PE

PROFESSIONAL ENGINEERING CONTINUING EDUCATION

Engineering Cybersecurity

Operational Technology Risk

Self-Paced Course Study Guide

PHYSICAL MISSION

Safe operation and dependable engineered function.

→

CONTROLLED TRUST

Known assets, defined paths and named identities.

→

VERIFIED RECOVERY

Protected configurations and demonstrated restoration.

Cybersecurity controls support safe physical operation, bounded trust and verified recovery.

COURSE	CREDIT / CATEGORY	INSTRUCTOR
ENG-011	2 CEH / PDH · Pertinent to Engineering Professional Practice / Technology Risk	Matthew Ward, MBA, CIA

Superior Learning, LLC d/b/a BuilderCE

For professional engineers, project leaders and engineering managers.

ENGINEERING CYBERSECURITY & OPERATIONAL TECHNOLOGY RISK

2 CEH

PERTINENT

ENGINEERING

Course Overview

Operational technology (OT) includes programmable systems and devices that monitor or directly influence physical processes. In engineering practice, OT appears in industrial control systems, supervisory control and data acquisition (SCADA), building automation systems, electrical and utility controls, transportation systems, process skids, safety systems, environmental monitoring, access control, and many other cyber-physical applications. As these systems become more connected to enterprise networks, cloud services, vendor support channels, and remote operations, the cyber risk becomes part of the engineering risk landscape.

This course is designed for professional engineers, engineering managers, project leaders, facility engineers, and technical staff who participate in the planning, design, procurement, integration, operation, maintenance, or oversight of connected engineering systems. It does not teach offensive security techniques. Instead, it gives engineers a practical framework for recognizing cyber-physical risk, working effectively with cybersecurity and IT specialists, making sound engineering decisions about architecture and controls, and supporting safe and resilient operations.

The central premise is that OT security cannot be reduced to conventional office-IT practices. Security decisions must be evaluated alongside safety, reliability, maintainability, process continuity, environmental obligations, and the realities of long-lived specialized equipment. An engineering team may be unable to patch a controller immediately because an outage is unavailable; a vendor may require remote access to maintain critical equipment; a building automation server may use a legacy operating system; or a utility may depend on a communications path that was never designed with modern threat assumptions. These constraints do not eliminate the cybersecurity obligation. They change the way the risk must be managed.

Throughout the course, the learner will use system boundaries, asset inventories, criticality, attack-surface mapping, segmentation, defense in depth, access control, remote-access governance, legacy-system compensating controls, backups and recovery, vendor risk, incident escalation, and procurement requirements to translate cybersecurity concerns into engineering actions.

Learning Objectives

- Distinguish information technology from operational technology and explain why safety, availability, reliability, and physical consequence change cybersecurity decisions.
- Identify OT assets, system boundaries, critical dependencies, remote-access pathways, and other attack-surface elements relevant to engineering systems.
- Apply engineering-focused controls including segmentation, defense in depth, identity and access management, secure configuration, monitoring, and controlled remote access.
- Describe practical approaches to legacy systems, vendor connectivity, secure procurement, backups, recovery, and operational resilience.
- Recognize abnormal cyber-physical behavior and support incident escalation, containment, recovery, engineering validation, and post-incident improvement.

> **Course boundary.** This course addresses defensive cybersecurity and operational resilience from an engineering professional-practice perspective. It does not teach offensive security, penetration testing, exploit development, or discipline-specific control-system design.

Study plan and navigation

MODULE	MINUTES	FOCUS
1	25	Cybersecurity in Engineering & Operational Technology
2	25	Assets, Architecture & Attack Surface
3	25	Engineering Controls & Secure System Design
4	25	Legacy Systems, Vendors & Operational Resilience
5	20	Incident Response, Escalation & Applied Engineering Scenarios

Planned instructional time: 120 minutes, exclusive of the final examination. Individual study time may vary.

How to complete this course

Read all modules in this course. No separate module acknowledgement is required. When you have finished reading, select Continue at the bottom of the BuilderCE lesson page to begin the 10-question final examination. If you are reading the PDF, return to that webpage first. A score of at least 70% is required to pass. Retakes are permitted.

Reference architecture and maintenance path · Completed incident record · Sources

MODULE 1 | 25 MINUTES

Cybersecurity in Engineering & Operational Technology

1.1 Why Cybersecurity Is an Engineering Issue

Cybersecurity becomes an engineering issue whenever a digital system can influence a physical outcome. A conventional business-system outage may interrupt email, invoicing, or document access. An OT outage can stop pumps, disable ventilation, interrupt power distribution, disrupt traffic control, change a process setpoint, suppress an alarm, degrade water treatment, or prevent an operator from seeing the true state of equipment. The engineering consequence is therefore not confined to information loss. It can involve safety, product quality, environmental compliance, service continuity, equipment damage, and public impact.

Engineers do not have to become penetration testers to manage this risk responsibly. They do need to understand how connectivity, system architecture, vendor access, configuration practices, unsupported equipment, and recovery capabilities affect engineered performance. During design, that means asking whether a control system is unnecessarily exposed to enterprise or internet-connected networks. During procurement, it means considering support lifecycle, authentication, logging, backup capability, and vendor remote-access requirements. During operations, it means understanding which systems are critical, who can change them, how abnormalities are detected, and how the process can be restored safely.

Cybersecurity also changes traditional engineering assumptions about failure. A component may be healthy mechanically and electrically yet behave incorrectly because logic, configuration, credentials, communications, or data have been compromised. Multiple redundant devices can fail in a correlated way if they depend on the same network, engineering workstation, vendor account, or software image. A remote connection intended for convenience can create a path around physical controls. These failure modes require multidisciplinary thinking.

The practical goal is not zero cyber risk. Connected systems exist because connectivity creates operational value. The goal is to make that connectivity intentional, bounded, monitored, and

recoverable. Engineers contribute by defining what the system must do safely, what dependencies are acceptable, what failure modes are tolerable, and what evidence is needed before a control is considered effective.

1.2 IT and OT Have Different Operating Priorities

IT and OT use many of the same technologies, but their operational priorities often differ. Enterprise IT commonly emphasizes confidentiality, integrity, and availability of data and business services. OT must also preserve integrity and availability, but the physical mission can make continuous operation, deterministic behavior, safety, and process stability dominant concerns. A security change that is routine on an office laptop may require extensive testing before it is applied to a controller that runs a continuous industrial process.

Asset lifecycles also differ. Enterprise computers may be replaced every few years. Industrial controllers, protective relays, building automation devices, and specialized instrumentation may remain in service for decades. Vendors may stop supporting software while the physical plant remains economically viable. Interfaces may use specialized or legacy protocols. The organization may have limited redundancy and only narrow outage windows. These conditions make standard patch-and-replace assumptions unrealistic.

The difference is not an excuse to leave OT insecure. It means controls should be selected with the physical system in mind. When patching cannot occur immediately, exposure may be reduced with segmentation, access restrictions, monitoring, application allowlisting where appropriate, disabled unused services, or other compensating controls. When a new security appliance is introduced, engineers should consider latency, failover behavior, maintenance, environmental conditions, and what happens if the appliance itself fails.

Effective programs therefore combine cybersecurity expertise with process and engineering expertise. Cybersecurity specialists understand threats, security architecture, identity, detection, and incident response. Operations and engineering personnel understand the process, safe states, equipment dependencies, outage constraints, and the meaning of abnormal behavior. Decisions are strongest when those perspectives are integrated rather than one group imposing generic controls on the other.

ENTERPRISE IT	OPERATIONAL TECHNOLOGY
Information services and business processes.	Physical processes, equipment, safety and service continuity.
Change cycles and recovery priorities depend on the business system.	Changes may require outages, process validation and long-lived equipment support.

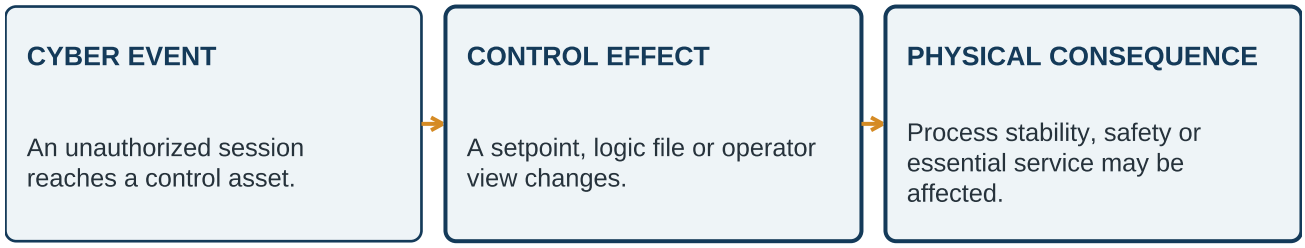
1.3 The Cyber-Physical Consequence Chain

A useful way to analyze OT cyber risk is to trace a consequence chain from the digital event to the physical outcome. The chain may begin with unauthorized remote access, compromised credentials, malicious software, a vulnerable engineering workstation, an altered configuration file, or an unexpected command. The next question is what control-system effect could result: loss of operator visibility, false sensor values, unauthorized logic changes, communications failure, or loss of a controller. Only then should the team ask how the physical process may respond.

For example, changing a pump setpoint is not itself the final consequence. The physical consequence may be insufficient flow, low tank level, cavitation, downstream process upset, or loss of fire-water capability. In a building automation system, loss of a supervisory server might cause local controllers to continue operating normally, while loss of a particular controller could affect ventilation in a critical space. In an electrical system, loss of monitoring may be tolerable for a short period even though unauthorized control of switching equipment is not.

This chain helps prioritize risk. Systems that appear similar from an IT perspective may have very different consequences because they perform different physical functions. It also improves incident response. If an abnormal cyber condition is detected, the team can focus first on safety and process stabilization rather than treating the event only as a network problem.

Engineers should document safe operating limits, critical functions, required manual fallback, and dependencies that can turn a cyber failure into a process failure. Those facts allow cybersecurity teams to prioritize protections around the physical mission. They also help leadership understand why a low-cost controller or minor communications link can deserve high protection if its failure creates a disproportionate consequence.



OT cybersecurity decisions connect a cyber event to control behavior and physical consequences.

1.4 Shared Responsibility Across Engineering, Operations, IT, and Cybersecurity

OT cybersecurity is a shared-responsibility problem. Engineering may specify the controller and network architecture. Operations owns the process and may control access to the facility. IT may manage enterprise identity, switching, firewalls, backups, or remote-access platforms. A cybersecurity team may monitor alerts and perform incident coordination. Vendors may maintain specialized equipment. If responsibility is unclear, security gaps form at the interfaces.

A practical responsibility model answers several questions. Who owns the asset inventory? Who approves new network connections? Who authorizes vendor remote access? Who manages privileged accounts? Who can change controller logic? Who reviews cybersecurity requirements in procurement specifications? Who monitors alerts? Who makes the decision to isolate equipment during an incident? Who validates that a restored system is safe to return to service? These responsibilities do not always belong to one department.

The engineering role is especially important where security controls affect function. An IT team may know how to disable a network route, but operations and engineering may need to confirm the consequence of doing so. A cybersecurity team may request an urgent patch, but the process engineer and system owner may need to determine whether applying it can affect validated behavior or require an outage. Likewise, engineering should not bypass security controls simply because a connection makes commissioning easier.

Successful organizations create agreed pathways before an incident. They know whom to call, which systems require special handling, what remote access is permitted, how emergency changes are recorded, and who can accept residual risk. The result is faster action with less improvisation when something abnormal occurs.

1.5 Risk Assessment for Cyber-Physical Systems

A useful OT risk assessment starts with the engineered mission, not with a vulnerability scanner. The team defines the system under consideration, the process functions that matter, the consequences of loss or manipulation, the pathways through which a cyber event could occur, the controls already in place, and the residual risk after those controls are considered. This sequence keeps the analysis connected to engineering consequence rather than producing a list of technical findings with no operational context.

The first question is consequence. What can happen if the system becomes unavailable, provides false information, accepts an unauthorized command, or is changed without authorization? Consequence categories may include personnel safety, public safety, environmental release, product or water quality, equipment damage, production loss, service interruption, regulatory impact, financial loss, and reputation. The team should distinguish direct process consequences from secondary business effects.

Likelihood is more difficult because cyber threats evolve and historical frequency may be a poor predictor. Rather than pretending to calculate a precise probability, many organizations use qualitative or semi-quantitative ratings informed by exposure, threat capability, known vulnerabilities, attractiveness of the target, access pathways, and control strength. The value of the assessment is consistency and prioritization, not false numerical precision.

Control effectiveness should be evaluated as installed and operated, not merely as designed. A firewall rule that no one reviews, a remote-access process that allows shared accounts, or a backup that has never been restored should not be credited at full value. Engineers are well positioned to test whether controls actually preserve the intended physical mission.

Residual risk is what remains after existing controls. If residual risk is above the organization's tolerance, the response may be additional controls, architectural redesign, replacement, increased monitoring, reduced connectivity, or formal acceptance by an authorized risk owner. The engineering team should be clear about which risks it can reduce technically and which require management decisions about cost, schedule, and operational tradeoffs.

1.6 Safety, Reliability, and Cybersecurity Are Interdependent

Safety, reliability, and cybersecurity are often managed by different programs, but in OT they can interact strongly. A cybersecurity control can improve safety by preventing unauthorized changes, yet it can also create a reliability concern if it blocks legitimate emergency access or introduces a new single point of failure. A reliability feature such as remote diagnostics can improve maintainability while increasing the cyber attack surface. A safety system may need strong independence from other networks precisely because its purpose is to remain trustworthy when the primary process system is not.

Engineers should therefore review cyber controls using familiar systems-engineering questions. What is the failure mode of the security device? Does the control fail open or fail closed, and what does that mean for the process? Is there bypass capability, and who can use it? Does the control add latency? Is time synchronization required? Can operators still see critical indications during an outage? Are backups and spares available for the security infrastructure itself?

Safety instrumented systems, protective relays, emergency shutdown systems, and other high-consequence controls may warrant additional separation and stricter change governance. The exact approach depends on the system and applicable standards, but the principle is stable: high-consequence protective functions should not inherit unnecessary trust from ordinary business or supervisory networks.

The same integrated thinking applies to resilience. Redundancy should be examined for common cyber dependencies. Two redundant controllers that use the same engineering workstation, shared credentials, and common update process may not be independent from a cyber perspective. A redundant network pair managed by the same compromised account may fail together. Cybersecurity can reveal common-mode dependencies that conventional reliability diagrams overlook.

The best design conversations therefore include safety, operations, reliability, and cybersecurity early enough that tradeoffs can be engineered rather than imposed after commissioning.

1.7 Common OT Threat Pathways Without Threat-Hunting Jargon

Engineers do not need an intelligence analyst's vocabulary to understand the pathways that repeatedly create OT risk. The most useful categories are straightforward: stolen or weak credentials, exposed remote access, compromised laptops and engineering workstations, vulnerable or unsupported software, infected removable media, vendor and supply-chain compromise, accidental misconfiguration,

and inappropriate connections between enterprise and control environments. Each pathway maps to practical engineering controls.

Credential misuse is especially important because an attacker using a legitimate account can look like an authorized user. Shared service credentials, default passwords, accounts that never expire, and vendor accounts left active after commissioning increase this risk. Strong authentication, named accounts, privilege limits, session logging, and access termination reduce the pathway without changing the physical process.

Compromised engineering workstations are another recurring concern. The workstation may legitimately communicate with many controllers and therefore can bypass network protections that block ordinary devices. Dedicated workstations, controlled software installation, restricted internet use, baseline configuration, removable-media procedures, and recoverable images reduce the chance that the engineering tool becomes the attack vehicle.

Vulnerable software matters most when it is reachable. A serious vulnerability on an isolated device may present less immediate risk than a moderate weakness on an internet-connected gateway. Engineers should therefore resist ranking risk by vulnerability score alone. Exposure, function, consequence, and compensating controls change the real priority.

Accidental cyber events deserve equal attention. A technician can load the wrong logic, connect the wrong network cable, create a routing loop, or install an incompatible update without malicious intent. Good cybersecurity engineering also reduces these errors through change control, configuration baselines, access limits, labeling, testing, and verification.

The practical benefit of this threat-pathway view is actionability. Rather than asking engineers to predict every adversary technique, it asks whether the system has unnecessary pathways and whether the organization can detect and recover when a trusted pathway is misused.

1.8 Translating Cyber Risk into an Engineering Risk Register

A cyber risk register can use the same disciplined logic engineers already apply to technical and project risks, but the entries should be written in terms of operational consequence rather than vague statements such as “cyberattack possible.” A useful risk statement identifies the condition or pathway, the event, and the consequence. For example: “Persistent vendor remote access with shared credentials could permit unauthorized control-system access, resulting in loss of process visibility or unauthorized configuration changes.”

The register should record the affected system, owner, current controls, residual risk, planned treatment, due date, and escalation status. High-consequence items should identify who has authority to accept the residual risk if mitigation is delayed. This is especially important when engineering teams recommend replacement or architectural changes that require capital funding.

Risk registers are most useful when they connect to real decisions. An unsupported controller may remain open for months because replacement is scheduled during a future outage. The register can document segmentation, access restrictions, monitoring, spare strategy, and the funded replacement date. A vendor gateway may be accepted temporarily if multifactor authentication is unavailable, provided access is disabled except during approved maintenance and sessions are monitored. These are explicit, reviewable decisions rather than invisible exceptions.

Risk treatment should distinguish reduction from transfer or acceptance. Buying cyber insurance does not reduce the physical likelihood of a process disruption. Contractually assigning responsibilities to a vendor does not remove the owner's operational exposure. Likewise, documenting acceptance does not make the technical risk disappear; it establishes that an authorized person understood the remaining exposure and chose to proceed.

The register should be reviewed when architecture changes, new external connections are added, vulnerabilities affect critical assets, support status changes, incidents occur, or major projects modify the process. Engineers can improve the quality of the register by ensuring the consequence language reflects actual system behavior and by challenging assumptions that a control is effective simply because it appears in policy.

A mature program does not attempt to list every conceivable cyber scenario. It maintains a manageable set of material risks that drive architecture, maintenance, procurement, monitoring, and capital decisions. The register becomes a bridge between technical observations and management action.

Applied Scenario 1 - Remote Access at a Water Facility

A water utility uses a vendor-supported SCADA system. A maintenance contractor requests permanent remote access because traveling to the site for routine support is expensive. The proposed approach would expose a remote desktop service through an internet-facing gateway and use one shared vendor credential. From a convenience perspective, the request is understandable. From a cyber-physical perspective, it creates a persistent pathway to a system that affects treatment and distribution.

A disciplined engineering review begins with the operational need. Is remote support necessary, or can access be enabled only during approved maintenance? Which systems must the vendor reach? Can the path terminate in a controlled jump host rather than directly on control-system devices? Can the user identity be individual, strongly authenticated, time-limited, logged, and disabled when the work ends? What happens if the remote service fails? Does the vendor require administrative privileges or only diagnostic access?

The team may decide that remote support is appropriate, but the final design should reflect the physical criticality of the system. The key lesson is that cybersecurity is not a binary choice between connectivity and no connectivity. The engineering task is to define a controlled architecture that provides the operational benefit without creating unnecessary persistent trust.

MODULE 2 | 25 MINUTES

Assets, Architecture & Attack Surface

2.1 Asset Inventory as an Engineering Foundation

An OT asset inventory is more than a list of IP addresses. To support engineering risk decisions, the inventory should identify the asset, its physical function, system owner, location, hardware and software or firmware version where relevant, network zone, communications dependencies, vendor, remote-access method, criticality, backup or spare strategy, and lifecycle status. CISA's current OT asset-inventory guidance emphasizes that organizations cannot build a defensible architecture if they do not know what they have.

The inventory should include controllers, remote terminal units, HMIs, engineering workstations, historians, servers, network devices, gateways, wireless devices, sensors with communications capability, safety-system interfaces, vendor appliances, and software services that support the OT environment. It should also account for virtual machines, cloud connections, cellular gateways, and temporary devices that can become permanent in practice.

Criticality is what turns inventory into engineering information. A modest panel-mounted controller may be more important than an expensive server if it operates a process with no practical manual fallback. Conversely, a supervisory display may be important to operators while local control continues safely for a limited period. The inventory should help answer what happens when an asset is unavailable, compromised, or misleading.

Ownership matters because unowned assets tend to become unmanaged assets. Every critical item should have someone responsible for lifecycle decisions, approved connectivity, backup and restoration, vendor support, and security requirements. Where ownership is split between IT and operations, the split should be explicit rather than assumed.

ASSET ATTRIBUTE	ENGINEERING USE
Identity / owner	Product, version, responsible owner and approved baseline.
Function / criticality	Physical duty, consequence and acceptable degraded state.
Dependencies / connectivity	Required services, approved paths and recovery order.
Recovery	Protected configuration, tools, spares and tested restoration.

2.2 Architecture and Trust Boundaries

Architecture determines how far a compromise can travel and which systems must trust one another. A flat network may allow a device in one part of the facility to communicate broadly with unrelated controllers. A more defensible design groups assets by function and risk, controls communications between zones, and limits paths to those that are operationally necessary. ISA/IEC 62443 uses the concepts of zones and conduits to organize assets with similar security requirements and the communications paths between them.

Engineers should be able to read an OT architecture diagram as a functional model, not only a network diagram. Which components directly control the process? Which provide supervisory visibility? Which services exchange data with enterprise applications? Where does remote access enter? Where are vendor systems connected? Which services are routed through an intermediate or demilitarized zone? Which communications are required continuously, and which are only needed during maintenance?

A trust boundary is any point where the risk context changes. Moving from a business network into an OT zone is a major trust transition. So is connecting a vendor laptop, adding a cellular modem, attaching removable media, or allowing a cloud service to issue commands rather than simply receive data. Security controls should be strongest where high-consequence systems meet less-trusted environments.

The architecture should also support failure containment. If an enterprise network is unavailable, can the plant continue operating? If a historian is compromised, can it influence controller logic? If a vendor connection is disabled, does the process remain controllable locally? These questions turn segmentation from a cybersecurity diagram into a resilience strategy.

FUNCTIONAL LEVEL	BOUNDARY TO UNDERSTAND
Enterprise / business	Business integration and remote dependencies.
DMZ / operations	Controlled shared services and supervisory functions.
Control / field	Controllers, operator interfaces, sensors and actuators; separate safety-related functions as needed.

2.3 Mapping the Attack Surface

The attack surface is the collection of pathways through which an unauthorized party or compromised device could influence the OT environment. Common pathways include remote-access services, vendor connections, engineering workstations, IT/OT interfaces, wireless and cellular communications, removable media, cloud services, exposed web interfaces, default credentials, and legacy protocols that provide little native authentication.

Attack-surface mapping should begin with data and command flows. What enters the OT system? What leaves? Which connections are bidirectional? Which systems can write configuration or logic? Which

devices accept remote administration? Which ports and services are necessary? A communications path that exists only because a default installation enabled it should not receive the same treatment as a path required for operations.

Temporary connectivity deserves special attention. Commissioning teams may add wireless access, direct laptop connections, temporary internet service, or vendor VPNs. Those paths can persist long after their original purpose ends. The same is true of emergency workarounds. A bypass that was reasonable for two days becomes an unmanaged exposure if no one owns removal.

Mapping is most useful when it is paired with criticality. A low-probability path to a high-consequence safety system may deserve more attention than a frequently used path to a noncritical reporting server. The objective is not to document every theoretical threat. It is to identify realistic pathways that matter to the physical mission.

ATTACK-SURFACE PATH	ENGINEERING CONTROL QUESTION
Remote / vendor / cloud	Who can connect or issue commands, to what destination and for how long?
Workstation / portable media	What can introduce configuration changes or executable content?
Wireless / network interface	Which actual paths cross trust boundaries and how are they constrained?
Legacy / support dependency	What cannot be patched, and what bounds exposure and supports recovery?

2.4 Engineering Workstations and Portable Media

Engineering workstations are high-value assets because they may contain controller programming software, configuration tools, project files, credentials, firmware, device libraries, and direct communications capability. A compromised workstation can be more dangerous than an ordinary office computer because legitimate engineering tools can be used to make legitimate-looking changes to physical systems.

Organizations should decide whether engineering workstations are dedicated to OT use, how they receive software updates, whether ordinary email and web browsing are permitted, how local administrator rights are handled, and how removable media is controlled. A laptop used both for internet browsing and direct controller programming creates a bridge between very different trust environments.

Portable media can be operationally necessary in isolated facilities. The risk should be managed through approved media, scanning or staging processes, controlled transfer procedures, and clear ownership. The process should be practical enough that field staff can follow it. Controls that are impossible during real maintenance often lead to informal bypasses.

Configuration files and logic should be treated as critical engineering records. Teams should know which version is approved, who can modify it, where backups are stored, how changes are reviewed, and how the installed state is verified. This links cybersecurity directly to configuration management and quality assurance without duplicating those disciplines: the cyber question is whether an unauthorized or unverified change can enter the operational environment.

2.5 Remote, Wireless, and Cloud Connectivity

Remote connectivity changes the system boundary. Cellular modems, vendor cloud dashboards, remote telemetry, wireless sensors, and cloud-hosted analytics may provide substantial operational benefit, but each introduces dependencies and trust relationships that must be understood. Engineers should ask whether the connection is inbound, outbound, or bidirectional; whether it can issue commands; how identity is managed; where data is stored; and what happens when the external service is unavailable.

Cloud connectivity is not automatically insecure and local connectivity is not automatically safe. The relevant questions are architecture, identity, monitoring, failure behavior, and consequence. An

outbound-only telemetry connection from an isolated gateway may have a very different risk profile from a cloud platform that can remotely modify controller parameters.

Wireless systems require similar discipline. Coverage, availability, interference, authentication, encryption, and device management affect both reliability and security. A remote radio that is physically difficult to reach may be harder to patch or inspect. A cellular gateway with a default management interface may create unexpected exposure.

Design documents should therefore show external dependencies explicitly. If the process relies on a third-party cloud, cellular carrier, managed service, or remote vendor, that dependency belongs in the engineering risk discussion, the asset inventory, and the recovery plan.



Contextual illustration of pumping equipment. Cyber decisions ultimately support physical system function; this is not a documented incident or verified installation.

2.6 Asset Criticality and Dependency Mapping

Criticality should be based on function and consequence, not asset cost or visual prominence. An inexpensive network switch can be critical if its failure disconnects all operators from a process. A large historian server may be operationally important but not immediately safety critical if local control continues. A small cellular gateway may represent a high cyber risk if it provides remote access into an otherwise isolated system.

A practical criticality method evaluates several dimensions: safety consequence, environmental consequence, service or production impact, quality impact, ability to operate manually, recovery time, availability of spares, and dependency on external services. The team can use a simple high/medium/low scale if the criteria are defined consistently. The purpose is to direct attention and resources toward assets whose compromise matters most.

Dependency mapping extends the analysis beyond the asset itself. A PLC may depend on an HMI for operator visibility, a time server for sequence-of-events data, a network switch for communications, an engineering workstation for maintenance, and a vendor license server for configuration software. A cloud-managed gateway may depend on internet service and vendor identity infrastructure. These dependencies can create unexpected single points of failure.

Engineers should also identify upstream and downstream dependencies between physical processes. A water treatment process may depend on upstream raw-water pumping and downstream storage. A building automation system may influence air handling, chilled water, and pressure relationships. A transportation control center may depend on field communications and enterprise identity services.

Cyber incidents can propagate through these functional relationships even when networks are segmented.

Dependency maps improve recovery planning because they show restoration order. Restoring an HMI before the network and controller it depends on accomplishes little. They also improve procurement because the organization can recognize when a seemingly minor third-party service has become mission critical.

2.7 Data Flows, Command Paths, and the Difference Between Read and Write

Not all connectivity carries the same risk. A one-way or tightly constrained data feed from OT to an enterprise historian can have a different consequence than a bidirectional connection that allows enterprise systems to change controller parameters. Engineers should therefore document not only that two systems communicate, but what information moves, which side initiates the session, whether commands or configuration can flow, and what happens if the path is unavailable.

Data-flow diagrams can be simple. For each conduit or interface, identify source, destination, protocol or service, purpose, frequency, direction, authentication, and whether the communication is required for control, monitoring, maintenance, or business reporting. Unnecessary paths can then be removed or constrained.

Read-only concepts deserve scrutiny because “read-only” may be enforced at different layers. A reporting application may be intended to read data, but the underlying account or protocol may still have write capability. A firewall may permit only a limited service, yet that service itself may support administrative commands. The team should verify the actual control boundary rather than relying on a label.

Command paths are especially important for cloud and remote systems. If a cloud dashboard can modify setpoints, the connection is part of the control path and should be treated accordingly. If a vendor portal can push firmware, that path can influence device behavior even if it is not part of normal process control.

By distinguishing data visibility from command authority, engineers can often preserve operational analytics and business integration while reducing unnecessary cyber-physical exposure.

2.8 Protocols, Time, and Hidden Infrastructure Dependencies

Many OT protocols were designed for isolated industrial environments in which network participants were implicitly trusted. Some provide limited authentication, confidentiality, or command authorization compared with modern enterprise protocols. Engineers do not need to memorize protocol vulnerabilities, but they should recognize that protocol choice and placement affect how much trust a network connection carries.

Where legacy protocols are required, architecture becomes especially important. The team may limit the devices that can communicate, restrict the path through firewalls or industrial gateways, monitor unusual commands, or isolate the network from less-trusted environments. Replacing every legacy protocol may be impractical; controlling where the protocol can travel is often the more immediate engineering measure.

Time synchronization is another easily overlooked dependency. Controllers, protection systems, historians, event logs, and security monitoring may rely on consistent time. If clocks drift or a time source is unavailable, sequence-of-events reconstruction becomes difficult and alarms can appear misleading. In some systems, incorrect time can affect control logic or scheduled operations. The architecture should identify authoritative time sources, what happens when they fail, and whether less-trusted systems can influence them.

Name services, directory services, licensing servers, certificate services, and virtualization platforms can also become hidden dependencies. A control application may appear local while its login depends on an enterprise directory. A virtual HMI server may depend on shared storage. A vendor application may stop functioning when a license server or cloud authentication service is unavailable.

Dependency reviews should therefore include supporting infrastructure, not only controllers and HMIs. The engineering question is simple: if this supporting service disappears or becomes untrustworthy, what happens to the physical system and how long can operations continue?

2.9 Vulnerability Management in an OT Context

Vulnerability management is the process of learning about weaknesses, determining whether they affect installed assets, prioritizing response, and tracking the risk until it is resolved or accepted. In OT, the process must be connected to asset inventory and operational context. A vulnerability bulletin is useful only if the organization can determine whether the affected product and version are installed and how exposed the device is.

The first step is applicability. Confirm product family, model, software or firmware version, configuration, and whether the vulnerable feature is enabled. Next consider exposure. Is the device reachable from enterprise or remote networks? Is access restricted? Does exploitation require local access or valid credentials? Then consider consequence. What physical function does the asset perform, and what happens if it becomes unavailable or untrustworthy?

This context can change priority significantly. A high-severity vulnerability on a device isolated behind strong controls may allow time for planned remediation. A lower-scored weakness on an internet-facing remote gateway to a critical process may deserve urgent action. Engineers should therefore support risk-based prioritization rather than allowing a numeric score to become the entire decision.

Remediation options include patching, firmware updates, configuration changes, disabling vulnerable services, restricting access, segmentation, monitoring, replacing the device, or formally accepting the residual risk with compensating controls. The organization should document why remediation is deferred and when the decision will be revisited.

Testing remains important. A vendor security update may affect communications drivers, control logic, device performance, or compatibility with engineering tools. A representative test environment is ideal where consequence justifies it, but many facilities have limited lab capability. In those cases, the team may rely on vendor validation, staged deployment, backups, rollback planning, maintenance windows, and increased observation.

Vulnerability management should also account for false certainty. The absence of known vulnerabilities does not mean an asset is secure, especially when old products are no longer researched or supported. Conversely, a long list of published vulnerabilities does not automatically mean the system is imminently compromised. The engineering objective is to understand exposure and maintain a rational treatment process.

PATCH DECISION	EVIDENCE
Applicability / exposure	Installed product, version, enabled feature and reachable path.
Consequence / compatibility	Physical function, vendor support and process impact of the change.
Treatment / verification	Test, deploy or apply documented compensating controls; assign an owner and revisit date.

MODULE 3 | 25 MINUTES

Engineering Controls & Secure System Design

3.1 Defense in Depth for Engineering Systems

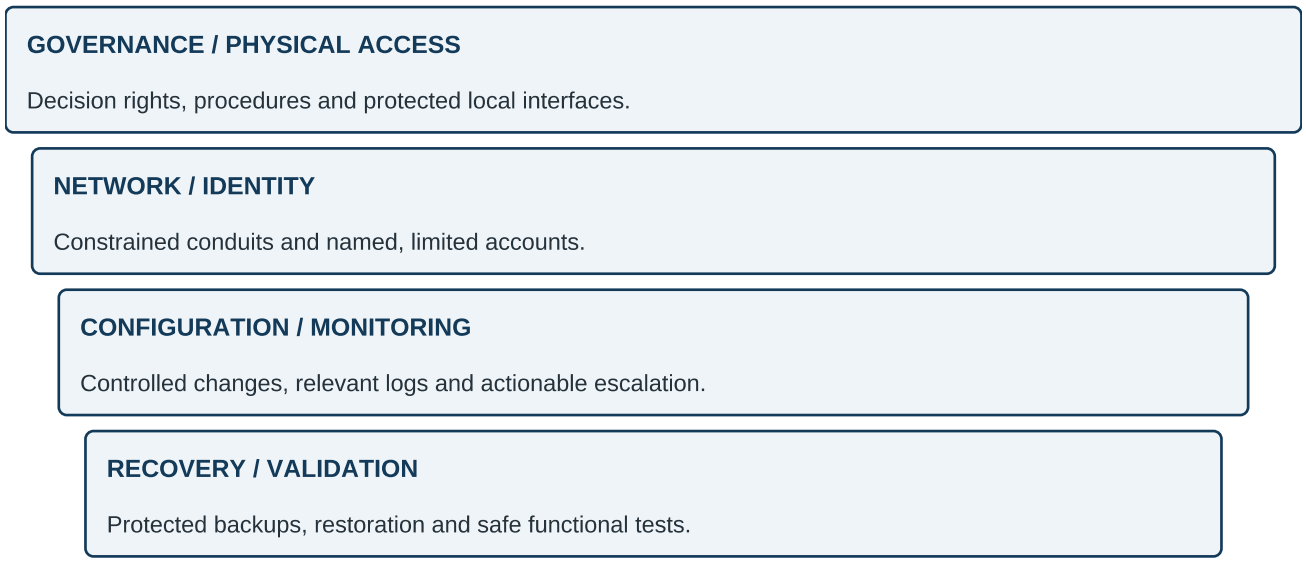
Defense in depth uses multiple independent layers so that failure of one control does not immediately expose the physical process. Layers can include governance and procedures, physical access, network architecture, identity and authentication, configuration controls, endpoint protections, monitoring,

backups, and incident-response capability. The value is not the number of controls; it is the independence and relevance of the layers.

A firewall does not compensate for shared administrator credentials. Strong passwords do not compensate for a flat network with unnecessary access. A segmented network does not compensate for an unmanaged engineering workstation that is authorized to cross the boundary. Likewise, a robust backup strategy does not prevent compromise but can reduce the consequence of destructive events.

Engineers should evaluate how security controls affect reliability. A protective device inserted into the communications path can become a single point of failure. An authentication mechanism may need an offline or local fallback for emergency operations. A logging system should not consume enough bandwidth or processor capacity to interfere with control traffic. Security design is strongest when cyber controls are engineered with the same attention to failure modes, maintainability, and commissioning as other system components.

The organization should also verify that controls remain effective after commissioning. Rules are changed, vendors add connections, projects expand, and temporary exceptions accumulate. Defense in depth is therefore a lifecycle discipline, not a one-time design review.



Independent and relevant layers support the physical mission; a larger count of controls is not proof of effectiveness.

3.2 Segmentation, Zones, and Conduits

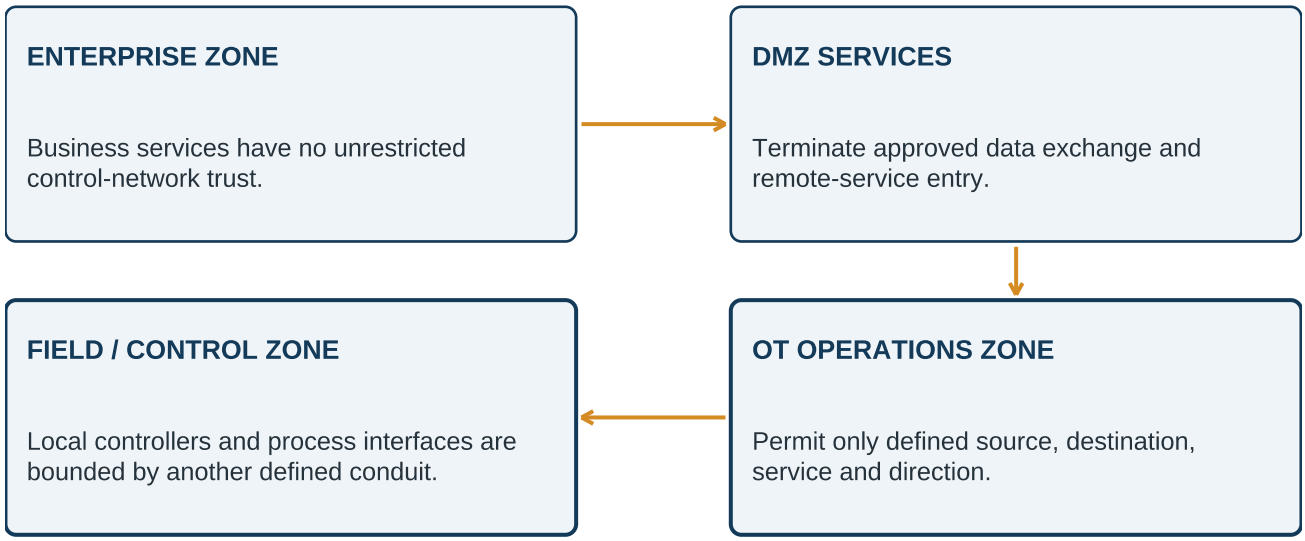
Segmentation limits unnecessary communication and reduces the chance that a compromise in one environment can move laterally into another. A practical design separates business functions, shared services, supervisory OT functions, controllers, safety-related functions where applicable, and vendor or remote-access pathways according to risk and operational need.

The ISA/IEC 62443 concept of zones and conduits is useful because it begins with security requirements rather than arbitrary network boundaries. Assets that share similar security needs can be grouped into a zone. Communications between zones occur through defined conduits. The design process asks which communications are necessary, who initiates them, what protocols are used, and what controls are required at the boundary.

Segmentation should be validated against operations. Maintenance personnel may need access to multiple zones. Historians may need data from control networks. Enterprise systems may need production data without being allowed to issue control commands. Remote vendors may need temporary access to one specific system. A well-designed architecture supports those needs through controlled paths rather than broad trust.

Engineers should be cautious about segmentation that exists only on paper. Misconfigured firewall rules, unmanaged switches, dual-homed computers, wireless bridges, and emergency bypass connections

can undermine the intended design. Verification should include both documentation and observation of the installed system.



Zones group assets with similar security needs. Defined conduits constrain necessary communications; this simplified model does not prescribe a universal topology.

3.3 Identity, Privilege, and Remote Access

Identity controls determine who can perform actions and whether those actions can be attributed to a specific person. Shared accounts are common in legacy OT environments because they simplify operations, but they reduce accountability and increase the risk that credentials remain active after staff or vendors change. Where technically feasible, privileged access should use named accounts, strong authentication, appropriate roles, and controlled escalation of administrative privileges.

Least privilege means users and systems receive only the access needed for their function. An operator may need to acknowledge alarms but not modify controller logic. A vendor may need diagnostic access to one subsystem but not unrestricted access across the control network. An engineering workstation may need programming access during maintenance but not continuous internet access.

Remote access deserves its own lifecycle: justify the need, approve the user, establish a controlled path, authenticate strongly, limit the destination and duration, monitor the session where appropriate, and terminate access when work ends. Persistent access should be treated as an architectural decision rather than a convenience setting.

Emergency-access procedures should exist before emergencies. If a cybersecurity control prevents safe restoration, staff need an approved method to obtain temporary elevated access without creating an undocumented permanent bypass. The record should show who authorized the exception, why it was needed, what was changed, and when normal controls were restored.

ACCESS LIFECYCLE	CONTROL TO VERIFY
Need / identity	Justified task, named approver and named user.
Path / privilege	Controlled entry, strong authentication and minimum destination/rights.
Window / monitoring	Approved duration and proportionate session evidence.
Closure	Terminate the session, remove the entitlement and check that access is denied.

3.4 Configuration, Patch, and Change Management

OT patching is risk management, not a reflex. A security update may close a vulnerability but also create compatibility, validation, or downtime concerns. The organization should know which assets are affected, whether the vendor supports the patch, how critical the vulnerability is, whether the system is exposed, what compensating controls exist, and how the update will be tested before deployment.

This is especially important for validated or safety-critical systems. Applying an update without understanding operational effect can create a different risk. Conversely, indefinitely deferring every patch because the system is “critical” allows exposure to accumulate. A risk-based process sets priorities, documents deferrals, applies compensating controls, and creates a credible maintenance window or upgrade plan.

Configuration control is equally important. Firewall rules, switch configurations, controller logic, HMI applications, security settings, and remote-access parameters should have approved baselines. Changes should be attributable, reviewed at a level proportional to consequence, backed up, and validated after implementation.

Engineers can add value by defining the functional acceptance criteria for changes. Cybersecurity staff may confirm that a vulnerability is closed, while engineering confirms that alarms, control loops, failover, communications, and process behavior still meet requirements. Secure change is complete only when both objectives are satisfied.

3.5 Monitoring, Logging, and Detection

OT monitoring should focus on behavior that matters. Useful signals can include failed logins, new remote sessions, unexpected controller downloads, configuration changes, new devices, abnormal network communications, disabled security services, time changes, or process behavior that conflicts with expected operating conditions. Security tools can provide alerts, but operators and engineers often recognize abnormal physical behavior first.

Logs are valuable only if the organization can access, interpret, and retain them. A device may technically produce logs that are never collected. A remote-access gateway may record sessions without anyone reviewing them. A controller may not support detailed event logs at all. The monitoring strategy should therefore combine technical telemetry with operational observations and change records.

Detection must also respect OT constraints. Aggressive active scanning can be inappropriate for fragile legacy devices. Security monitoring should be selected and tested for the environment. Passive monitoring, switch data, firewall logs, system events, and controlled vulnerability assessment may provide useful visibility without unnecessary process risk.

A practical design question is: what abnormal condition would make us act, and who receives the signal? Alerts without ownership create noise. Engineers should help define which cyber events could affect the process and what immediate operational checks should follow.

3.6 Secure Design Review and Commissioning

Cybersecurity should be reviewed at the same project milestones as other engineering requirements. During conceptual design, the team can define zones, external connections, critical assets, remote-support philosophy, and owner cybersecurity standards. During detailed design, drawings and specifications can identify network boundaries, approved protocols, account requirements, logging, backup, and vendor-access provisions. During procurement, submittals can demonstrate product capabilities. During commissioning, the installed configuration can be verified before turnover.

A secure design review should ask whether the architecture contains unnecessary pathways, whether high-consequence functions are appropriately separated, whether privileged access is controlled, whether the system can operate if enterprise services are unavailable, and whether required cyber controls create unacceptable reliability effects. Reviewers should also confirm that cybersecurity requirements are assigned to the correct party. Requirements that sit between electrical, controls, IT, and vendor scopes are easy to miss.

Commissioning is a high-risk transition because temporary arrangements are common. Vendors may use shared accounts, direct laptop connections, temporary wireless networks, open firewall rules, or internet access for troubleshooting. Those measures can be reasonable during controlled commissioning, but the project should have a turnover checklist that removes or converts them before final acceptance.

The turnover package should include current network and architecture drawings, asset inventory data, account and privilege lists, backup files, configuration baselines, approved remote-access procedures, vendor contacts, software and firmware versions, support and end-of-life information, and known exceptions. The owner should know not only how the system works but how to maintain its trusted state.

A cybersecurity acceptance test does not need to simulate an attack. It can verify observable controls: default credentials changed, named accounts established, unused services disabled, expected network paths documented, remote access controlled, logging available, backups created, and restoration procedure demonstrated at an appropriate level.

3.7 Control Verification and Evidence of Effectiveness

A control exists only if it operates as intended. Written policy may require multifactor authentication, but the actual vendor gateway may still allow a local shared account. A segmentation diagram may show an OT boundary, while a maintenance laptop has interfaces on both sides. A backup schedule may report success, but no one has verified that the controller project can be restored to compatible hardware.

Control verification turns cybersecurity from paperwork into engineering evidence. For network segmentation, evidence may include firewall rule review, connection testing, and confirmation that unauthorized paths are blocked. For identity controls, evidence may include account lists, privilege review, authentication settings, and terminated-user checks. For remote access, evidence may include session logs and confirmation that access expires. For configuration control, evidence may include baseline comparisons and change records.

Verification frequency should reflect consequence and change rate. A high-criticality remote-access gateway may justify more frequent review than an isolated low-risk controller. Controls that are repeatedly changed should be checked more often than static ones. The objective is to detect drift before it becomes an incident pathway.

Engineers can help define functional evidence. If a firewall rule is changed, does the control system still exchange the required data? If remote access is disabled, is local maintenance still possible? If an endpoint tool is installed, does the HMI remain stable? Cybersecurity effectiveness and operational acceptability should be demonstrated together.

This approach also supports audits and management review. Instead of asserting that a control is “implemented,” the system owner can show what was tested, when, by whom, and with what result.

3.8 Physical Security and Local Interfaces

OT cybersecurity is not only a network problem. Many devices have local programming ports, maintenance switches, USB interfaces, removable memory, cabinet doors, serial connections, or physical reset mechanisms. If an unauthorized person can reach those interfaces, network controls may be bypassed entirely. Physical protection should therefore be proportional to consequence and location.

A controller in a locked electrical room has a different exposure than a roadside cabinet, remote pump station, rooftop unit, or publicly accessible building space. Remote sites may need tamper-resistant enclosures, locked cabinets, access monitoring, controlled keys, or documented inspection. Portable programming ports should not be left exposed merely because the device is not connected to the internet.

Physical access also affects maintenance practice. If staff routinely prop open cabinets or share master keys, formal policy may not reflect the actual control. Engineers conducting site walks can identify these realities better than a remote cybersecurity assessment. Field observations should be incorporated into risk reviews.

Local interfaces may need procedural controls even when physical access is trusted. A USB port used for firmware updates can introduce malicious or incorrect files. A laptop connected directly to a controller can create a temporary bridge around segmentation. A local maintenance password written inside the cabinet defeats identity controls. The design should consider how legitimate maintenance will be performed and make the secure method practical.

Physical and cyber security also intersect during incidents. If a remote command is suspected, staff may need to verify switch positions, controller indicators, or local process values independently. Accessible, well-labeled field instrumentation and documented local operating procedures can improve both cyber resilience and ordinary troubleshooting.

3.9 Secure Remote Operations and Human Factors

Remote operations can improve staffing flexibility and response time, but they shift more authority into digital identity and communications. An operator who once had to be physically present may now control equipment from a distant location. That change deserves the same engineering scrutiny as any other modification to the control philosophy.

The remote operating environment should preserve situational awareness. Screen size, alarm presentation, communications delay, loss of voice contact with local staff, and inability to observe physical conditions can affect decision quality. Cyber controls such as session timeout and reauthentication should be designed so they do not create unsafe interruptions during critical operations while still limiting unattended access.

Human factors matter because security controls that are difficult to use often generate bypasses. If multifactor authentication takes several minutes during every routine maintenance task, staff may seek shared accounts. If the approved remote-access platform is unreliable, vendors may request direct alternatives. If passwords must be changed in a way incompatible with controllers, technicians may write them down. These are design signals, not excuses for weak security.

Engineers, operations personnel, and cybersecurity teams should observe real workflows and choose controls that support the mission. Training should explain why a control exists, what behavior it is intended to prevent, and how to handle exceptions. Emergency access should be tested so staff do not discover during an outage that the only authorized administrator is unavailable.

Remote operations should also be revocable. The organization should be able to disable a user, vendor, or pathway quickly without redesigning the system. This capability is valuable during both cyber incidents and ordinary personnel or contract changes.

3.10 Writing Cybersecurity Requirements into Engineering Specifications

Cybersecurity requirements are most effective when they are specific enough to be designed, priced, reviewed, and accepted. Generic language such as “system shall be secure” provides little guidance to a vendor and little basis for the owner to determine compliance. Engineering specifications can instead define outcomes that relate to the system architecture and operating model.

Examples include requiring unique credentials or named accounts where the product supports them; documenting all network ports, protocols, and external connections; supporting owner-approved authentication methods; providing logs of security-relevant events; allowing configuration backup and restoration; disabling unused services; documenting firmware and software versions; identifying vendor remote-access requirements; and providing support and end-of-life information. High-criticality systems may require additional controls based on organization standards or sector guidance.

Requirements should be coordinated across specification sections. Network switches may be supplied by one contractor, controllers by another, servers by a systems integrator, and remote access by an owner IT team. If no one owns the interface, cybersecurity obligations can fall between scopes. Drawings, control narratives, network diagrams, and responsibility matrices can clarify the intended architecture.

Submittal review should focus on deviations and dependencies. Does the product require an outbound cloud connection not shown in design? Does the vendor insist on a shared administrator account? Is a

cellular modem included by default? Does the equipment require an obsolete operating system? These facts may be acceptable, but they should be visible before purchase.

Acceptance criteria should be practical. The project can verify that default accounts were changed, remote access uses the approved path, unauthorized connections are blocked, configuration backups were delivered, specified logs are available, network diagrams match installation, and temporary commissioning access was removed. More specialized security testing may be appropriate for certain systems and should be performed by qualified personnel.

The commercial value of good specification is substantial. Cybersecurity features added after commissioning often require redesign, outages, change orders, or replacement. Clear requirements allow bidders to price the intended system and reduce disputes about whether a security capability was included.

Applied Scenario 3 - Building Automation Segmentation

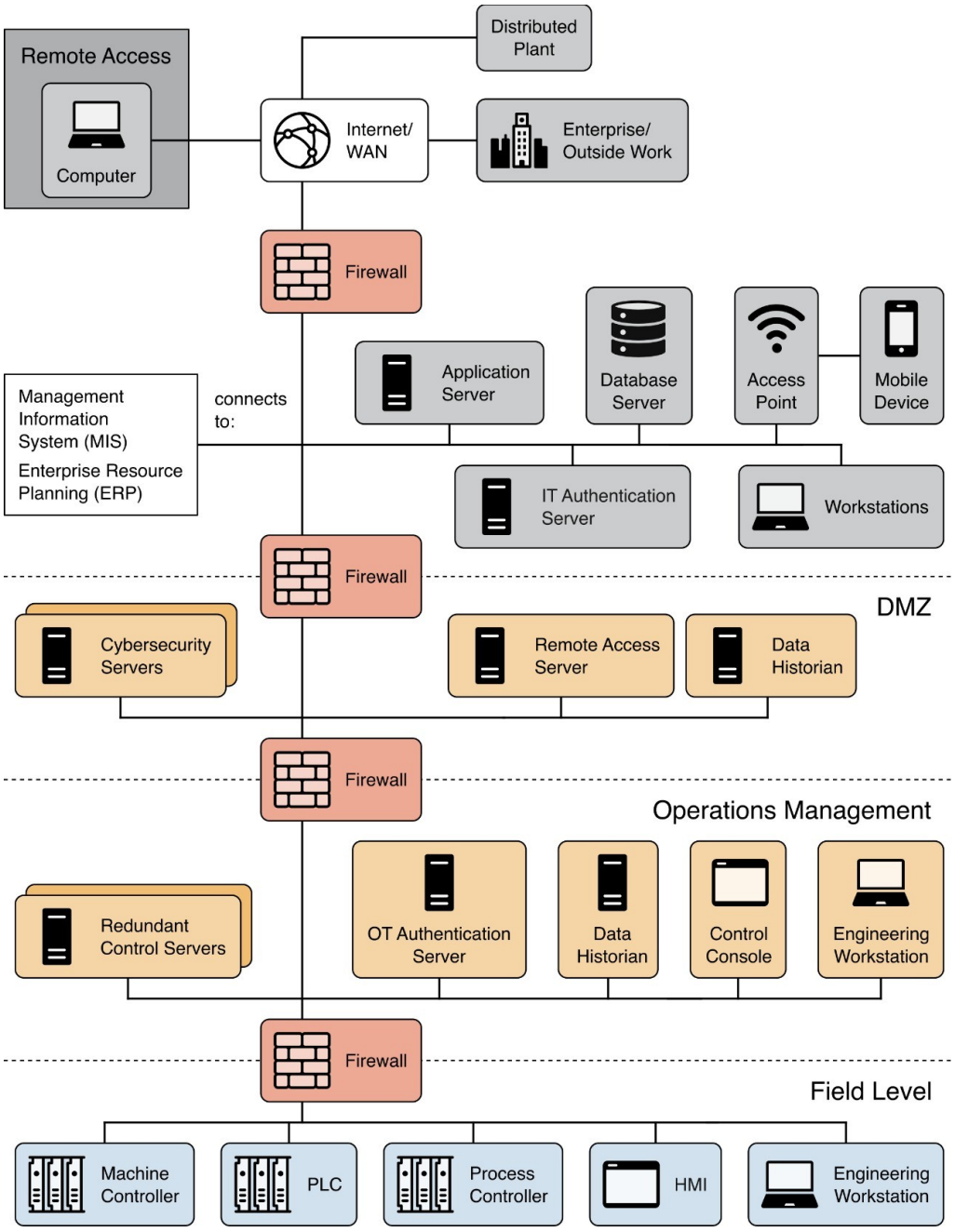
A hospital's building automation system controls air handling, chilled water, heating, and pressure relationships in critical spaces. The BAS server is connected to the same general business network used for ordinary workstations because facilities staff need remote access. A cybersecurity review recommends moving the BAS into a separate zone, routing approved access through a controlled gateway, and removing direct workstation-to-controller paths.

The engineering review considers more than the firewall change. Which systems require continuous communication? How will operators access the BAS during an enterprise outage? Are local controllers capable of autonomous operation? Does the gateway create a single point of failure? Which alarms must remain available? How will service vendors connect? The final design preserves the operational mission while reducing unnecessary trust.

The scenario demonstrates why segmentation is not an IT-only task. Cybersecurity defines the threat-reduction objective; engineering validates that the architecture remains safe, maintainable, and operable.

Reference architecture and the maintenance path

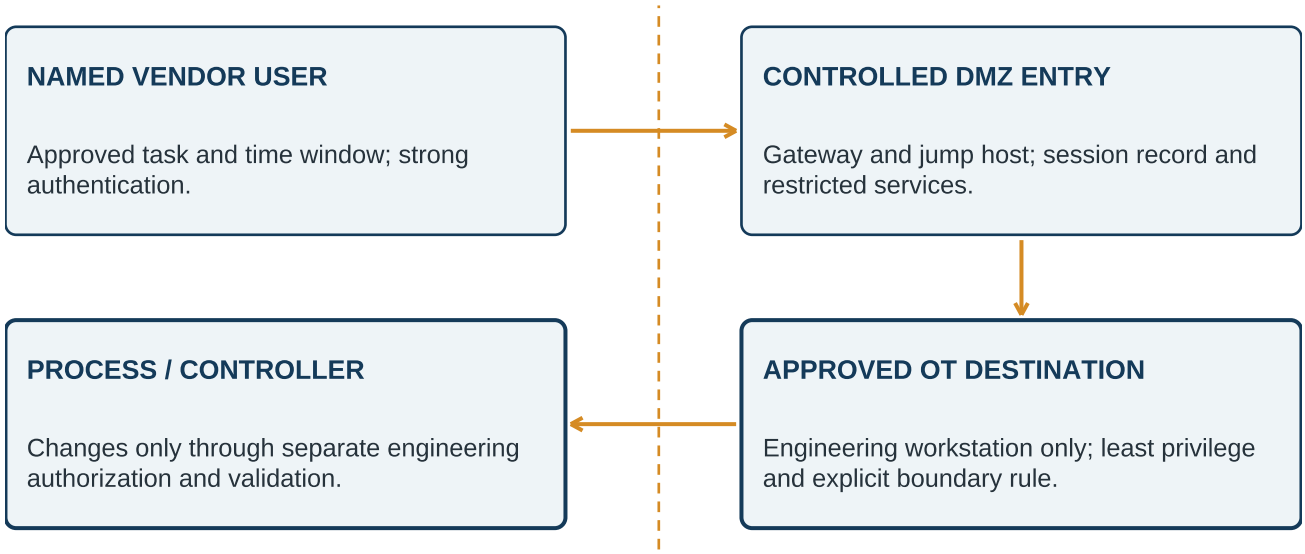
NIST SP 800-82 Revision 3, Figure 18, illustrates defense in depth across enterprise, DMZ, operations-management and field networks. The source figure below is a reference architecture, not an actual customer installation or proof of a secure configuration. Its boundaries help identify where remote sessions, credentials and control-capable communications need engineering scrutiny.



NIST SP 800-82 Rev. 3, Figure 18, printed p. 85 (PDF p. 102): DCS defense-in-depth reference architecture. Enterprise, DMZ, operations-management and field levels are distinct.

Open the full NIST figure

Enlarged engineering interpretation: an authorized maintenance session



No direct vendor-to-controller route. Ordinary process control does not depend on an open remote session.

Defined trust boundary

BuilderCE explanatory model of a bounded maintenance path. It simplifies the reference architecture and adds explicit approval controls; it is not a tracing of every connection in Figure 18.

BOUNDARY / CONTROL	EVIDENCE THAT THE PATH IS BOUNDED
Vendor → controlled entry	Named identity, strong authentication and approved time window; no exposed direct remote desktop on the controller.
DMZ → approved OT workstation	Only the approved destination and required service; session logging and restricted privileges. A permitted protocol is not proof that all commands are safe.
Workstation → controller	Separate engineering change authorization, baseline comparison and process-specific safeguards before any write-capable action.
Session closure	Disable the temporary entitlement, terminate the session and retain the task, time and destination record. Verify that local process control remains available.

The model intentionally separates identity approval, network reachability and permission to change the process. A login can be valid while the requested action is unauthorized. Multi-factor authentication does not validate controller logic, and a jump host does not eliminate the need to protect the workstation and verify boundary rules. Test actual permitted and denied communications within an approved, operations-compatible procedure.

Lifecycle evidence. Request, named approval, controlled connection, task-specific action, monitored completion and verified termination form one access record. NIST SP 1800-45 offers three representative water-sector remote-access implementations; the appropriate design depends on the utility's capacity and operational needs.

MODULE 4 | 25 MINUTES

Legacy Systems, Vendors & Operational Resilience

4.1 Legacy and Unsupported Systems

Legacy OT is common because physical assets often outlive the software platforms that control them. Unsupported operating systems, obsolete controllers, unavailable firmware, proprietary communications, and vendor-discontinued equipment can remain operationally important. The risk is not simply age. The concern is whether the organization understands the exposure, can reduce it, can detect abnormal behavior, and has a plan for failure or compromise.

A structured assessment begins with function and consequence. What does the system control? Is it connected to less-trusted networks? Does it require remote support? Are known vulnerabilities relevant to the actual architecture? Can unused services be disabled? Can access be restricted? Can the system be isolated behind a controlled gateway? Are replacement parts available? Is there a tested backup of logic or configuration?

Compensating controls are appropriate when direct remediation is not feasible. Examples include stronger segmentation, removing internet access, restricting management interfaces, monitoring communications, limiting users, application allowlisting where supported, physical access control, disabling unnecessary services, and maintaining tested spares or recovery images. These controls should be documented as intentional risk treatments, not informal workarounds.

The long-term plan matters. A system that cannot be patched forever should have a replacement, upgrade, isolation, or retirement strategy tied to capital planning. Without a lifecycle plan, temporary compensating controls become permanent by default.

LEGACY ASSET	DOCUMENTED TREATMENT
Constraint	Unsupported product, unavailable patch or operational compatibility limit.
Exposure / consequence	Physical function and actual reachable pathways.
Compensating control	Segmentation, restricted services, monitoring, recoverability and defined exceptions.
Lifecycle plan	Owner, funded migration or replacement strategy, due date and review trigger.

4.2 Vendor and Third-Party Risk

OT systems often depend on vendors for proprietary software, commissioning, firmware, diagnostic tools, and specialized maintenance. That dependence can create hidden cybersecurity pathways. Vendors may use remote access, shared service accounts, unmanaged laptops, cloud portals, or subcontractors. Engineers who write specifications or manage service contracts can influence these risks before the system is installed.

A vendor-access process should define identity, approval, connection method, destination, privilege level, duration, monitoring, and termination. Shared persistent accounts should be avoided where technically feasible. If a vendor requires a remote appliance or cloud service, the owner should understand what information leaves the site, whether commands can be issued remotely, how updates are delivered, and how the service behaves if the vendor platform is unavailable.

Third-party risk also includes software and firmware supply chain. Engineers may not be able to evaluate source code, but they can require support commitments, vulnerability-notification processes, secure update methods, documented end-of-life dates, and clear responsibility for cybersecurity maintenance.

The key principle is that outsourced maintenance does not outsource consequence. The asset owner remains responsible for safe and reliable operation. Vendor arrangements should therefore be governed as part of the system architecture and lifecycle.

4.3 Procurement and Cybersecurity by Design

Procurement is one of the highest-leverage points in OT cybersecurity because requirements can be established before technical lock-in. A low bid that lacks named accounts, secure remote-access options, logging, backup capability, supported update mechanisms, or reasonable product lifecycle may create a much higher total risk and cost over the asset life.

Engineering specifications can address cybersecurity at an appropriate level. Requirements may include supported authentication methods, role-based access, disabling unused services, secure configuration defaults, event logging, backup and restore capability, documentation of ports and protocols, remote-access architecture, vulnerability notification, support duration, update procedures, and cooperation with owner security requirements. The exact requirements should match the system's criticality and organization policy.

Cybersecurity requirements should be testable. A specification that says “provide a secure system” gives little acceptance basis. A better approach identifies observable outcomes: default credentials changed, unnecessary services disabled, remote access routed through the approved platform, backups demonstrated, named privileged accounts established, required logs available, and network communications documented.

Commissioning should verify the installed state, not only the design intent. Temporary vendor accounts, test wireless networks, open switch ports, bypass rules, and commissioning laptops should be removed or converted to approved permanent controls before turnover.

PROCUREMENT FIELD	TESTABLE ACCEPTANCE
Connectivity / identity	Approved paths, accounts, authentication and least privilege.
Lifecycle / supplier	Support period, update provenance, notifications and access termination.
Recovery / verification	Configuration backups, restoration prerequisites and witnessed functional evidence.

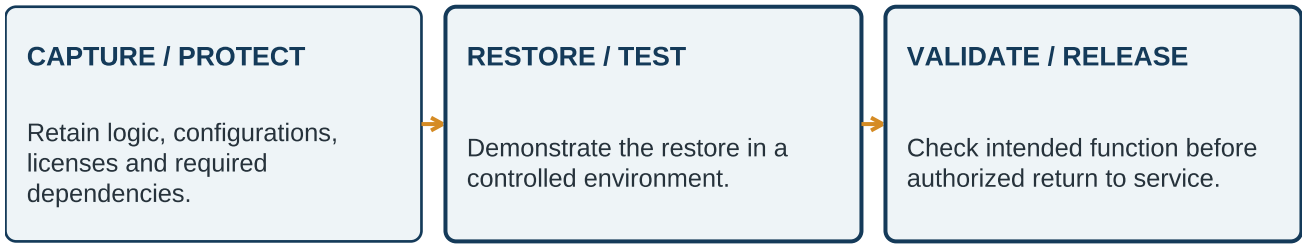
4.4 Backup, Recovery, and Operational Resilience

Backups are a resilience control, but the term “backup” can hide important gaps. OT recovery may require controller logic, HMI applications, historian databases, network configurations, security-device configurations, license files, encryption keys, recipes, calibration data, virtual-machine images, and vendor installation media. If only the server data is backed up, the organization may still be unable to rebuild the system.

Recovery planning should start with the process. Which functions must be restored first? Which can operate manually? What dependencies must be available before a controller or server is useful? How long can the process tolerate degraded operation? These questions define recovery time objectives in operational terms rather than purely IT terms.

Copies should be protected from the same event that affects production. A ransomware incident can encrypt online backups if they are continuously reachable from compromised accounts. Offline, isolated, immutable, or otherwise separately protected copies may be needed for critical systems. Physical spares and known-good configurations can be equally important for specialized equipment.

Testing is the difference between having backups and having recovery capability. A periodic restore test can reveal missing software, undocumented passwords, incompatible firmware, broken licenses, or staff knowledge gaps before an emergency. The test should include engineering validation that the restored system actually performs its intended function.



A backup becomes recovery capability only when restoration and intended function are demonstrated.

4.5 Resilience, Manual Operations, and Safe Degraded States

Operational resilience is the ability to continue or restore essential functions despite disruption. Cybersecurity supports resilience when the system can fail in controlled ways, isolate affected components, continue local control when higher-level systems are unavailable, and recover from known-good states. The appropriate design depends on the process.

Manual operation can be a powerful resilience measure, but it should not be assumed. Some processes can operate locally for hours with supervisory systems unavailable. Others depend on coordinated automation, remote telemetry, or complex sequencing that operators cannot reasonably reproduce manually. The engineering team should identify realistic degraded modes and ensure staff understand them.

Safe degraded operation should be documented and exercised where consequence justifies it. If a remote connection is disabled, can the process continue? If the HMI is unavailable, can operators see critical indications locally? If the historian is offline, what decision-making capability is lost? If the engineering workstation is quarantined, is there a clean spare or recovery image?

Resilience decisions also affect capital planning. A system with no spare hardware, no tested backup, and no manual fallback may require investment even if no current vulnerability is known. Cybersecurity provides another lens through which reliability and lifecycle risk are evaluated.

4.6 Supply Chain, Firmware, and Update Provenance

Modern OT products contain complex supply chains: embedded operating systems, third-party libraries, firmware, cloud components, drivers, communications modules, and vendor service tools. Engineers may not be able to inspect every software component, but they can establish expectations for how products are supported and updated.

Update provenance means knowing that firmware or software came from an authorized source and has not been altered. Vendors should provide documented update methods, integrity validation where supported, release notes, compatibility information, and a vulnerability-notification process. Organizations should avoid downloading critical firmware from unverified sources or using files passed informally between contractors without traceability.

Supply-chain risk also includes counterfeit or substituted hardware, unmanaged subcontractors, compromised service tools, and loss of vendor support. Procurement and quality processes should therefore coordinate. A replacement controller that is electrically compatible may still introduce a different firmware baseline or unsupported software dependency.

Engineers should maintain records of installed versions and approved update packages for critical assets. When a vulnerability is announced, that inventory allows the team to determine exposure quickly. Without version data, the organization may spend days discovering whether a bulletin even applies.

Long-term support commitments are especially important for assets expected to operate for decades. The organization should understand how the vendor communicates end of support, whether migration

paths exist, how licenses are managed, and what happens if the vendor or cloud service is no longer available.

4.7 Recovery Engineering: Spares, Golden Configurations, and Restoration Order

Recovery engineering extends beyond backups. Critical OT systems may require spare controllers, compatible firmware, programming cables, license files, known-good configuration images, replacement network devices, or vendor-specific tools. A digital backup is not useful if there is no compatible hardware on which to restore it.

A golden configuration is an approved, known-good state for a device or system. It may include controller logic, HMI application, firewall rules, switch configuration, server image, and documented software versions. The organization should protect these baselines from unauthorized modification and periodically confirm that the installed state remains consistent with approved changes.

Restoration order matters because dependencies are layered. Network infrastructure may need to be restored before servers. Identity or name services may be needed before applications. Controllers may require local verification before supervisory systems are connected. External vendor or cloud services may need to be available for licenses or diagnostics. A recovery runbook should reflect the actual dependency chain.

Spares should be maintained as assets, not forgotten on a shelf. Firmware compatibility, batteries, storage conditions, and configuration should be considered. If a spare must be updated before use, the team should know how that will be done during an outage.

Recovery exercises can be scaled. A full plant shutdown is rarely necessary to test readiness. Teams can restore a controller project to a spare device in a lab, rebuild a virtual server, recover a firewall configuration, or walk through a restoration sequence. The goal is to discover missing dependencies before the real event.

4.8 Vendor Lifecycle: From Submittal to Service Termination

Vendor cyber risk changes across the asset lifecycle. During procurement, the owner evaluates product capability and support commitments. During submittal and design, the team confirms architecture, ports, protocols, remote access, and account requirements. During commissioning, temporary vendor privileges may be broader than the final operating model. During warranty and service, remote diagnostics and software updates become routine. At end of support, the owner must decide whether to upgrade, isolate, or replace the system.

Each transition can leave residual access or uncertainty. Commissioning accounts may remain active. Temporary firewall rules may become permanent. A vendor laptop certificate may never be revoked. A subcontractor may retain credentials after the prime service contract ends. Cloud portals may continue collecting data even after local service is terminated.

A lifecycle checklist should therefore include access review at turnover, warranty transition, personnel changes, service-provider changes, major upgrades, and contract closeout. The system owner should know which external organizations can connect and who internally approves that connection.

Contract language can support this discipline. Service agreements can require compliance with owner access procedures, named users where feasible, prompt notification of security incidents affecting the product or service, controlled update methods, cooperation during incident response, return or destruction of owner credentials, and termination of access at the end of the engagement. Exact terms should be reviewed by qualified contracting and legal personnel.

The engineering takeaway is that vendor access is not a one-time technical configuration. It is a relationship that changes over years. Without lifecycle ownership, access tends to accumulate while accountability becomes less clear.

4.9 Continuity Planning for Cyber-Dependent Engineering Services

Operational continuity planning asks how essential engineering services continue when digital support is degraded. The plan should not assume the only choices are normal operation or complete shutdown. Many systems have intermediate states: local control without supervisory visibility, manual sampling instead of online analyzers, field dispatch instead of remote telemetry, or limited operation at reduced capacity.

Engineers can define these states because they understand process constraints. For each critical system, the team can identify minimum information needed to operate safely, maximum tolerable duration of degraded operation, staffing requirements, manual tasks, spare equipment, external dependencies, and triggers for controlled shutdown. These facts support both cyber incident planning and ordinary disaster recovery.

Communications are part of continuity. If enterprise email, directory services, or voice-over-IP systems are affected, can operators reach vendors and leadership? Are emergency contacts available offline? Are current network diagrams and recovery procedures stored somewhere accessible if the document-management system is unavailable? Can credentials needed for emergency recovery be obtained securely?

The organization should also identify decision thresholds. A process may continue locally for four hours without historian data but not for several days. A remote station may operate autonomously until a tank level reaches a defined threshold. A building may tolerate loss of central BAS supervision if local controllers remain stable but require increased rounds by facilities staff. These thresholds convert continuity planning into actionable engineering guidance.

Exercises should test more than technology restoration. Ask staff to operate using the degraded-mode plan, locate offline procedures, contact the vendor without ordinary systems, and determine which process functions receive priority. Gaps found in these exercises often reveal inexpensive improvements such as updated printed contact lists, local indication, spare communication methods, or clearer shutdown criteria.

Cyber resilience becomes strongest when continuity measures are integrated into normal operations. A manual fallback that no one practices, a spare that has not been powered in years, or an offline procedure that references obsolete equipment cannot be credited as dependable resilience.

Applied Scenario 4 - Legacy PLC with No Patch Path

A manufacturing line depends on a PLC family that is no longer supported. The controller cannot use modern authentication and the programming software requires an obsolete operating system. Replacing the line immediately would be economically disruptive. The organization decides not to treat “unsupported” as either an automatic shutdown or an excuse for inaction.

The engineering and cybersecurity teams document the controller's function, isolate the control network from ordinary business traffic, restrict programming access to a dedicated engineering workstation, remove unnecessary remote access, maintain a tested spare and verified logic backup, monitor the limited communications path, and develop a funded replacement plan. The risk remains, but it is visible, bounded, and managed.

The lesson is that cybersecurity engineering often involves managing imperfect systems responsibly over time rather than achieving an ideal architecture overnight.

MODULE 5 | 20 MINUTES

Incident Response, Escalation & Applied Engineering Scenarios

5.1 Recognizing Abnormal Cyber-Physical Behavior

An OT cyber incident may first appear as an operational anomaly rather than a security alert. Operators may see a setpoint change without explanation, a controller restart unexpectedly, an HMI show values that do not match local instruments, a remote session outside a maintenance window, repeated authentication failures, unexplained communications loss, or a device that behaves differently after a software update.

Engineers should know which abnormalities deserve cybersecurity escalation. The threshold should not require proof of malicious activity. Early escalation allows specialists to investigate while engineering and operations assess process safety and continuity. A false alarm is usually less costly than allowing a real compromise to continue unnoticed in a critical system.

The first operational question is whether the process is safe and stable. Staff may need to verify critical values through independent indications, place equipment in a known state, restrict remote control, or increase local observation. Cybersecurity personnel may simultaneously preserve evidence, analyze logs, and identify affected accounts or devices.

Training should emphasize that staff should not improvise destructive “cleanup” actions on critical systems. Power cycling, wiping logs, reinstalling software, or disconnecting equipment without coordination can destroy evidence or create process hazards. The response plan should connect cybersecurity actions to engineering consequences.

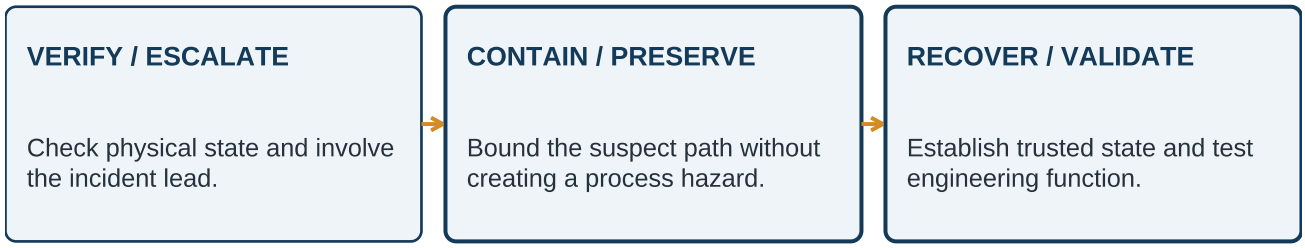
5.2 Incident Escalation and Decision Rights

Incident response requires clear decision rights because containment actions can affect operations. Cybersecurity may recommend isolating a network segment. Operations may be concerned about loss of visibility. Engineering may need to determine whether local control remains stable. Leadership may need to decide whether a service interruption is acceptable. Those roles should be understood before an incident.

An escalation matrix can identify who must be contacted for safety impact, loss of critical control, suspected unauthorized logic change, vendor-account compromise, or widespread network disruption. Contact information should be maintained outside systems that may become unavailable during the event.

The team should distinguish technical authority from business authority. An engineer may determine that a control action is technically necessary to protect equipment, while a senior leader authorizes a service shutdown or external notification. Legal, regulatory, communications, and insurance obligations may also apply depending on the event. This course does not attempt to prescribe those obligations; it emphasizes the need to integrate them into the organization's plan.

During response, decisions and assumptions should be recorded. A short contemporaneous log of who decided what, when, and why supports continuity across shifts and later review.



Containment is coordinated with operations; safe engineered function must be established before normal service resumes.

5.3 Containment Without Creating a Process Hazard

Containment seeks to limit the spread or effect of a cyber incident, but OT containment must be coordinated with the physical process. Disconnecting an affected HMI may be low risk if local control continues. Isolating a controller network may be much more consequential. Blocking a vendor connection may remove a compromise path but also eliminate the vendor's ability to support recovery.

The response team should identify the minimum action that reduces cyber risk while preserving a safe state. Options can include disabling a specific account, terminating a remote session, blocking a communications path, isolating a workstation, moving a system to local control, restricting write access, or shutting down part of a process under controlled conditions.

Preplanned containment strategies are valuable. If the team has already documented network boundaries, critical dependencies, manual operating capability, and emergency access procedures, containment can be more precise. Without that preparation, organizations may choose between overly broad shutdowns and risky continued operation.

Engineers should validate the process after containment. A network rule that appears successful from a security console may have unintended effects on alarms, time synchronization, historian data, interlocks, or supervisory functions. Cyber containment is not complete until the physical system is understood to be in an acceptable state.

5.4 Recovery, Validation, and Return to Service

Recovery should restore the system to a known, trusted condition rather than simply make the screen turn green again. The team may need to restore configurations, replace devices, reset credentials, rebuild workstations, verify controller logic, validate communications, and confirm that the vulnerability or pathway used in the incident has been addressed.

Engineering validation is essential. A restored system should be checked against approved configuration and functional requirements. Alarms, interlocks, failover, control loops, remote communications, and data interfaces should be tested at a level appropriate to the change and consequence. If temporary compensating controls are used, they should be documented with an owner and expiration or review date.

Return-to-service decisions should consider residual uncertainty. It may be reasonable to operate with increased monitoring while a full investigation continues, or it may be necessary to remain offline until confidence is higher. The decision depends on safety, mission, evidence, available controls, and organizational risk tolerance.

Recovery also includes restoring the governance environment: disabling emergency accounts, removing temporary network rules, terminating vendor access, updating asset records, and confirming backups reflect the approved post-recovery state.

5.5 Post-Incident Learning and Engineering Change

A cyber incident should produce more than an IT ticket closure. The post-incident review should identify how the event entered the environment, why it was able to affect the system, which controls succeeded or failed, whether the architecture increased consequence, whether monitoring was adequate, and what engineering changes are warranted.

Corrective actions may involve technology, procedures, procurement, training, vendor contracts, network architecture, spare strategy, backups, or capital replacement. The organization should distinguish immediate fixes from systemic improvements. Resetting one password does not solve a pattern of shared accounts. Closing one remote port does not address a culture of unmanaged vendor connectivity.

Lessons learned should feed design standards and project specifications. If an incident exposed poor asset visibility, future projects should strengthen turnover requirements. If recovery failed because configuration backups were incomplete, commissioning procedures should require restore testing. If a vendor account remained active after service ended, contract and access-termination processes should change.

This closes the lifecycle: cybersecurity risk is identified, controlled, monitored, tested in operation, and improved through experience. Engineers participate throughout because the objective is not only secure information - it is safe, reliable, resilient engineered performance.

5.6 Tabletop Exercises and Cross-Functional Preparedness

A tabletop exercise allows engineering, operations, IT, cybersecurity, leadership, and vendors to practice decisions without disrupting the process. The scenario should be realistic and tied to actual system architecture. Examples include suspicious remote access, loss of HMI visibility, ransomware on an engineering workstation, compromised vendor credentials, or discovery that backups cannot be restored.

The exercise should test decisions rather than trivia. Who notices the event? Who is called? Who can disable remote access? Who determines whether the process can remain online? Where are network diagrams and contact lists stored? How are vendors reached after hours? Who decides to shut down? What evidence must be preserved? How are regulators, customers, or leadership informed if required?

Observers should record gaps in roles, information, tools, and procedures. A successful exercise is not one in which participants already know every answer. It is one that reveals ambiguity while the organization still has time to correct it.

Engineers should pay particular attention to physical assumptions. A cyber team may assume a system can be isolated without consequence. Operations may assume a manual fallback exists that has not been practiced in years. A vendor may be the only party with the software needed to restore a controller. These are the kinds of dependencies tabletop exercises expose.

Corrective actions should have owners and due dates. Repeating the same exercise a year later without closing the identified gaps is not preparedness. The exercise becomes valuable when lessons lead to updated architecture, contacts, backups, procedures, procurement requirements, or training.

5.7 Evidence Preservation, Forensics, and the Engineer's Role

Cyber investigations may require logs, disk images, network records, controller project files, account data, and other technical evidence. Engineers are not expected to become forensic examiners, but they should understand that ordinary troubleshooting actions can alter evidence. Rebooting a workstation, clearing logs, restoring a controller, or replacing a device may be operationally necessary, yet those actions should be coordinated when possible so critical evidence is preserved first.

The priority remains safety and process control. If an action is required immediately to prevent harm, the organization should act. When time permits, the response team can photograph screens, export logs, record timestamps, capture configuration hashes or copies, note physical switch positions, and document the state before changes are made.

Engineering records can be highly valuable in determining whether a change was authorized. Approved logic, configuration baselines, commissioning records, change tickets, and vendor service reports provide comparison points. Good configuration management therefore supports both quality and cybersecurity investigations.

Chain-of-custody and formal forensic procedures may become important in significant incidents, legal matters, or regulatory investigations. Those processes should be directed by qualified cybersecurity, legal, and investigative personnel. The engineer's role is to preserve relevant engineering context, explain process consequences, and avoid unintentionally destroying data that others may need.

After the investigation, the organization should decide which evidence and lessons belong in permanent engineering records. Temporary incident artifacts are not necessarily part of the normal design record, but changes made to architecture, configuration, or operating procedures should be controlled and documented.

5.8 Scenario Set: Three Different OT Incidents, Three Different Responses

Scenario A: A wastewater facility loses its supervisory HMI, but local PLCs continue controlling pumps and valves. Operators still have local indications and the process is stable. The response may prioritize preserving local control, isolating the affected supervisory server, increasing field observation, and restoring visibility from a known-good state. An immediate plant shutdown may be unnecessary if the engineered degraded mode is understood and safe.

Scenario B: A substation operator sees a remote switching command that no one authorized. Here the integrity of control authority is in question. The response threshold is higher because an unauthorized command can directly affect energized equipment and public service. The team may need to restrict remote-control pathways, verify device states independently, coordinate with protection and operations personnel, and preserve event records before making configuration changes.

Scenario C: A building automation vendor reports that its cloud management platform may have been compromised. The local building continues operating normally. The owner must determine whether the cloud service can issue commands, what credentials or tokens are exposed, whether the connection can be disabled without losing local control, and whether any configuration changes occurred. The appropriate response depends on the architecture, not the headline that a vendor had a cyber incident.

These scenarios demonstrate why OT response cannot be a single playbook action such as “disconnect everything.” The physical function, architecture, local autonomy, consequence, and evidence all matter. A mature response plan provides common escalation and documentation while allowing engineering judgment to tailor containment.

The same principle applies to recovery. In Scenario A, restoring the HMI may be sufficient. In Scenario B, the team may need to validate relay or control logic and credentials. In Scenario C, the organization may need to rotate tokens, review cloud integrations, and confirm local configuration. The cyber event category is only the starting point; engineering consequence determines the response depth.

5.9 Communication, External Coordination, and Decision Discipline

Significant OT cyber incidents can require coordination beyond the control room. Depending on the organization and event, participants may include executive leadership, legal counsel, regulators, emergency management, law enforcement, customers, insurers, equipment vendors, managed-service providers, and public communications personnel. Engineers should not improvise those external obligations, but they should be prepared to provide accurate technical facts about system function, operational impact, and recovery status.

Technical communication should distinguish confirmed facts from hypotheses. Early in an incident, the team may know that an unauthorized remote session occurred and that one engineering workstation is suspect, but not whether controller logic changed. Saying “the control system was hacked” can create unnecessary confusion if the evidence is incomplete. A more disciplined statement identifies what is known, what is being investigated, and what operational safeguards are in place.

Engineers can help leadership understand consequence in practical terms. Instead of reporting only that "SCADA is unavailable," explain whether local PLC control continues, whether operators have independent indications, what functions are lost, how long degraded operation is acceptable, and what triggers a shutdown or restoration milestone. This information supports better decisions about continuity, public messaging, and resource allocation.

Vendor coordination should also be controlled. During an incident, a well-intentioned vendor may want immediate remote access or may recommend reinstalling software before evidence is preserved. The incident lead should coordinate access, define the task, and document actions. If external responders connect to OT, their access should use the same principles of identity, least privilege, monitoring, and termination discussed elsewhere in the course.

Decision discipline becomes especially important under time pressure. Teams should avoid undocumented emergency bypasses that remain after recovery, broad credential sharing, or uncontrolled parallel troubleshooting by multiple parties. A short incident decision log can record the problem, action, responsible person, time, and rationale. That record supports shift handoff, later investigation, and restoration of normal controls.

Communication quality is therefore part of resilience. Accurate, bounded technical information reduces panic, prevents conflicting actions, and helps the organization maintain a common operating picture while cybersecurity and engineering work proceed together.

Final Applied Scenario - Compromised Engineering Workstation

A utility engineer notices that a controller configuration timestamp changed overnight even though no maintenance was scheduled. The engineering workstation used for programming shows no obvious alarm, but the remote-access gateway log indicates a vendor account connected outside the approved window. Operators report that the process is stable.

A weak response would be to dismiss the issue because the plant is still running or immediately power-cycle devices to "clear" the problem. A disciplined response verifies process safety, escalates to the defined incident team, preserves logs and configuration evidence, terminates or restricts the suspicious access path, compares installed logic with an approved baseline, and determines whether other systems share the same credentials or workstation trust.

Recovery may require rebuilding the engineering workstation, resetting accounts, validating controller logic, reviewing vendor-access controls, and testing backups. The post-incident review asks why the account was usable outside the maintenance window, whether session monitoring was adequate, and whether the workstation had unnecessary trust to multiple systems.

The scenario integrates the course themes: know the assets, understand architecture, control trust boundaries, manage identities and remote access, preserve recoverability, recognize abnormal behavior, and coordinate cyber actions with engineering judgment.

Completed illustrative access and incident record: OT-17

This hypothetical water-facility record develops the course's compromised-workstation scenario. The plant has a documented safe local operating mode and independently available process indications. Actual containment and test actions must be chosen by qualified operations, engineering and incident-response personnel for the installed process. The times and asset names below are illustrative, not facts from an actual incident.

TIME / OWNER	OBSERVATION, DECISION AND EVIDENCE
09:00 · operator	Finds vendor account V-17 active outside its 08:00–08:30 diagnostic window. Process is stable on independent indications. An unexpected controller configuration timestamp is also reported. Log L-17 is preserved.
09:05 · incident lead + operations	Declare a suspected incident. Record confirmed session evidence separately from the unverified possibility of a logic change. Operations confirms that blocking remote maintenance will not interrupt local control.
09:10 · authorized network administrator	Terminate V-17's remote session and disable its temporary entitlement at the gateway. Keep controller power and needed local communications in service. Record rule change R-17 and monitor for reconnection.
09:20 · response specialist	Preserve available workstation and gateway evidence under the incident procedure, with source, collector, time, clock offset and integrity records. Avoid uncoordinated rebooting or scanning.
10:00 · controls engineer	Compare the controller's logic and parameters with approved baseline B-17 using controlled tools. No logic difference is found in this illustrative comparison; the timestamp alone is not treated as proof of altered logic. Investigate other reachable assets.
11:30 · response + controls team	Rebuild the suspect workstation from a trusted image, rotate affected credentials and restore approved engineering configuration. Retain evidence before remediation where safety and the response plan permit.
13:00 · operations + engineering authority	Complete the approved validation plan, record results V-17, and authorize normal maintenance capability only through a new bounded access approval. Keep enhanced monitoring and an assigned corrective action.

Return-to-service record V-17

VALIDATION FIELD	COMPLETED ILLUSTRATIVE RESULT
Trusted configuration	Workstation image, approved tools, controller logic and parameters reconciled to retained baselines; affected credentials replaced.
Communications / access	Required operational flows pass. Unapproved vendor path is denied. New temporary access requires named approval and expiry.
Physical function	Operations and engineering verify process indications, alarms, interlocks, control-loop behavior and applicable failover under a process-specific test plan. Tests are not improvised on a live hazardous process.
Recovery evidence	Protected backup restored in the controlled test environment; required licenses and configuration dependencies available. Restoration evidence is distinct from merely possessing backup files.
Decision / follow-up	Operations manager and engineering authority record release at 13:00. Access owner must correct the failed expiry rule before the next vendor session; the incident lead checks implementation and evidence within seven days.

The completed record supports a bounded conclusion: the preserved logs establish an out-of-window access event; the specified baseline comparison found no logic difference in the assets tested. It does not establish that no compromise occurred anywhere. The incident team retains the remaining

investigation scope and documents uncertainty. Successful process testing likewise supports release of the tested functions, not a universal declaration that every device is secure.

> **Verified closure.** The failed access-expiry control is corrected, a test session expires at its approved end time, a subsequent connection is denied, and the access owner records the evidence. The incident lead confirms closure rather than accepting a policy update alone.

Engineer's OT Cybersecurity Checklist

- Know which OT assets are critical and who owns them.
- Keep system architecture and approved communications paths current.
- Control remote and vendor access with named identities and defined lifecycle.
- Segment high-consequence systems from less-trusted environments.
- Use a risk-based process for patches, configuration changes, and legacy-system compensating controls.
- Protect engineering workstations, logic/configuration files, and portable media pathways.
- Require cybersecurity capabilities and lifecycle support during procurement.
- Maintain protected backups and periodically demonstrate restoration.
- Define abnormal conditions that require cybersecurity escalation.
- Coordinate containment with process safety and operational continuity.
- Validate the engineered function before returning recovered systems to normal service.
- Feed lessons learned into design standards, specifications, access procedures, and lifecycle plans.

> **Practice principle.** Cybersecurity is most effective when it is integrated into engineering architecture, operations, maintenance, procurement, and recovery rather than added as a separate checklist after the system is already built.

Principal references and source links

- NIST Cybersecurity Framework 2.0, NIST CSWP 29, February 2024. Open source
- NIST SP 800-82 Rev. 3, Guide to Operational Technology (OT) Security, September 2023; the final edition used for this course. Open source
- NIST SP 1339, OT Backup Quick Start Guide, June 2026. Open source
- NIST SP 1800-45, Cybersecurity for the Water and Wastewater Sector: Build Architecture (Operational Technology Remote Access), June 2026. Open source
- CISA and partners, Foundations for OT Cybersecurity: Asset Inventory Guidance for Owners and Operators, 2025. Open source
- ISA/IEC 62443 series: risk assessment, system partitioning, zones and conduits. Public overview; installed-system and sector requirements determine application. Open source

NIST's publication record was checked November 15, 2026. Revision 3 remains the final SP 800-82 edition used here. The September 2026 successor consultation is not a final replacement. Organization-specific policies, sector requirements, vendor documentation and qualified engineering/cybersecurity judgment control implementation.

Sources checked November 15, 2026. Hypothetical records and thresholds are identified within their examples. Official source images retain their provenance and limitations.

Instructor

Matthew Ward, MBA, CIA - background in business, internal audit, governance, risk, controls, compliance, and professional education. Course scope is limited to cybersecurity risk management, governance, control, resilience, documentation, and technology-risk concepts as applied to engineering organizations and operational technology. Discipline-specific OT design and specialized cybersecurity engineering should be performed or reviewed by appropriately qualified professionals.

> **Study guide complete.** Return to the BuilderCE lesson page and select Continue to begin the final examination. A score of at least 70% is required to pass. Retakes are permitted.