Certified Specialist Programme in Medical Device Cybersecurity

Introduction to Medical Device Cybersecurity

Medical device cybersecurity is a multidisciplinary field that combines principles of engineering, information technology, regulatory affairs, and clinical practice to protect devices that diagnose, treat, or monitor patients. The following key terms and vocabulary form the foundation for anyone studying the Certified Specialist Programme in Medical Device Cybersecurity. Understanding these concepts is essential for performing risk assessments, designing secure systems, and complying with international standards.

Asset – any component of a medical device that has value and can be targeted by a cyber-attack. Assets include hardware such as sensors, processors, and power supplies; software modules, firmware, and operating systems; data such as patient records, device logs, and configuration settings; as well as network interfaces and cloud services. For example, the firmware that controls the infusion pump's motor is an asset because manipulation could cause an overdose.

Threat – a potential cause of an unwanted incident that may result in damage to an asset. Threats can be intentional, such as a hacker attempting to gain unauthorized access, or unintentional, such as a software bug that leads to a crash. In the context of a cardiac monitor, a threat could be a malicious actor trying to alter heart-rate measurements to hide a patient's arrhythmia.

Vulnerability – a weakness in a device's design, implementation, or operation that could be exploited by a threat. Vulnerabilities may arise from insecure coding practices, default passwords, unpatched operating systems, or inadequate physical protection. A classic example is the use of hard-coded credentials in a device's web server, which allows an attacker to log in without authentication.

Risk – the combination of the likelihood that a threat will exploit a vulnerability and the impact of that exploitation on patients, data, and the organization. Risk is expressed numerically in many risk-assessment frameworks, often using a matrix that multiplies severity by probability. For a ventilator, a high-risk scenario might be a remote code execution vulnerability that could cause the device to stop breathing support, resulting in severe patient harm.

Risk assessment – the systematic process of identifying assets, threats, and vulnerabilities; estimating likelihood and impact; and determining risk levels. The assessment drives mitigation strategies and informs regulatory submissions. In practice, a risk assessment for an insulin pump would list the pump's wireless communication module as an asset, identify "unauthorized remote command injection" as a threat, note the lack of encrypted traffic as a vulnerability, and calculate the resulting risk.

Threat actor – the individual, group, or entity that initiates a cyber-attack. Threat actors can be nation-states, organized crime groups, hacktivists, disgruntled employees, or curious researchers.

Understanding the motivations and capabilities of each actor helps prioritize defenses. A nation-state might target high-value hospital infrastructure for espionage, while a hacktivist may seek to expose perceived security failures.

Attack surface – the sum of all points where an unauthorized user could attempt to enter or influence a device. This includes physical ports, wireless interfaces (Bluetooth, Wi-Fi, NFC), software APIs, web services, and maintenance ports. Reducing the attack surface is a core design principle; for example, disabling unused USB ports on a bedside monitor eliminates a potential entry vector.

Threat vector – the path or method used by a threat actor to exploit a vulnerability. Common vectors for medical devices include network protocols (TCP/IP, HTTP), wireless protocols (BLE, Zigbee), removable media (USB drives), and social engineering (phishing emails). An attacker might use a phishing email to deliver ransomware that encrypts imaging data stored on a PACS server.

Malware – malicious software designed to disrupt, damage, or gain unauthorized access to a system. Types of malware relevant to medical devices include viruses, worms, trojans, ransomware, and spyware. Ransomware that encrypts a hospital's electronic health record (EHR) system is a well-known example that can indirectly affect device operation by delaying critical patient data.

Ransomware – a subset of malware that encrypts files or locks systems until a ransom is paid. In a medical context, ransomware can target the device itself (e.G., Locking a dialysis machine) or supporting infrastructure (e.G., The network that delivers software updates). The 2020 attack on a major U.S. Hospital network demonstrated how ransomware can force clinicians to revert to manual processes, increasing the risk of errors.

Phishing – a social-engineering technique that attempts to trick users into revealing credentials or downloading malicious content. Phishing emails often masquerade as legitimate communications from vendors or IT departments. A common scenario involves a technician receiving an email that appears to be from the device manufacturer, prompting the download of a fake firmware update that contains a trojan.

Insider threat – a risk posed by individuals with authorized access who misuse that access, either intentionally or unintentionally. Insiders may be employees, contractors, or third-party service providers. An insider could deliberately alter device settings to sabotage a clinical trial, or unintentionally expose patient data by using weak passwords.

Supply chain – the network of organizations, processes, and resources involved in creating and delivering a medical device. The supply chain includes component manufacturers, software developers, integrators, distributors, and service providers. Compromise of any link in the supply chain can introduce vulnerabilities; for example, a compromised microcontroller firmware supplied by a third-party vendor could contain hidden backdoors.

Lifecycle – the series of phases a medical device undergoes from concept through design, production,

deployment, maintenance, and retirement. Cybersecurity considerations must be integrated throughout the entire lifecycle to ensure continuous protection. Early-stage threat modeling, mid-stage secure coding, and post-market vulnerability management are all parts of the lifecycle approach.

Secure development lifecycle (SDL) – a framework that embeds security activities into each stage of product development. SDL activities include threat modeling, secure coding standards, static analysis, penetration testing, and incident response planning. Applying SDL to a pulse oximeter means performing code reviews during firmware development and conducting penetration tests before the device is released.

Threat modeling – a structured approach to identifying, enumerating, and prioritizing threats based on the system's architecture and intended use. Common methodologies include STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege) and PASTA (Process for Attack Simulation and Threat Analysis). For a wireless infusion pump, threat modeling would examine how an attacker could spoof the pump's controller to deliver an unauthorized dose.

STRIDE – an acronym that categorizes threats: Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service, Elevation of privilege. Using STRIDE helps analysts systematically explore security gaps. In a remote monitoring system, Spoofing could involve an attacker impersonating a legitimate sensor, while Tampering could involve altering transmitted vital-sign data.

PASTA – a seven-step process that aligns business objectives with technical threat analysis. Steps include defining business objectives, identifying technical assets, threat enumeration, vulnerability analysis, attack simulation, and risk assessment. Applying PASTA to a tele-health platform enables the organization to understand how a compromised video stream could affect patient confidentiality and clinical decision-making.

Vulnerability scanning – the automated process of probing a device or network for known weaknesses. Scanners compare detected configurations against databases of known CVEs (Common Vulnerabilities and Exposures). Regular scanning of a device's operating system can reveal missing patches that need to be applied before the device is placed in a clinical environment.

Penetration testing – a controlled, manual attempt to exploit vulnerabilities in order to evaluate the effectiveness of security controls. Pen testers simulate real-world attacks, often using the same tools and techniques as adversaries. A penetration test on a robotic surgery system might attempt to gain privileged access to the control console to assess the risk of unauthorized movement.

Static analysis – the examination of source code or binaries without executing them, to detect insecure coding patterns, hard-coded credentials, or unsafe library calls. Tools such as SonarQube or Coverity can be integrated into the build pipeline for continuous inspection. Static analysis of a device's firmware can uncover buffer-overflow vulnerabilities before they are shipped.

Dynamic analysis – the evaluation of software while it is running, often using instrumentation to monitor

memory usage, system calls, and network traffic. Dynamic analysis can reveal runtime behaviors that static analysis may miss, such as insecure handling of user input. For a device that streams patient data, dynamic analysis can verify that data is encrypted in transit.

Patch management – the process of acquiring, testing, and applying software updates to address known vulnerabilities. Effective patch management requires coordination between manufacturers, healthcare providers, and regulators. A patch that resolves a CVE affecting the TLS library of a MRI scanner must be validated for compatibility before deployment to avoid unintended downtime.

Configuration management – the systematic handling of device settings to ensure consistency, traceability, and security. Configuration baselines define secure default settings, and any deviation is logged and reviewed. For example, disabling the default “admin” account on a bedside monitor and enforcing strong password policies is a configuration-management best practice.

Hardening – the process of strengthening a device by removing unnecessary services, applying security patches, configuring strong authentication, and enforcing encryption. Hardening reduces the attack surface. A hardening checklist for a wireless ECG device might include disabling unused Bluetooth profiles, enabling TLS for all communications, and enforcing device-level password complexity.

Encryption – the transformation of data into a coded form that can only be read by authorized parties possessing the appropriate key. Encryption can be applied at rest (e.G., Storing patient data on a device’s flash memory) and in transit (e.G., TLS for network communication). Using AES-256 for data at rest and TLS 1.3 For data in transit is a common practice.

Authentication – the verification of an entity’s identity before granting access. Methods include passwords, certificates, biometric factors, and multi-factor authentication (MFA). In a hospital setting, a clinician may authenticate to a smart infusion pump using a smart card and a PIN, providing two factors of verification.

Authorization – the process of granting specific privileges to an authenticated entity. Role-based access control (RBAC) is a typical model, where users are assigned roles (e.G., Nurse, technician) that dictate which functions they can perform. An authorized nurse may adjust dosage parameters on a pump, whereas a technician may only be allowed to perform firmware updates.

Integrity – the assurance that data has not been altered in an unauthorized manner. Integrity mechanisms include digital signatures, hash verification, and checksums. For a pacemaker’s firmware, digital signatures ensure that only firmware signed by the manufacturer can be installed, preventing tampering.

Availability – the guarantee that a system or data is accessible when needed. Denial-of-service (DoS) attacks threaten availability by overwhelming resources. In a critical care unit, loss of availability of a ventilator’s control interface could have immediate life-threatening consequences.

Non-repudiation – the ability to prove that a specific action was performed by a particular entity, preventing

denial of involvement. Digital signatures and audit logs provide non-repudiation. An audit log that records every firmware update with a signed entry offers evidence that the update was authorized.

Audit log – a chronological record of events, such as user logins, configuration changes, and system errors. Logs are essential for forensic investigation and compliance verification. A device that records every access to patient data can help detect insider misuse and support regulatory reporting.

Incident response – a structured approach to detecting, analyzing, containing, and recovering from security incidents. Incident response plans typically include preparation, detection, containment, eradication, recovery, and post-incident lessons learned. For a medical device manufacturer, an incident response plan may outline how to notify customers of a vulnerability, release a patch, and provide remediation guidance.

Forensic analysis – the examination of digital evidence to determine the cause and impact of a security breach. Forensics may involve memory dumps, network traffic captures, and log correlation. After a ransomware event affecting a cardiac monitoring system, forensic analysis can identify the entry point (e.G., A compromised remote desktop protocol) and guide remediation.

Regulatory framework – the collection of laws, standards, and guidance documents that govern the design, testing, and post-market surveillance of medical devices. Key frameworks include the U.S. FDA's "Cybersecurity Guidance for Medical Devices," the European Union's "Medical Device Regulation (MDR)," and the International Electrotechnical Commission (IEC) standards such as IEC 62304, IEC 62443, and ISO 14971.

IEC 62304 – an international standard for the software life-cycle processes of medical device software. It defines requirements for software development, maintenance, risk management, and configuration control. IEC 62304 classifies software safety classes (A, B, C) based on potential impact on patient health, guiding the depth of verification and validation needed.

IEC 62443 – a series of standards that address industrial automation and control system security. Although originally focused on manufacturing, IEC 62443 is increasingly applied to medical devices because many devices share similar control-system architectures. The standard provides a risk-based approach to defining security levels, zones, and conduits.

ISO 14971 – the international standard for medical device risk management. It outlines a systematic process for identifying hazards, estimating risk, evaluating acceptability, and implementing controls. ISO 14971 is the foundation for integrating cybersecurity risk into overall device risk management.

FDA Guidance – a set of documents issued by the U.S. Food and Drug Administration that provide recommendations on cybersecurity best practices. Notable guidance includes "Content of Premarket Submissions for Management of Cybersecurity in Medical Devices" and "Postmarket Management of Cybersecurity Vulnerabilities." These documents clarify expectations for manufacturers regarding threat analysis, vulnerability disclosure, and remediation.

CE Marking – the certification that a medical device complies with EU regulations, including the MDR. The CE process requires a demonstration of conformity with safety and performance requirements, which now explicitly include cybersecurity considerations. A device seeking CE Marking must provide evidence of risk mitigation and post-market surveillance for cyber threats.

NIST – the National Institute of Standards and Technology, which publishes frameworks such as the NIST Cybersecurity Framework (CSF) and Special Publication 800-53. Although not regulatory, NIST guidance is widely adopted for establishing security controls and maturity models. Using NIST CSF, a hospital can map its security controls to identify gaps in device protection.

CVSS – the Common Vulnerability Scoring System, a standardized method for rating the severity of software vulnerabilities. CVSS scores range from 0 to 10, with higher values indicating greater risk. When a device's operating system receives a CVE with a CVSS score of 9.8, The manufacturer must prioritize remediation due to the high severity.

Zero-day – a vulnerability that is unknown to the vendor and for which no patch exists at the time of discovery. Zero-day exploits are especially dangerous because they can be leveraged before defenses are in place. An attacker discovering a zero-day in a widely used Bluetooth stack could potentially compromise thousands of devices before a fix is released.

Supply-chain attack – a compromise that occurs at any point in the product supply chain, often by inserting malicious code or hardware into components before they reach the manufacturer. Notable examples include the insertion of a backdoor into a microcontroller firmware during fabrication. Mitigating supply-chain risk involves vetting suppliers, performing code signing, and conducting integrity checks on received components.

Code signing – the practice of digitally signing software or firmware to verify its authenticity and integrity. Code signing keys must be protected, and the verification process must be enforced on the device. When a device receives a firmware update, it validates the signature before installation, preventing unauthorized modifications.

Secure boot – a mechanism that ensures only trusted software is executed during device startup. Secure boot verifies the digital signatures of the bootloader, kernel, and firmware, creating a chain of trust. A medical imaging device employing secure boot can resist attacks that attempt to replace the bootloader with malicious code.

Trusted Platform Module (TPM) – a hardware component that provides secure storage for cryptographic keys and supports platform integrity measurements. TPMs can be used to store code-signing keys, protect encryption keys, and enable secure boot. Integration of a TPM in a portable ultrasound device enhances protection of patient data and firmware integrity.

Network segmentation – the practice of dividing a network into isolated zones to limit the spread of

malware and restrict access to critical assets. Segmentation can be achieved using VLANs, firewalls, or air-gap techniques. In a hospital, placing surgical devices on a dedicated VLAN separate from the general office network reduces the risk of cross-contamination.

Air gap – a physical separation of a device or network from other networks, eliminating any electronic connection. Air-gapped devices are often used in high-risk environments where any remote access is considered unacceptable. For example, a legacy infusion pump that cannot be updated may be kept on an air-gapped network to prevent remote attacks.

Firewalls – security devices that filter inbound and outbound traffic based on defined rules. Firewalls can be hardware appliances or software components. Configuring a firewall to allow only HTTPS traffic to a device's management interface helps prevent unauthorized access.

Intrusion detection system (IDS) – a monitoring tool that analyzes network traffic or host behavior to identify suspicious activity. IDS can be signature-based (detecting known attack patterns) or anomaly-based (detecting deviations from normal behavior). Deploying an IDS on the network segment that hosts implantable cardiac devices provides early warning of potential attacks.

Intrusion prevention system (IPS) – similar to IDS but capable of actively blocking or mitigating detected threats. An IPS can drop malicious packets, terminate sessions, or quarantine compromised devices. Integrating IPS with device management platforms enables automated response to detected exploits.

Endpoint protection – security solutions installed on devices to protect against malware, unauthorized changes, and data leakage. Endpoint protection may include anti-virus, host-based firewalls, and application whitelisting. For a bedside monitor, endpoint protection ensures that only approved applications can run, reducing the risk of rogue code execution.

Application whitelisting – a security approach that permits only pre-approved software to execute, blocking all other binaries. Whitelisting is particularly effective for devices with limited functionality, where the set of legitimate applications is small and well-defined. A medical imaging workstation can use whitelisting to prevent the execution of unknown utilities.

Data diodes – unidirectional network devices that allow data to flow only in one direction, ensuring that information can be transmitted without the risk of inbound attacks. Data diodes are useful for transmitting patient data from a secure zone to an external analytics platform while preventing any command traffic from entering the secure zone.

Patch latency – the time elapsed between the release of a security patch and its application on a device. Reducing patch latency is critical for minimizing exposure to known vulnerabilities. Organizations often track patch latency as a key performance indicator (KPI) for cybersecurity readiness.

Vulnerability disclosure – the process by which researchers, users, or vendors report discovered

vulnerabilities to the responsible party. Disclosure can be coordinated (private) or public. Coordinated disclosure allows manufacturers time to develop a fix before the vulnerability is publicly disclosed, reducing the window of exploitation.

Bug bounty program – an initiative where organizations reward external security researchers for responsibly reporting vulnerabilities. Bug bounty programs can accelerate the discovery of hidden flaws and improve overall security posture. A medical device company may run a bug bounty to identify weaknesses in its remote monitoring platform.

Security by design – the principle of incorporating security considerations from the earliest stages of product development rather than adding them as an afterthought. Security by design includes threat modeling, secure coding, and rigorous testing. Devices built with security by design are typically more resilient to attacks.

Security by obscurity – a flawed approach that relies on hiding system details to achieve security. While obscurity may add a minor layer of difficulty for attackers, it does not replace robust security controls. Relying on undocumented communication protocols without encryption is an example of security by obscurity that should be avoided.

Risk mitigation – the implementation of controls and measures to reduce the likelihood or impact of a risk. Mitigation strategies may include technical controls (encryption, authentication), procedural controls (training, policies), and physical controls (tamper-evident seals). For a drug-delivery device, risk mitigation might involve encrypting wireless commands and requiring dual-operator approval for dosage changes.

Residual risk – the amount of risk remaining after mitigation measures have been applied. Residual risk must be evaluated to determine whether it is acceptable according to regulatory standards. If the residual risk of a remotely programmable pacemaker remains above the acceptable threshold, additional controls such as intrusion detection may be required.

Acceptable risk – the level of risk that is deemed tolerable based on clinical benefit, regulatory guidance, and stakeholder consensus. Determining acceptable risk involves balancing patient safety against device functionality and market needs. Manufacturers document acceptable risk decisions in their risk-management files.

Threat intelligence – information about emerging threats, attacker tactics, techniques, and procedures (TTPs). Threat intelligence can be sourced from industry sharing groups, commercial feeds, or governmental alerts. Leveraging threat intelligence enables a device manufacturer to anticipate and defend against new attack vectors.

Security policy – a formal document that defines an organization’s security objectives, responsibilities, and rules. Security policies cover areas such as acceptable use, password management, incident response, and data protection. A hospital’s security policy may mandate that all medical devices be scanned for

vulnerabilities quarterly.

Security awareness training – educational programs designed to inform staff about cybersecurity risks and safe practices. Training topics include phishing recognition, proper handling of removable media, and reporting suspicious activity. Effective security awareness reduces the likelihood of human-error-driven incidents.

Device hardening checklist – a structured list of actions to secure a device before deployment. Items typically include disabling unused services, enabling encryption, configuring strong passwords, and applying the latest patches. Checklists help ensure consistency across multiple devices and simplify audits.

Compliance audit – an independent review that assesses whether an organization meets regulatory and contractual requirements. Audits may examine documentation, processes, and technical controls. A compliance audit for a manufacturer's ISO 14971 implementation validates that cybersecurity risks are appropriately managed.

Change control – a formal process for managing modifications to a device's hardware, software, or documentation. Change control ensures that any alteration is reviewed, tested, and approved before implementation. For instance, updating the cryptographic library of a monitoring device requires a documented change request, impact analysis, and regression testing.

Configuration drift – the gradual divergence of a device's configuration from its intended baseline due to ad-hoc changes, updates, or mismanagement. Drift can introduce security gaps, such as re-enabling a default password. Regular configuration audits help detect and correct drift.

Business continuity plan (BCP) – a strategy that outlines how essential services will be maintained during disruptions, including cyber incidents. A BCP for a critical care unit may define alternative workflows, backup equipment, and communication protocols in case of a ransomware event that disables primary devices.

Disaster recovery – the set of procedures for restoring systems after a catastrophic failure. Disaster recovery focuses on data restoration, system rebuild, and verification of functionality. For a radiology department, disaster recovery includes restoring PACS servers, re-establishing network connections, and validating image integrity.

Red team exercise – a simulated attack performed by an internal or external group that adopts the perspective of an adversary. Red team exercises test detection, response, and resilience. Conducting a red team exercise on a hospital's network that includes connected infusion pumps can reveal gaps in segmentation and monitoring.

Blue team – the defenders who monitor, detect, and respond to security incidents. Blue team activities include log analysis, threat hunting, and incident response. In a medical device environment, the blue team may be responsible for monitoring device health dashboards for anomalous behavior.

Threat hunting – the proactive search for hidden threats within a network or system, often using advanced analytics and hypothesis-driven investigations. Threat hunting can uncover stealthy malware that evaded automated detection. A threat-hunting campaign targeting anomalous outbound traffic from a cardiac monitor can reveal covert data exfiltration attempts.

Secure coding standards – documented rules that guide developers in writing code that resists common vulnerabilities. Examples include the CERT C Secure Coding Standard and OWASP Secure Coding Practices. Applying secure coding standards reduces the likelihood of buffer overflows, injection flaws, and improper error handling.

Input validation – the process of checking data received from external sources to ensure it conforms to expected formats and ranges. Proper input validation prevents injection attacks, such as SQL injection or command injection. A device that receives configuration commands over a network must validate that numeric parameters fall within safe clinical limits.

Output encoding – the transformation of data before it is rendered to a user interface or transmitted to another system, ensuring that special characters are not interpreted as code. Output encoding mitigates cross-site scripting (XSS) attacks in web-based device interfaces. When a device displays a user-provided label, encoding prevents malicious scripts from executing in the browser.

Memory safety – techniques that prevent common memory-related bugs such as buffer overflows, use-after-free, and integer overflows. Languages like Rust provide built-in memory safety, while C/C++ require careful manual management. Selecting memory-safe languages for new device firmware can dramatically reduce vulnerability density.

Static Application Security Testing (SAST) – a type of static analysis that focuses specifically on security flaws within source code. SAST tools scan for patterns that indicate insecure practices, such as hard-coded credentials or weak cryptographic algorithms. Integrating SAST into the continuous integration pipeline enables early detection of security defects.

Dynamic Application Security Testing (DAST) – a testing approach that evaluates running applications for security weaknesses, often by sending crafted inputs and observing responses. DAST can uncover runtime issues such as insecure session handling or improper error messages. Performing DAST on a device’s web management console helps verify that authentication mechanisms cannot be bypassed.

Fuzz testing – an automated technique that supplies random or malformed inputs to a program to discover crashes, memory leaks, or unexpected behavior. Fuzz testing is effective for uncovering hidden bugs in parsers, protocol stacks, and file-format handlers. Applying fuzz testing to a device’s Bluetooth stack can reveal vulnerabilities that could be exploited remotely.

Side-channel attack – an attack that exploits indirect information, such as power consumption, electromagnetic emissions, or timing differences, to infer secret data. Side-channel attacks are relevant for

devices that perform cryptographic operations, especially when hardware protections are weak. Protecting against side-channel attacks may involve adding noise, randomizing execution timing, or using dedicated cryptographic hardware.

Supply-chain verification – the process of confirming the authenticity and integrity of components received from suppliers. Verification methods include cryptographic hash checks, serial-number tracking, and visual inspection of tamper-evident seals. Conducting supply-chain verification on each batch of microcontrollers helps prevent the introduction of compromised parts.

Digital twin – a virtual replica of a physical device that can be used for testing, simulation, and monitoring. Digital twins enable security teams to assess the impact of hypothetical attacks without risking patient safety. By creating a digital twin of an insulin pump, developers can simulate firmware attacks and evaluate mitigation strategies.

Secure firmware update – a process that ensures firmware is delivered, verified, and installed without exposing the device to tampering. Secure update mechanisms typically involve encrypted transport, digital signatures, and rollback protection. A secure firmware update for a wireless pulse oximeter might use TLS for transmission and verify an ECDSA signature before flashing.

Rollback protection – a safeguard that prevents an attacker from installing an older, vulnerable version of firmware after a patch has been applied. Rollback protection can be enforced by maintaining a monotonic version counter and refusing to install firmware with a lower version number. Implementing rollback protection on a cardiac defibrillator helps ensure that a fixed vulnerability cannot be re-introduced.

Secure key storage – the practice of protecting cryptographic keys from unauthorized access or extraction. Hardware security modules (HSMs), TPMs, and secure elements provide tamper-resistant storage. A device that encrypts patient data must store its encryption keys in a secure element to prevent extraction by physical attackers.

Key rotation – the periodic replacement of cryptographic keys to limit the amount of data protected by any single key. Key rotation mitigates the impact of a key compromise. For a cloud-based analytics service that aggregates data from multiple devices, rotating the TLS certificates every 90 days reduces exposure if a certificate is leaked.

Certificate revocation – the process of invalidating a digital certificate before its expiration date, typically because the private key has been compromised. Revocation mechanisms include Certificate Revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP). Devices must check revocation status when establishing TLS connections to ensure the peer’s certificate remains trustworthy.

Secure enclave – an isolated execution environment within a processor that protects sensitive code and data from the rest of the system. Secure enclaves can be used to perform cryptographic operations or store secrets. Leveraging a secure enclave in a wearable cardiac monitor adds an extra layer of protection for

patient-specific keys.

Hardware root of trust – a set of hardware components that provide a trusted foundation for secure boot, attestation, and key management. The root of trust is immutable and verified at the earliest stage of device startup. Establishing a hardware root of trust in a surgical robot ensures that only authenticated firmware can run.

Attestation – the process by which a device proves its integrity and software state to a remote verifier. Attestation can be based on measured boot values, TPM PCRs, or secure enclave reports. A hospital's management system may request attestation from a connected infusion pump before allowing dosage commands.

Secure development environment – a controlled workspace where developers write, build, and test code with security controls such as isolated networks, restricted internet access, and mandatory code-signing. Using a secure development environment reduces the risk of introducing malicious code during development.

Continuous integration/continuous deployment (CI/CD) – an automated pipeline that builds, tests, and releases software changes. Integrating security checks into CI/CD (often called "DevSecOps") ensures that security is evaluated on every commit. For a device firmware project, the CI/CD pipeline could run static analysis, unit tests, and a signed artifact generation step before publishing the update.

DevSecOps – the practice of embedding security into DevOps workflows, fostering collaboration between development, security, and operations teams. DevSecOps encourages early detection of vulnerabilities, automated compliance checks, and rapid remediation. Applying DevSecOps to medical device software helps align rapid innovation with stringent regulatory expectations.

Pen-test report – a document that details the findings, methodology, and recommendations from a penetration test. The report includes identified vulnerabilities, proof-of-concept exploits, risk ratings, and remediation guidance. Reviewing the pen-test report for a ventilator's network interface can guide the engineering team in prioritizing fixes.

Remediation plan – a structured plan that outlines how identified vulnerabilities will be addressed, including timelines, responsibilities, and verification steps. A remediation plan may specify that a critical buffer-overflow vulnerability will be patched within 30 days, with regression testing to confirm that the fix does not affect dosage calculations.

Post-market surveillance – the ongoing monitoring of a device's performance, safety, and security after it has been released to the market. Post-market surveillance includes collecting field data, analyzing incident reports, and updating risk assessments. In the context of cybersecurity, post-market surveillance tracks emerging threats and coordinates vulnerability disclosures.

Vulnerability management – the end-to-end process of identifying, assessing, prioritizing, and mitigating vulnerabilities throughout the device's lifecycle. Effective vulnerability management requires a clear governance structure, defined service-level agreements (SLAs), and communication channels with customers. A robust vulnerability-management program can reduce the average time to patch from months to weeks.

Incident reporting – the formal communication of a security incident to internal stakeholders, regulators, and possibly affected patients. Reporting timelines are often mandated by regulations; for example, the FDA requires manufacturers to report certain cybersecurity events within 30 days. Incident reporting must include details about the nature of the incident, impact assessment, and corrective actions.

Root cause analysis – the systematic investigation to determine the underlying cause(s) of an incident. Root cause analysis helps prevent recurrence by addressing fundamental weaknesses rather than symptoms. After a ransomware infection of a device's management server, root cause analysis might reveal that an outdated remote-desktop protocol was the entry point.

Patch deployment strategy – the plan for delivering patches to devices in the field, balancing speed, safety, and operational continuity. Strategies may include staged rollouts, automated updates, or manual installation by service technicians. For life-supporting devices, a patch deployment strategy often incorporates a validation phase to ensure that the patch does not affect critical functionality.

Service-level agreement (SLA) – a contract that defines the expected level of service, including response times, availability, and support. In cybersecurity, SLAs may specify maximum time to acknowledge a vulnerability report or to apply a critical security update. Clear SLAs help set expectations with healthcare providers and ensure accountability.

Security governance – the set of policies, processes, and organizational structures that direct and control security activities. Governance establishes roles such as Chief Information Security Officer (CISO), defines authority, and aligns security objectives with business goals. Effective security governance ensures that cybersecurity is not an afterthought but a core responsibility.

Risk appetite – the amount of risk an organization is willing to accept in pursuit of its objectives. Risk appetite influences decisions about which controls to implement and which residual risks are tolerable. A hospital with a high risk appetite for innovative therapies may accept certain cyber risks in exchange for faster deployment of cutting-edge devices.

Security control – any safeguard—technical, administrative, or physical—that reduces risk. Controls can be preventive (e.G., Firewalls), detective (e.G., IDS), or corrective (e.G., Patching). Selecting appropriate security controls for a device involves mapping threats to mitigations in a risk-treatment plan.

Technical control – a security measure implemented through technology, such as encryption, access control lists, or intrusion detection. Technical controls are often automated and can be enforced at scale. For a

network of infusion pumps, technical controls may include mandatory TLS for all device-to-server communication.

Administrative control – policies, procedures, and training that manage human behavior and organizational processes. Administrative controls complement technical measures. Examples include password policies, security awareness programs, and incident-response procedures.