the latter in the absence of security participation at design time. Identity anti-patterns concern how individual actors and systems are authenticated. Implicit Trust Architecture (AP2) and Single Trust Anchor (AP5) both weaken the assurance that a claimed identity is genuine, either by substituting network location for verification or by concentrating verification in one unredundant service. Trust anti-patterns concern how authority is delegated once identity is established. Shared Administrative Domains (AP3) and Dependency Concentration (AP11) both extend trust further than a single compromise should permit, whether internally through shared credentials or externally through shared suppliers.

Segmentation anti-patterns concern how the network itself is structured, and accordingly contain the largest cluster: Flat Network Architecture (AP1), Legacy Protocol Exposure (AP4), and Monolithic Security Perimeter (AP7) each describe a different way in which network boundaries fail to constrain an attacker's movement, whether by omission, by protocol weakness, or by concentrating defence at a single point. Visibility is represented by Monitoring Blind Spots (AP6), the condition in which defenders cannot observe what segmentation and identity controls are meant to protect. Operations is represented by Static Security Policies (AP8), the condition in which controls exist but fail to adapt to a changing architecture. Recovery is represented by Recovery Monoculture (AP9), the condition in which the response to a successful attack is itself architecturally fragile.

6.2. Cross-Cutting Anti-Patterns and Scope

The seven-layer classification in Table 3 assigns each anti-pattern to a single primary domain for clarity, but several anti-patterns exert cross-cutting effects that a single-domain assignment understates. Flat Network Architecture (AP1), classified under Segmentation, also has direct Operations

consequences, because a flat topology makes Static Security Policies (AP8) harder to scope meaningfully, and direct Recovery consequences, because it removes the isolation a recovery domain would otherwise rely on. Legacy Protocol Exposure (AP4) is simultaneously a Segmentation and a Visibility concern, since unauthenticated legacy protocols are both a boundary weakness and, as Section 5 discusses under AP6, a source of monitoring difficulty in their own right. This cross-cutting behaviour is not a flaw in the classification; it is evidence that governance-domain anti-patterns and segmentation-domain anti-patterns interact causally rather than merely co-occurring statistically, a relationship examined directly in Section 6.3.

Table 3. Anti-Pattern Classification

Domain	Anti-Patterns
Governance	AP10 Architecture Drift; AP12 Security-by-Afterthought
Identity	AP2 Implicit Trust Architecture; AP5 Single Trust Anchor
Trust	AP3 Shared Administrative Domains; AP11 Dependency Concentration
Segmentation	AP1 Flat Network Architecture; AP4 Legacy Protocol Exposure; AP7 Monolithic Security Perimeter
Visibility	AP6 Monitoring Blind Spots
Operations	AP8 Static Security Policies
Recovery	AP9 Recovery Monoculture

This paper's seven-layer classification is also deliberately scoped to cyber-architectural domains and does not include physical security, such as facility access control to control rooms, or personnel security, such as insider-threat vetting for engineering staff, both of which are increasingly treated as architectural concerns in integrated CDI security programmes, and both of which are directly relevant to incidents such as Maroochy, which exploited a former insider's retained access rather than a purely network-architectural weakness [26]. Extending the classification to formally incorporate physical and personnel domains is identified in Section 9 as a direction for future work rather than attempted here, where doing so would require a comparable evidentiary base of physical- and personnel-security incidents that this paper's five source categories were not designed to collect systematically.

6.3. Anti-Pattern Dependency Analysis

The domain classification in Table 3 organises anti-patterns by where they originate, but it does not by itself show how they interact once present. These enabling relationships were identified inductively rather than through a separate empirical study: during the formalisation stage described in Section 3, cross-references embedded in each anti-pattern's own root-cause field, for instance AP10's stated role in AP1, AP6, and AP8, were collected once the full twelve-entry catalogue existed and checked for consistency on both sides of each relationship, since a dependency spanning two catalogue entries could only be confirmed after both entries had been drafted. Re-examining the root-cause fields across all twelve catalogue entries in Section 5 in this way reveals that several anti-patterns are not independent: the presence of one measurably increases the likelihood that others persist undetected or unremediated. Table 4 summarises five such enabling relationships, each supported by the root-cause evidence already presented in Section 5 rather than by new data. The strongest of the five also has the clearest independent support in the wider literature: Verdecchia et al.'s characterisation of architectural technical debt as an accumulating gap between intended and as-built architecture is a general mechanism for exactly this kind of downstream re-emergence, not a claim specific to CDI, which is why Architecture Drift's enabling role in AP1, AP6, and AP8 is treated here as the best-supported relationship in Table 4 rather than merely the first-listed one [22].

Architecture Drift (AP10) is the most consequential enabling condition in the catalogue: because it is defined as the undetected divergence between intended and as-built architecture, its presence is definitionally what allows Flat Network Architecture (AP1), Monitoring Blind Spots (AP6), and Static Security Policies (AP8) to re-emerge even in environments that were correctly architected at commissioning. Security-by-Afterthought (AP12) plays an analogous enabling role at design time rather than over the system's operational life: because security is not present at the point structural decisions are made, the default outcomes of unconstrained design, namely flat topology (AP1), location-based trust (AP2), and perimeter-only defence (AP7), are what remain unless a later, harder retrofit corrects them. Shared Administrative Domains (AP3) tends to enable Single Trust Anchor (AP5), because the operational convenience that motivates administrative consolidation in the first place typically motivates consolidating identity verification onto one platform at the same time, and Single Trust Anchor (AP5) in turn frequently enables Dependency Concentration (AP11), because the consolidated identity platform is itself commonly an externally hosted, single-supplier service. Static Security Policies (AP8) tends to entrench Monolithic Security Perimeter (AP7), because a policy set that is never revisited cannot extend enforcement inward as the network grows, freezing protection at its original, perimeter-only scope.

These dependencies are not always pairwise; some compound. Shared Administrative Domains (AP3) and Dependency Concentration (AP11) are conceptually distinct, as Section 5 establishes, but they are also mutually reinforcing in a specific, commonly observed configuration: an externally hosted identity platform, itself an instance of Dependency Concentration, that is also configured with a single shared administrative credential set valid across every environment it serves combines both concentration conditions into one compound vulnerability, in which compromising a single credential at a single external provider yields the combined blast radius of both anti-patterns rather than either alone. The 2023 Okta breach discussed under AP5 illustrates this compounding directly: Okta functioned

as a Dependency Concentration point for its customers collectively, while each affected customer's own internal administrative practices, specifically how broadly a compromised support-account credential was trusted internally, determined how far the compromise reached within that organisation [21], [34].

This dependency structure has a direct, practical implication for remediation sequencing, addressed further in Section 8: because Architecture Drift and Security-by-Afterthought are upstream of several other anti-patterns rather than parallel to them, governance-domain remediation undertaken first tends to make downstream remediation more durable, whereas remediating a downstream anti-pattern, for instance re-segmenting a flat network, without addressing the governance condition that produced it risks the same anti-pattern re-emerging through the same undetected drift that created it originally. This makes the seven-layer hierarchy in Figure 2 partially prescriptive as well as descriptive: it describes how weaknesses typically manifest, and, through the dependency relationships identified here, it also indicates a defensible remediation order, without claiming that governance remediation must always precede every other action in every organisation regardless of immediate risk.

7. Mapping Anti-Patterns to Security Patterns

One of the catalogue's central contributions is that every anti-pattern is paired with at least one named, standards-aligned architectural security pattern capable of eliminating it. This mapping, summarised in Table 5, identifies the pattern that most directly negates each anti-pattern's root cause, not merely a pattern that generally improves security posture. Zero Trust Segmentation, for instance, is paired with Flat Network Architecture (AP1) because NIST SP 800-207's core tenet, that no implicit trust is granted based on network location, is the precise structural inverse of a network with no enforced boundaries [6]. Identity-Centric Architecture is paired with Implicit Trust Architecture (AP2) for the same reason, applied to authentication rather than segmentation.

Several pairings draw on empirical evidence of the pattern's effectiveness rather than standards language alone. Adaptive Microsegmentation is paired with Monolithic Security Perimeter (AP7) partly because recent zero-trust microsegmentation implementations demonstrate measurable containment: Arora and Hastings report that a microsegmented architecture can isolate a compromised host from its segment in under 250 milliseconds during simulated ransomware propagation in a cloud network context, directly addressing the internal blast-radius problem that a perimeter-only defence cannot [23]. Secure Legacy Gateway is paired with Legacy Protocol Exposure (AP4) because it is the only pattern that resolves the underlying protocol weakness without requiring wholesale replacement of field devices that may remain in service for decades, an economic constraint IEC 62443's zone-and-conduit model explicitly accommodates [9].

This mapping is what connects the anti-pattern catalogue to the broader architectural security pattern literature. Where Schumacher et al. [2] and Yoder and Barcalow [3] document security patterns as solutions in search of a problem statement, this paper's contribution is to supply the missing problem statement, each catalogued anti-pattern, in a form specific enough to be independently detected, audited, and prioritised. An organisation using Table 5 does not need to select security patterns speculatively; it can instead assess which anti-patterns are present using the detection indicators in Section 5, and adopt only the patterns that correct a confirmed structural weakness. Figure 3 depicts this relationship as a direct correspondence between the anti-pattern catalogue and the security pattern catalogue, positioning the two as complementary rather than overlapping bodies of work.

Table 4. Anti-Pattern Dependency Relationships

Enabling Anti-Pattern	Enabled / Reinforced Anti-Patterns	Mechanism
AP10 Architecture Drift	AP1 Flat Network Architecture; AP6 Monitoring Blind Spots; AP8 Static Security Policies	Undetected divergence between intended and as-built architecture lets segmentation, telemetry, and policy scope silently erode
AP12 Security-by-Afterthought	AP1 Flat Network Architecture; AP2 Implicit Trust Architecture; AP7 Monolithic Security Perimeter	Absence of security participation at design time leaves flat topology, location-based trust, and perimeter-only defence as the unexamined default
AP3 Shared Administrative Domains	AP5 Single Trust Anchor	Administrative consolidation typically consolidates identity verification onto the same platform
AP5 Single Trust Anchor	AP11 Dependency Concentration	The consolidated identity or certificate platform is itself commonly an externally hosted, single-supplier dependency
AP8 Static Security Policies	AP7 Monolithic Security Perimeter	Policy that is never revisited cannot extend enforcement inward as the network grows, freezing protection at its original perimeter-only scope

Table 5. Anti-Pattern to Security Pattern Mapping

Anti-Pattern	Security Pattern
AP1 Flat Network Architecture	Zero Trust Segmentation (with Identity-Centric Architecture)
AP2 Implicit Trust Architecture	Identity-Centric Architecture
AP3 Shared Administrative Domains	Distributed Policy Enforcement
AP4 Legacy Protocol Exposure	Secure Legacy Gateway
AP5 Single Trust Anchor	Redundant Trust Anchors
AP6 Monitoring Blind Spots	Layered Telemetry
AP7 Monolithic Security Perimeter	Adaptive Microsegmentation
AP8 Static Security Policies	Architecture-Aware Orchestration
AP9 Recovery Monoculture	Recovery Domain Isolation (with Redundant Trust Anchors)
AP10 Architecture Drift	Continuous Architectural Assurance
AP11 Dependency Concentration	Architectural Redundancy and Dependency Diversification
AP12 Security-by-Afterthought	Architecture-First Secure Design

Figure 3. Relationship between anti-patterns and security patterns

The one-to-one correspondence in Table 5 names the single pattern that most directly negates each anti-pattern's root cause, and this precision is deliberate: it gives an organisation an unambiguous starting point. In practice, however, complex anti-patterns are rarely closed by one pattern in isolation. Flat Network Architecture (AP1), for example, is formally corrected by Zero Trust Segmentation, but a segmentation redesign that enforces zone boundaries

without also verifying the identity of what crosses them merely relocates Implicit Trust Architecture (AP2) to the zone boundary rather than eliminating it; a durable remediation of AP1 therefore typically requires Zero Trust Segmentation and Identity-Centric Architecture to be implemented together, even though only the former is listed as AP1's primary corrective pattern in Table 5. Table 5 should accordingly be read as identifying the necessary pattern for each anti-pattern, not always the sufficient one.

Organisations facing budget, staffing, or legacy-technology constraints frequently cannot implement a primary pattern in full immediately, and the catalogue's practical value would be limited if it offered no guidance for that circumstance. For Legacy Protocol Exposure (AP4), where a full Secure Legacy Gateway deployment may require a capital cycle to fund, an interim compensating step is to deploy the passive, protocol-aware monitoring described under Layered Telemetry for AP6, which does not eliminate the exposure but converts an undetected weakness into a detected and actively monitored one while the primary remediation is planned. For Recovery Monoculture (AP9), where fully isolated recovery infrastructure may not be immediately fundable, a documented interim step is to separate, at minimum, the credential namespace used for backup and recovery from the production identity provider identified in Single Trust Anchor (AP5), even before physical or network isolation is achieved, since credential separation alone removes the specific failure mode that forced Colonial Pipeline's full shutdown [11]. These partial pathways are compensating controls, not substitutes for the primary pattern, and should be tracked as open remediation items rather than treated as closure.

Recommended patterns can also conflict with operational requirements, most notably in the tension between Zero Trust Segmentation's per-request authentication overhead and the deterministic, low-latency communication that safety instrumented systems and real-time control loops require. IEC 62443's security-level tiering offers one reconciliation mechanism: applying full zero-trust enforcement, including authentication and inspection latency, at lower security-level zones while relying on pre-authenticated, cached, or session-persistent trust relationships within the highest security-level zones closest to physical process control, so that the pattern's overhead is concentrated where risk tolerance is highest and avoided where determinism is safety-critical [27]. Evidence of this trade-off's practical resolution remains limited, and this scarcity of evidence is itself worth stating plainly rather than glossing over. To the author's knowledge, no published study measures microsegmentation containment speed within an OT or safety-instrumented-system control loop specifically; the effectiveness figures reported by Arora and Hastings remain the closest available evidence despite being drawn from a cloud network context, and this absence of OT-specific evidence is precisely the empirical gap Section 9 identifies as a direction for future pattern-effectiveness research [23].

8. Practical Implications

(1) Architects

For architects, the catalogue offers a checklist that can be applied during initial design and during subsequent architecture reviews, using the detection indicators in Section 5 to identify which anti-patterns are already present in a system under design or modification. Because each anti-pattern is paired with at least one recommended pattern in Table 5, architects can prioritise remediation investment against confirmed structural weaknesses rather than generic best-practice guidance, while treating the multi-pattern and partial-remediation guidance in Section 7 as the practical starting point rather than the one-to-one mapping alone.

(2) Operators

For operators, the catalogue reframes day-to-day operational decisions, such as approving a new vendor remote-access path, consolidating administrative credentials for convenience, or accepting a temporary segmentation exception during a maintenance window, as architectural choices with anti-pattern implications rather than purely operational conveniences. Recognising that a proposed change would introduce Shared Administrative Domains (AP3) or Dependency Concentration (AP11) allows operators to raise the same objection an architect would, without requiring a separate architecture review for every operational decision. Because operators are typically the first to encounter proposed changes and the last to be consulted on their architectural consequences, giving them the vocabulary and detection indicators in Section 5 directly is, in practice, a faster route to catching an anti-pattern before it is committed to production than routing every change through a centralised architecture review board.

(3) Auditors

For auditors, the nine-field template in Section 4 supplies a standardised basis for assessment findings that is more specific than generic control-catalogue compliance. An audit finding framed as "Flat Network Architecture (AP1) present, evidenced by absence of internal segmentation between engineering workstations and safety controllers" is directly actionable in a way that a finding framed only against a control catalogue's paragraph number is not, because it names both the structural condition and its corrective pattern.

(4) Regulators

For regulators, the catalogue provides a vocabulary for distinguishing between organisations that have merely implemented required controls and organisations whose underlying architecture no longer generates the conditions those controls exist to compensate for. This distinction matters increasingly as regulatory regimes, following CISA's Secure by Design initiative [20], shift expectations from control compliance toward demonstrated architectural resilience.

(5) Researchers

For researchers, the catalogue is deliberately incomplete

rather than final. Each anti-pattern is a falsifiable claim, supported by cited evidence, that can be challenged, refined, or extended as new incidents and sectors are studied, and the methodology in Section 3, including the validation limitations acknowledged in Section 3.1, is reusable for that purpose independent of this paper's specific twelve-item catalogue.

(6) Implementation Barriers

None of the recommended patterns in Table 5 are cost-free, and the catalogue's practical value depends on being honest about that. Economically, segmentation hardware, secure legacy gateways, and redundant trust infrastructure are capital expenditures, for example a hardware data diode deployment sized for a single AP4 conduit can run into the tens of thousands of dollars once installation and commissioning are included, that compete with production-focused investment in environments where security spending has historically been justified reactively rather than proactively; the CISA advisories cited throughout this paper [11], [12], [14], [15] are themselves evidence that reactive, post-incident funding remains the norm this paper's proactive framing is trying to change. Organisationally, the IT/OT cultural divide that motivates Shared Administrative Domains (AP3) in the first place, for instance a single engineer being the only person on staff able to administer both the corporate directory and the SCADA historian, is a scarcity of dual-domain staff that also tends to be the primary obstacle to correcting it, since Distributed Policy Enforcement requires governance capacity that a merged, understaffed team may not have. Technically, some recommended patterns cannot be applied to the oldest field devices at all: a programmable logic controller with no spare compute or memory headroom cannot terminate a modern authenticated session, which is precisely why Secure Legacy Gateway, AP4's recommended pattern, is designed to enforce a boundary around such a device rather than modify it, and why the partial-remediation pathways described in Section 7 are a first-class part of this paper's guidance rather than an afterthought.

(7) Prioritisation Guidance

For organisations that cannot remediate all twelve anti-patterns simultaneously, and in practice none can, the dependency analysis in Section 6.3 provides a defensible starting point rather than an arbitrary one. Because Architecture Drift (AP10) and Security-by-Afterthought (AP12) are upstream of several other catalogued anti-patterns, establishing continuous architectural governance and a design-time security review gate first tends to make every subsequent remediation more durable, even though it is not the most visible or immediately risk-reducing action available. Where immediate risk reduction is the binding constraint rather than long-term durability, for instance following a specific threat intelligence indicator or a recent sector-wide incident, prioritising the segmentation-domain anti-patterns, Flat Network Architecture (AP1), Legacy Protocol Exposure (AP4), and Monolithic Security Perimeter (AP7), directly addresses the lateral-movement and blast-radius mechanisms responsible for the largest documented incidents in this paper's evidence base [11], [13]. Neither sequencing is universally correct; the choice between governance-first durability and segmentation-first risk reduction is itself a risk-tolerance decision that this paper deliberately leaves to the organisation rather than prescribes, consistent with the descriptive, rather than strictly prescriptive, character of the classification established in Section 6.

9. Future Research

Six directions extend this work, ordered here from nearest-term to longest-horizon; the ordering is not strictly independent, and it is worth stating the dependency explicitly rather than leaving it implicit: automated detection (the second direction) supplies the prevalence data that prospective validation (the first direction) needs to be conclusive, and AI-assisted architecture review (the fifth direction) in turn depends on the labelled prevalence data both earlier directions would produce, so the sequence below should be read as a dependency chain rather than a strict priority ranking. First, and most immediately, prospective validation: the multi-source corroboration described in Section 3.1 establishes that each anti-pattern is evidenced, but a near-term structured expert-elicitation exercise, such as a Delphi panel of practising ICS security architects rating each anti-pattern's prevalence and severity, or a small number of case-study applications of the full template against live CDI architectures with practitioner sign-off, would establish what this paper cannot: how common and how severe each anti-pattern actually is across the operating population of CDI environments, rather than only whether it recurs in documented evidence. Second, in the near-to-medium term, automated anti-pattern detection tooling could operationalise the detection indicators in Section 5 as queries against network discovery, identity management, and configuration management data, turning today's manual audit findings into continuous monitoring; this is a natural companion to the first direction, since detection tooling also generates the prevalence data that prospective validation requires. Third, empirical pattern-effectiveness studies, in the medium term, are needed to quantify, across multiple organisations and specifically within safety-critical OT control loops rather than cloud network contexts, how much risk reduction each mapped pattern in Table 5 delivers once implemented, extending the single-study, non-OT evidence currently available for patterns such as Adaptive Microsegmentation [23]. Fourth, digital twin validation, also medium-term, is suggested by recent surveys of digital twins for verification and validation of industrial automation systems: anti-pattern presence and pattern effectiveness, including the segmentation-versus-latency trade-off identified in Section 7, could be tested in simulation before committing to physical architecture changes [25]. Fifth, over a longer horizon, AI-assisted architecture review, applying large language models or graph-based classifiers trained on architecture documentation and as-built network topology

data, could surface candidate anti-patterns for human validation at a scale manual review cannot match, though this direction depends on the automated detection groundwork in the second direction and the labelled prevalence data from the first. Sixth, and open-ended, anti-pattern evolution: this catalogue reflects architectural conditions evidenced primarily in on-premises IT/OT environments, and emerging deployment models, cloud-hosted SCADA and historian platforms, private 5G replacing wired field-device networks, and AI/ML components embedded directly in control loops, plausibly introduce new trust, dependency, and visibility conditions that do not map cleanly onto the twelve anti-patterns catalogued here. Tracking whether this catalogue requires revision, extension, or a companion catalogue as these technologies mature is accordingly this paper's most open-ended and longest-running research direction, alongside the physical- and personnel-security extension noted in Section 6.2, and cross-sector benchmarking extending the incident evidence base in this paper beyond energy, water, and manufacturing into healthcare and transportation, which would test whether the twelve-anti-pattern catalogue generalises as claimed or requires sector-specific extension.

10. Conclusions

Architectural weaknesses in Critical Digital Infrastructure are often recurrent rather than accidental. The same handful of structural conditions, flat networks, implicit trust, shared administrative domains, unprotected legacy protocols, concentrated trust anchors, monitoring blind spots, monolithic perimeters, static policy, fragile recovery, undetected drift, concentrated dependency, and late-stage security involvement, reappear across sectors and across incidents, from Maroochy in 2000 to Colonial Pipeline in 2021, that are otherwise unrelated in attacker, target, and timeline. This recurrence is precisely what distinguishes an anti-pattern from an isolated mistake, and it is why a catalogue, rather than a single case study, is the appropriate response.

By identifying these twelve recurring anti-patterns, analysing how they enable and reinforce one another, and mapping each to one or more corresponding, standards -aligned security patterns, this paper gives organisations a means to proactively eliminate structural weaknesses before they evolve into systemic cyber risk, rather than discovering them retrospectively through incident forensics as occurred at Maroochy, in Ukraine, at the TRISIS-affected facility, at SolarWinds, at Okta, and at Colonial Pipeline. The structured, nine-field template used throughout Section 5 ensures that this identification is auditable and comparable rather than anecdotal, and the dependency analysis in Section 6.3 shows that these anti-patterns are not an unordered list but a structured, partially causal hierarchy in which governance failures enable, and therefore warrant remediation ahead of, the more visible technical failures they produce.

This paper's twelve anti-patterns are, by design, evidenced rather than exhaustively validated: each is corroborated by at least two independent sources spanning at least two sectors, but prospective validation by practitioners who did not participate in constructing the catalogue remains the most important open task, one this paper deliberately assigns to future work in Section 9 rather than claims to have completed. The proposed anti-pattern catalogue is offered, on that basis, as a practical foundation for architecture-first cybersecurity in Critical Digital Infrastructure, not as a closed or final taxonomy. Its value lies in giving architects, operators, auditors, regulators, and researchers a shared, evidence-grounded vocabulary, including its implementation barriers and remediation trade-offs, for the structural mistakes that standards compliance alone does not catch, and in directly connecting each named mistake to the architectural pattern, or patterns, that correct it.

REFERENCES

[1] W. J. Brown, R. C. Malveau, H. W. McCormick, and T. J. Mowbray, AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis. New York, NY, USA: Wiley, 1998.

[2] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, Security Patterns: Integrating Security and Systems Engineering. Chichester, U.K.: Wiley, 2006. [Online]. Available: https://dl.acm.org/doi/10.5555/1076903.

[3] J. Yoder and J. Barcalow, "Architectural patterns for enabling application security," in Proc. 4th Conf. Pattern Languages of Programs (PLoP), Monticello, IL, USA, Sep. 1997. [Online]. Available: https://hillside.net/plop/plop97/Proceedings/yoder.pdf.

[4] The Open Group, "The TOGAF Standard, 10th Edition," The Open Group, Reading, U.K., 2022. [Online]. Available: https://www.opengroup.org/togaf.

[5] J. Sherwood, A. Clark, and D. Lynas, Enterprise Security Architecture: A Business-Driven Approach. Boca Raton, FL, USA: CRC Press, 2005.

[6] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST Special Publication 800-207, Aug. 2020, doi: 10.6028/NIST.SP.800-207.

[7] National Institute of Standards and Technology, "Guide to Operational Technology (OT) Security," NIST, Gaithersburg, MD, USA, NIST Special Publication 800-82r3, Sep. 2023, doi: 10.6028/NIST.SP.800-82r3.

[8] National Institute of Standards and Technology, "The NIST Cybersecurity Framework (CSF) 2.0," NIST, Gaithersburg, MD, USA, NIST Cybersecurity White Paper 29, Feb. 2024, doi: 10.6028/NIST.CSWP.29.

[9] International Society of Automation, "ISA/IEC 62443 Series of Standards," ISA, Research Triangle Park, NC, USA. [Online]. Available: https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards.

[10] T. J. Williams, "The Purdue Enterprise Reference Architecture," Computers in Industry, vol. 24, nos. 2-3, pp. 141-158, Sep. 1994, doi: 10.1016/0166-3615(94)90017-5.

[11] Cybersecurity and Infrastructure Security Agency, "Dark Side ransomware: Best practices for preventing business disruption from ransomware attacks," CISA, Washington, DC, USA, Cybersecurity Advisory AA21-131A, May 2021. [Online]. Available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa21-131a.

[12] Cybersecurity and Infrastructure Security Agency, "Advanced persistent threat compromise of government agencies, critical infrastructure, and private sector organizations," CISA, Washington, DC, USA, Cybersecurity Advisory AA20-352A, Dec. 2020. [Online]. Available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa20-352a.

[13] Electricity Information Sharing and Analysis Center and SANS Industrial Control Systems, "Analysis of the cyber attack on the Ukrainian power grid," E-ISAC/SANS ICS, Defense Use Case Report, Mar. 2016. [Online]. Available: https://nsarchive.gwu.edu/sites/default/files/documents/3891751/SANS-and-Electricity-Information-Sharing-and.pdf.

[14] Cybersecurity and Infrastructure Security Agency, "Cyber-attack against Ukrainian critical infrastructure," CISA, Washington, DC, USA, ICS Alert IR-ALERT-H-16-056-01, Feb. 2016. [Online]. Available: https://www.cisa.gov/news-events/ics-alerts/ir-alert-h-16-056-01.

[15] Cybersecurity and Infrastructure Security Agency, "MAR-17-352-01 HatMan - Safety system targeted malware (Update B)," CISA, Washington, DC, USA, Malware Analysis Report, 2019. [Online]. Available: https://www.cisa.gov/sites/default/files/documents/MAR-17-352-01%20HatMan%20-%20Safety%20System%20Targeted%20Malware%20(Update%20B).pdf.

[16] Dragos, Inc., "TRISIS: Analyzing safety system targeted malware," Dragos, Inc., Hanover, MD, USA, Dec. 2017. [Online]. Available: https://www.dragos.com/blog/trisis/TRISIS-01.pdf.

[17] G. Murray, M. N. Johnstone, and C. Valli, "The convergence of IT and OT in critical infrastructure," in Proc. 15th Australian Information Security Management Conf., Perth, Australia, Dec. 2017, pp. 149-155. [Online]. Available: https://ro.ecu.edu.au/ism/217/.

[18] S. Kapoor, S. Kumar, and H. Vardhan, "Cyber security of OT networks: A tutorial and overview," arXiv:2502.14017, Feb. 2025. [Online]. Available: https://arxiv.org/abs/2502.14017.

[19] S. East, J. Butts, M. Papa, and S. Shenoi, "A taxonomy of attacks on the DNP3 protocol," in Critical Infrastructure Protection III, C. Palmer and S. Shenoi, Eds., IFIP Advances in Information and Communication Technology, vol. 311. Berlin, Germany: Springer, 2009, pp. 67-81, doi: 10.1007/978-3-642-04798-5_5.

[20] Cybersecurity and Infrastructure Security Agency, "Shifting the balance of cybersecurity risk: Principles and approaches for secure by design software," CISA, Washington, DC, USA, Oct. 2023. [Online]. Available: https://www.cisa.gov/sites/default/files/2023-10/SecureByDesign_1025_508c.pdf.

[21] B. Krebs, "Okta: Breach affected all customer support users," KrebsOnSecurity, Nov. 2023. [Online]. Available: https://krebsonsecurity.com/2023/11/okta-breach-affected-all-customer-support-users/.

[22] R. Verdecchia, P. Kruchten, P. Lago, and I. Malavolta, "Building and evaluating a theory of architectural technical debt in software-intensive systems," Journal of Systems and Software, vol. 176, art. 110925, Jun. 2021, doi: 10.1016/j.jss.2021.110925.

[23] S. Arora and J. Hastings, "Microsegmented cloud network architecture using open-source tools for a zero trust foundation," in Proc. 17th IEEE Int. Conf. Security of Information and Networks (SIN), 2024. [Online]. Available: https://arxiv.org/abs/2411.12162.

[24] Cybersecurity and Infrastructure Security Agency, "Critical Infrastructure Sectors," CISA, Washington, DC, USA. [Online]. Available: https://www.cisa.gov/topics/critical-infrastructure-security-and-resilience/critical-infrastructure-sectors.

[25] A. Locklin, M. Muller, T. Jung, N. Jazdi, D. White, and M. Weyrich, "Digital twin for verification and validation of industrial automation systems - a survey," in Proc. 25th IEEE Int. Conf. Emerging Technologies and Factory Automation (ETFA), 2020, pp. 851-858, doi: 10.1109/ETFA46521.2020.9212051.

[26] J. Slay and M. Miller, "Lessons learned from the Maroochy water breach," in Critical Infrastructure Protection, E. Goetz and S. Shenoi, Eds., IFIP International Federation for Information Processing, vol. 253. Boston, MA, USA: Springer, 2007, pp. 73-82. [Online]. Available: https://dl.ifip.org/db/conf/ifip11-10/cip2007/SlayM07.pdf.

[27] IEC 62443-3-2:2020, Security for Industrial Automation and Control Systems - Part 3-2: Security Risk Assessment for System Design, International Electrotechnical Commission, Geneva, Switzerland, 2020. [Online]. Available: https://webstore.iec.ch/en/publication/30727.

[28] SAFECode, "Fundamental Practices for Secure Software Development," 3rd ed., Software Assurance Forum for Excellence in Code, Mar. 2018. [Online]. Available: https://safecode.org/wp-content/uploads/2018/03/SAFECode_Fundamental_Practices_for_Secure_Software_Development_March_2018.pdf.

[29] MITRE Corporation, "ATT&CK for Industrial Control Systems," MITRE, Bedford, MA, USA. [Online]. Available: https://attack.mitre.org/matrices/ics/.

[30] G. Falco, A. Viswanathan, C. Caldera, and H. Shrobe, "A master attack methodology for an AI-based automated attack planner for smart cities," IEEE Access, vol. 6, pp. 48360-48373, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8449268.

[31] K. Stouffer, T. Zimmerman, C. Tang, J. Lubell, J. Cichonski, and J. McCarthy, "Cybersecurity Framework Version 1.1 Manufacturing Profile," National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST IR 8183r1, Oct. 2020, doi: 10.6028/NIST.IR.8183r1.

[32] European Union Agency for Cybersecurity, "Protecting Industrial Control Systems: Recommendations for Europe and Member States," ENISA, Athens, Greece, Dec. 2011. [Online]. Available: https://www.enisa.europa.eu/publications/protecting-industrial-control-systems.-recommendations-for-europe-and-member-states.

[33] A. Scott, "Tactical Data Diodes in Industrial Automation and Control Systems," SANS Institute Information Security Reading Room, Jun. 2015. [Online]. Available: https://www.sans.org/reading-room/whitepapers/firewalls/tactical-data-diodes-industrial-automation-control-systems-36057.

[34] Okta Security, "Unauthorized access to Okta's support case management system: Root cause and remediation," Okta Security, Nov. 2023. [Online]. Available: https://sec.okta.com/articles/2023/11/unauthorized-access-oktas-support-case-management-system-root-cause/.