PARASOFT®

WHITEPAPER

How to Accelerate ISA/IEC 62443 Compliance With Automated Software Testing

Overview

Industrial automation and control systems (IACS) are increasingly under threat from cyberattacks as they become more interconnected and exposed to the internet. Malicious actors are targeting industrial systems for motives ranging from cybercrime to nation-state disruption.

On May 30, 2024, the U.S. Cybersecurity and Infrastructure Security Agency (CISA) released seven Industrial Control Systems (ICS) advisories, warning that IACS breaches can be devastating as they can disrupt critical infrastructure and threaten human safety.

To address these escalating cyber risks, the International Society of Automation (ISA) and the International Electrotechnical Commission (IEC) in 2018 developed the IEC 62443 series of standards for securing IACS. IEC 62443 provides a comprehensive framework for implementing cyber defenses across all IACS domains, including people, processes, and technology.

However, ensuring full IEC 62443 compliance can be overwhelming for asset owners and operators. IACS typically involve heterogeneous systems across multiple sites with complex integration requirements. Maintaining comprehensive asset inventories and testing all possible attack surfaces is difficult without automated tooling.

This whitepaper explores how automated software testing techniques can accelerate and streamline IEC 62443 compliance.

ISA/IEC 62443: Understanding the Framework

The ISA/IEC 62443 is a series of standards developed to provide a comprehensive framework for securing IACS. The standard is divided into several parts, each addressing different aspects of IACS security, from general concepts and models to system and component security requirements.

One of the key parts of the ISA/IEC 62443 standard is Part 4-1: Secure Product Development Lifecycle Requirements. This part focuses on the security aspects of the product development life cycle, providing guidelines for secure product design, implementation, verification, and validation. It emphasizes the importance of incorporating security considerations right from the initial stages of product development, rather than as an afterthought.

IEC 62443-4-1 is foundational for enabling secure IACS because it mandates "cybersecurity by design" principles across all phases of the software development life cycle (SDLC). From the initial requirements analysis through design, implementation, testing, and maintenance, security controls must be an integral part of the process.

Some key requirements covered in IEC 62443-4-1 include:

- Secure SDLC processes and documentation
- Specifying security requirements based on risk assessments
- Using secure coding practices and static analysis
- Robust configuration management and change control
- Penetration testing and security verification testing
- Defect tracking and effective patch management processes

The provisions of IEC 62443-4-1 call for Industrial Control Systems makers to address security throughout the SDLC. This is to ensure that IACS products have cybersecurity deeply embedded into their core design and implementation. This "secure by design" approach is critical for protecting IACS from rapidly evolving cyber threats.

From a software testing perspective, IEC 62443-4-1 highlights the need for automated security testing as it makes it easier to validate the cyber integrity of IACS applications continuously. Automated security testing is essential for thorough security requirements verification, regression testing for vulnerabilities, and maintaining IEC 62443 compliance as products evolve.

Why Is ISA/IEC 62443 Crucial for Industrial Automation and Control Systems?

The IACS industry, including the internet of things (IoT) devices that are increasingly integrated into these environments, faces mounting cybersecurity threats as these control systems become more interconnected and exposed.

Recent years have seen an alarming rise in cyberattacks specifically targeting industrial control systems across sectors like energy, manufacturing, transportation, and utilities. Since 2010, we've heard about ICS-centered malware such as Stuxnet and more recent ones like Triton and PIPEDREAM.

Beyond malware, Kaspersky's ICS CERT Predictions for 2024 report warns that ransomware will remain the top cybersecurity threat for industrial businesses this year. The report found that 18% of ransomware attacks on industrial firms led to halts in production or delivery of products like medical devices, power grids, and transportation systems in 2023. This demonstrates the consequences of cyberattacks on critical infrastructure.

With IACS increasingly converging with enterprise IT systems, AI and connect to the cloud, their attack surface is expected to expand much more than we've seen in the last few years. Legacy "insecure-by-design" control systems never intended for internet exposure are now accessible entry points for adversaries.

All the above point to the urgent need to secure IACS from escalating cyber threats; and adhering to the ISA/IEC 62443 standards is a crucial starting point for asset owners and integrators. This is because it offers a holistic framework for implementing multi-layered cybersecurity controls across people, processes, and technology for IACS.

While initially focused on industrial sectors, ISA/IEC 62443 has also been the preeminent cybersecurity standard for the medical industry that relies on networked medical devices and control systems. However, to better align with the development of medical devices, new standard IEC 81001-5-1 "Health software and health IT systems safety, effectiveness and security – Part 5-1: Security – Activities in the product lifecycle," has been approved by the FDA.

Therefore, implementing ISA/IEC 62443 enables organizations to proactively manage cybersecurity risk across the entire IACS life cycle through processes like:

- Rigorous risk assessment and mitigation
- Secure system design and hardening
- Continuous monitoring and incident response
- Automation of vulnerability testing and security validation

Software Complexity: A Key Hurdle in Satisfying ISA/IEC 62443

While the ISA/IEC 62443 standards provide a comprehensive framework for securing IACSs, achieving and maintaining full compliance presents significant challenges for asset owners and operators. A primary hurdle is the sheer complexity of the software that powers modern IACS environments.

IACS increasingly rely on heterogeneous software components sourced from multiple vendors and integrators, often built using diverse coding languages, frameworks, and toolchains. From real-time operating systems and programmable logic controllers to supervisory control and data acquisition (SCADA) systems, distributed control systems, safety instrumented systems, and more, each of these elements represents a potential attack surface that must be comprehensively secured.

The interconnected nature of IACS amplifies this complexity. Plant systems are integrated across functional boundaries and geographic locations, inheriting risk from every interdependent application, middleware component, and infrastructure layer. Monolithic legacy applications designed with insecure practices further compound these challenges.

Keeping pace with the daily deluge of new disclosed vulnerabilities that could impact a facility's unique risk profile is virtually impossible through manual processes alone. Consistently applying security patches and validating whether actual vulnerabilities have been effectively remediated requires continuous automated security testing.

Beyond dealing with the complexity of deployed IACS software assets, organizations must also enforce security throughout each stage of the software development life cycle per IEC 62443-4-1 requirements. From secure coding to penetration testing, every phase mandates robust processes and granular traceability that is difficult to achieve cost-effectively without automation.

Additionally, IACS software is constantly evolving through updates, integrations with new components, and migration initiatives. This necessitates comprehensive regression testing to continuously validate the cumulative security posture of these dynamic, multi-layered applications.

The following sections will explore specific automated testing techniques to accelerate this journey.

Cutting Complexity Through Software Testing: A Path to ISA/IEC 62443 Compliance

Addressing software complexity challenges that impede ISA/IEC 62443 compliance requires adopting highly automated testing techniques across the entire SDLC. Let's check out how highlighting some of the security testing practices below can help organizations identify and remediate vulnerabilities in their ICS at scale.

- **Static analysis.** Implementing robust static analysis, also known as static application security testing (SAST), is a core practice for identifying security defects early in the SDLC before they become entrenched vulnerabilities. SAST analyzes source code, bytecode, binaries, and build artifacts to detect insecure coding patterns like SQL injection, buffer overflows, cryptography flaws, and more. Advanced SAST techniques also assess complex software compositions by analyzing open source risk, software bills-of-materials (SBOMs), and mapping security guidance.
- **Unit testing.** Comprehensive unit test suites with security tests embedded directly into developer workflows are vital for validating that security requirements are satisfied by implementation code. Unit tests focused on authentication, data encryption, data decryption, access control, input/output validation, error handling, and other security functions provide fast feedback cycles.
- **Integration testing.** As software integrations proliferate across IACS environments, automated integration tests are imperative for uncovering security vulnerabilities that stem from multi-component interactions. Simulating realistic runtime conditions, integration tests validate correct security configurations and identify breaks caused by connected software dependencies.
- **Requirements-based testing.** A core tenet of ISA/IEC 62443 is to design security from initial requirements analysis. By transforming specified security requirements into executable tests, organizations can automatically validate that system behavior correctly implements defined security controls. Requirements traceability reports capture proof that all security requirements were comprehensively implemented and tested.
- **Code coverage.** Using code coverage tools brings to light which portions of the security code have been duly tested versus untested areas that represent risk. Coverage metrics like modified condition/decision coverage (MC/DC) are best to use to drive complete security verification in safety-critical and high-risk software modules.

By applying these testing practices in alignment with ISA/IEC 62443 guidance, organizations can pinpoint and remediate security vulnerabilities in a rapid, cost-effective manner as IACS software evolves. This significantly reduces overall risk while accelerating compliance initiatives.

A Broader Look at the IACS Industry Compliance Landscape

While ISA/IEC 62443 is the primary standard for cybersecurity in IACS, it's just one piece of a broader regulatory and standards landscape that IACS developers and operators must navigate.

Another key standard that intersects with IEC 62443 for safety-critical IACS applications is IEC 61508 for functional safety of electrical/electronic/programmable electronic safety-related systems. IEC 61508 provides requirements for ensuring systems operate correctly in response to potentially dangerous conditions.

Both IEC 62443 and IEC 61508 mandate rigorous software development processes, but 61508 focuses specifically on systematic capabilities to handle relevant safety functions. A concept from IEC 61508 called the V-model depicts the relationships between different phases of the software safety life cycle.

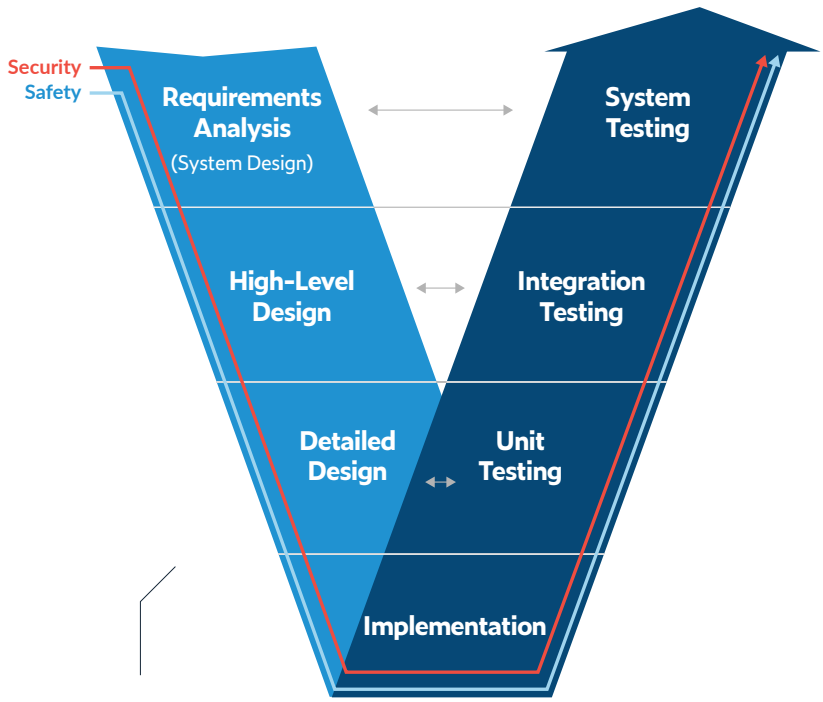

Figure 1: Integrated safety and security verification in the V-model software life cycle

The V-model emphasizes the need for traceability. Each development phase has associated verification and validation activities to ensure that teams meet requirements. This aligns with the core principles of 62443 for security requirements traceability and robust testing and validation.

For IACS deployments, organizations may also need to comply with other standards like:

- ISO/IEC 2700. General requirements for information security management systems.
- ISO 13849. Safety requirements for control systems on machinery.
- IEC 62061. Functional safety for machinery control systems.
- NIST SP 800-82. Guidelines for securing industrial control systems.

While each standard has unique requirements, they're unified in mandating secure development practices coupled with rigorous testing and validation activities, making automated security testing an essential compliance enabler across the entire IACS regulatory landscape.

Related Coding Standards for ISA/IEC 62443

While ISA/IEC 62443 provides overarching guidance for secure development practices like static code analysis, it does not reference specific coding standards. However, adopting complementary coding standards and security standards is essential for IACS developers to avoid common vulnerabilities.

CERT Coding Standards

Developed by the computer emergency response team (CERT), these standards provide rules and recommendations for secure coding in various programming languages like C, C++, Java, and more. CERT secure coding rules cover areas like memory management, input validation, encryption, and mitigation of common weakness types cataloged by CWE.

MISRA Coding Standards

The Motor Industry Software Reliability Association (MISRA) maintains standards focused on embedded software security and safety. MISRA C/C++ coding guidelines promote safe and secure programming styles by restricting the use of insecure language features that can lead to vulnerabilities.

AUTOSAR C++14 Guidelines

The AUTOSAR (AUTomotive Open System ARchitecture) developed C++14 coding rules tailored for automotive and embedded systems. Like MISRA, these guidelines facilitate writing robust and secure C++ code by steering developers away from language constructs prone to weakness and undefined behavior.

CWE/SANS Top 25

The Common Weakness Enumeration (CWE) maintained by MITRE provides a standardized list of the most prevalent and critical software vulnerabilities. The SANS Top 25 ranks and analyzes the most widespread and impactful weaknesses from CWE to help organizations prioritize secure coding efforts.

OWASP Secure Coding Practices

These secure coding practices provide developers with a comprehensive look at core security concepts and techniques for building secure software across major programming languages and platforms. Their resources include coding requirements, vulnerability mitigation, and security testing.

IEC 62443-4-1 provides requirements for a secure development life cycle to reduce security defects in IACS software, although it does not mandate specific coding standards or vulnerability taxonomies. Automating the analysis and enforcement of secure coding practices through static analysis and coding principles can be vital for achieving compliance with IEC 62443 at scale and effectively mitigating security risks.

How Organizations Can Simplify This Broader Compliance Spectrum

Achieving comprehensive compliance across the expansive ecosystem of regulations, standards, and best practices relevant to modern IACS environments is an immense challenge. From ISA/IEC 62443 to functional safety standards like IEC 61508, quality systems mandates, machinery safety controls, and more, organizations face a daunting matrix of complex, interconnected requirements.

However, by embracing key strategic principles around automation, optimization, integration and collaboration, IACS companies can simplify their compliance initiatives and enhance their overall cyber resilience posture.

Automation

Given the sheer volume of security testing, code analysis, risk assessments, documentation, and other verification activities specified by standards like IEC 62443, automation is paramount. Manual processes simply cannot scale to efficiently validate adherence to hundreds of discrete requirements across complex, rapidly evolving IACS applications and environments.

Organizations must adopt intelligent test automation solutions that can scan software for compliance gaps related to secure coding, vulnerability management, security requirements traceability, penetration testing, and more. Automating compliance evidence collection and reporting is also essential.

Optimization

Rather than implementing separate siloed processes for each relevant standard, a streamlined “compliance by design” methodology should optimize activities to ensure efficient coverage across all applicable regulations. For example, a unified secure development life cycle can concurrently enforce cybersecurity, safety, and quality controls throughout all product phases.

Thus, by mapping all standards to an optimized compliance framework, asset owners and integrators can eliminate redundant workstreams while still satisfying every external mandate through an economical set of integrated practices.

Integration

No single standard provides a complete picture of cyber resilience for industrial environments. IACS compliance obligations span cybersecurity, functional safety, risk management, quality systems and more across different vertical domains. An integrated GRC (Governance, Risk & Compliance) approach is required.

Modern compliance automation solutions play a vital role in this integrated approach. These solutions can aggregate rules from diverse standards, merging them into a unified control framework. Subsequently, they coordinate testing and verification processes for all compliance mandates using a cohesive set of toolchains and workflows. This integrated approach prevents organizations from operating in isolated silos and enhances traceability across compliance efforts. Ultimately, it enables organizations to manage their compliance obligations more effectively and efficiently in complex industrial environments.

Collaboration

Establishing a culture of security where compliance is a shared responsibility across the entire IACS ecosystem is critical. OEMs, asset owners, operators, integrators and third-parties must collaborate on harmonizing development, deployment and maintenance activities to a common set of unified secure practices.

Beyond collaboration within internal teams, organizations should participate in industry associations and working groups that help further evolve and align relevant standards to the cutting edge of ICS/OT cyber defense. A collaborative mindset enables faster response to emerging challenges.

Not only does combining the above measures help organizations navigate the complex IACS regulatory landscape, but it also reduces overhead while elevating their security posture through a streamlined set of cohesive processes.

Best Practices for Satisfying ISA/IEC 62443 Verification Requirements

Adhering to the rigorous verification methods outlined in ISA/IEC 62443 requires IACS organizations to adopt robust secure development practices. Following are some key best practices.

Continuous Code Review

Per IEC 62443-4-1 requirements, comprehensive code review and static analysis must be an integral part of the SDLC to identify vulnerabilities early before they become entrenched defects. Automating static application security testing within developer IDEs and commit workflows enables continuous code scanning. This provides rapid feedback loops for fixing security flaws aligned with secure coding standards like CERT.

Set Coding Standards From the Outset

It's crucial to clearly define and enforce security-focused coding policies that conform to industry standards like CERT, MISRA, OWASP, and so on, from the outset of the project. Developers should follow secure conventions for areas like memory management, input validation, authentication, cryptography, and more.

Integrating SAST rules that automatically validate adherence to these coding standards helps prevent security anti-patterns from springing up. Baseline security should be established through policies and gating criteria.

Follow Secure Supply Chain Practices for Third-Party Software

The proliferation of third-party components and open source dependencies significantly expands the attack surface of IACS applications. Organizations should implement secure software composition analysis (SCA) to manage and secure the use of open source software components within a larger software application.

Maintaining detailed and updated software bills-of-materials (SBOM) for all product versions is also critical to support ongoing monitoring and mitigation, aligned with IEC 62443 guidance. SCA and SBOM practices must be automated at scale to analyze the massive volume of external software being consumed into software factories today.

Perform Continuous Testing

In addition to automated code analysis, IEC 62443 specifies stringent security testing throughout all SDLC phases and after any changes or updates. This includes everything from security unit testing to integration tests, risk-based scenario tests, penetration testing, and full regression testing. Organizations should equip DevOps pipelines to continuously execute automated security tests as part of their CI/CD workflows.

Require Suppliers to Show Documentation of Their Own Cybersecurity Policies

Per IEC 62443's software supply chain risk management controls, IACS asset owners should validate that any third-party suppliers can document their adherence to secure development processes. This involves reviewing evidence like cybersecurity policies/standards, process definition documentation, risk assessments, tool usage, testing methodologies, defect management, and more. Without rigorous verification of security practices throughout their supplier ecosystems, IACS cyber resilience is compromised.

Use TARA Strategy

Developers working in IACS organizations should adopt a Threat Assessment & Remediation Analysis (TARA) as it's strategically aligned with IEC 62443 directives. This involves having an inventory of all deployed software assets and components, analyzing them for vulnerabilities based on threat intelligence sources, and rapidly prioritizing and remediating the most critical exposed risks.

By monitoring new vulnerability disclosures and updated guidance from sources like ICS-CERT, NVD, and CWE and automatically correlating them to an updated SBOM, organizations can maintain full traceability of their cyber exposure and orchestrate remediation activities accordingly. Interactive application security testing (IAST) techniques can also provide runtime visibility into real exploits and attacks across the IACS attack surface.

Automated Testing for ISA/IEC 62443 Compliance

Achieving comprehensive, continuous compliance with the rigorous verification and validation requirements outlined in ISA/IEC 62443 demands highly automated testing capabilities. Manual processes simply cannot scale to validate adherence across the complex collection of security controls defined by IEC 62443 and its complementary standards. Below lists how intelligent automation can accelerate IEC 62443 compliance.

Comprehensive Support for IACS Standards

Advanced compliance automation platforms provide out-of-the-box support for importing rules and requirements from IEC 62443 along with other relevant industry regulations like IEC 61508, ISO 27001, NIST 800-82, and more. This centralized repository maps all mandates to an integrated control framework for automated testing and verification. Some platforms may offer more extensive support for or provide additional features tailored to specific compliance requirements. Organizations should evaluate different platforms to determine which one best meets their needs and aligns with the regulations they need to comply with.

Customized Compliance Reporting and Advanced Analytics

Beyond prebuilt forms, organizations can fully customize reporting templates to capture verifiable evidence tailored to their unique auditing needs. Interactive dashboards with advanced visualizations and analytics provide continuous visibility into IEC 62443 compliance posture across all applicable security controls.

Continuous Software Testing

IEC 62443 software verification and validation through the development life cycle is best achieved by integrating security testing procedures directly into the development and operations (DevOps) pipelines. As the software undergoes changes and updates, security measures are consistently validated to ensure ongoing protection against potential threats. This approach ensures that security is not an afterthought but a fundamental aspect of software development and delivery.

Automated Verification of Internal Coding Standards

Alongside adopting coding standards like CERT, many organizations implement their own tailored internal security coding policies. These internal guidelines are crafted to tackle the distinct security challenges and specifications of their systems. Automated code analysis tools are frequently employed to ensure adherence to both external standards and internal policies. This process guarantees that software development practices consistently align with essential security standards, minimizing potential risks across various development environments.

AI and ML for Better Code Analysis

The integration of AI and ML capabilities into static analysis tools marks a significant advancement for organizations seeking to bolster the security and reliability of their software systems. One noteworthy application of AI in static analysis involves leveraging historical code interactions and prior analysis outcomes to contextualize and prioritize identified coding violations and vulnerability findings.

These innovative technologies offer the promise of streamlining software quality processes by automating manual efforts and enhancing the effectiveness of static analysis. This represents just the beginning of a transformative journey, with continued innovation and refinement poised to further elevate the efficacy of AI-driven static analysis in the safety- and security-critical software domain.

Automated Unit Test Creation

Unit testing is a fundamental principle outlined in IEC 62443 and plays a critical role in enhancing the quality and reliability of software codebases. By facilitating the systematic verification of individual code units, unit testing enables developers to identify security defects early in the development process, thereby reducing the risk of costly errors and vulnerabilities reaching production.

Automated execution of unit tests streamlines the testing workflow, providing developers with swift feedback on the functionality of their code. Additionally, support for mocking and stubbing allows developers to isolate code units and simulate dependencies, ensuring comprehensive testing coverage. Ultimately, unit testing cultivates a culture of quality and empowers development teams to deliver secure, safe, and dependable software products.

Summary

The stakes of compromised industrial automation and control systems (IACS) environments are too high to overlook. Ensuring comprehensive compliance with the ISA/IEC 62443 standard has become imperative to safeguarding these critical systems from the increasing threat of cyberattacks.

However, maintaining continuous adherence to the ISA/IEC 62443 standard presents significant challenges, particularly given the intricate nature of IACS software. The standard's stringent verification requirements encompassing secure coding, vulnerability testing, penetration assessments, and requirements traceability pose considerable obstacles that cannot be efficiently addressed through manual processes alone.

This is where intelligent test automation emerges as a crucial solution. By integrating automated security scanning, dynamic analysis, and validation testing directly into CI/CD pipelines, organizations can achieve end-to-end ISA/IEC 62443 compliance as IACS applications evolve.

By leveraging test automation, organizations can proactively mitigate risks, ensure regulatory compliance, and fortify the resilience of their industrial automation and control systems against evolving cyber threats.

Take the Next Step

Talk to a compliance expert to learn how your embedded software development team can accelerate ISA/IEC 62443 compliance with automated software testing

About Parasoft

Parasoft helps organizations continuously deliver high-quality software with its AI-powered software testing platform and automated test solutions. Supporting the embedded, enterprise, and IoT markets, Parasoft's proven technologies reduce the time, effort, and cost of delivering secure, reliable, and compliant software by integrating everything from deep code analysis and unit testing to web UI and API testing, plus service virtualization and complete code coverage, into the delivery pipeline. Bringing all this together, Parasoft's award-winning reporting and analytics dashboard provides a centralized view of quality, enabling organizations to deliver with confidence and succeed in today's most strategic ecosystems and development initiatives—security, safety-critical, Agile, DevOps, and continuous testing.