Defensible 10

Annex G (Normative): D07-Threat & Vulnerability Security Engineering

Technical Standards

Standards Committee
1-5-2026

About ISAUnited

The Institute of Security Architecture United is the first dedicated Standards Development Organization (SDO) focused exclusively on cybersecurity architecture and engineering through security-by-design. As an international support institute, ISAUnited helps individuals and enterprises unlock the full potential of technology by promoting best practices and fostering innovation in security.

Technology drives progress; security enables it. ISAUnited equips practitioners and organizations across cybersecurity, IT operations, cloud/platform engineering, software development, data/AI, and product/operations with vendor-agnostic standards, education, credentials, and a peer community—turning good practice into engineered, testable outcomes in real environments.

Headquartered in the United States, ISAUnited is committed to promoting a global presence and delivering programs that emphasize collaboration, clarity, and actionable solutions to today's and tomorrow's security challenges. With a focus on security by design, the institute champions the integration of security into every stage of architectural and engineering practice, ensuring robust, resilient, and defensible systems for organizations worldwide.

Disclaimer

ISAUnited publishes the ISAUnited Defensible 10 Standards Technical Guide to provide information and education on security architecture and engineering practices. While efforts have been made to ensure accuracy and reliability, the content is provided “as is,” without any express or implied warranties. This guide is for informational purposes only and does not constitute legal, regulatory, compliance, or professional advice. Consult qualified professionals before making decisions.

Limitation of Liability

ISAUnited - and its authors, contributors, and affiliates - shall not be liable for any direct, indirect, incidental, consequential, special, exemplary, or punitive damages arising from the use of, inability to use, or reliance on this guide, including any errors or omissions.

Operational Safety Notice

Implementing security controls can affect system behavior and availability. First, validate changes in non-production, use change control, and ensure rollback plans are in place.

Third-Party References

This guide may reference third-party frameworks, websites, or resources. ISAUnited does not endorse and is not responsible for the content, products, or services of third parties. Access is at the reader’s own risk.

Use of Normative Terms (“Shall,” “Should,” “Must”)

- Must / Shall: A mandatory requirement for conformance to the standard.
- Must Not / Shall Not: A prohibition; implementations claiming conformance shall not perform the stated action.
- Should: A strong recommendation; valid reasons may exist to deviate in particular circumstances, but the full implications must be understood and documented.

Acceptance of Terms

By using this guide, readers acknowledge and agree to the terms in this disclaimer. If you disagree, refrain from using the information provided.

For more information, please visit our Terms and Conditions page.

License & Use Permissions

The Defensible 10 Standards (D10S) are owned, governed, and maintained by the Institute of Security Architecture United (ISAUnited.org).

Practitioner & Internal Use (Allowed):

- You are free to download, share, and apply this standard for non-commercial use within your organization, departments, or for individual professional, academic, or research purposes.
- Attribution to ISAUnited.org must be maintained.
- You may not modify the document outside of Sub-Standard authorship workflows governed by ISAUnited, excluding the provided Defensible 10 Standards templates and matrices.

Commercial Use (Prohibited Without Permission):

- Commercial entities seeking to embed, integrate, redistribute, automate, or incorporate this standard in software, tooling, managed services, audit products, or commercial training must obtain a Commercial Integration License from ISAUnited.

To request permissions or licensing:
info@isaunited.org

Standards Development & Governance Notice

This standard is one of the ten Parent Standards in the Defensible 10 Standards (D10S) series. Each Parent Standard is governed by ISAUnited's Standards Committee, peer-reviewed by the ISAUnited Technical Fellow Society, and maintained in the Defensible 10 Standards GitHub repository for transparency and version control.

Contributions & Collaboration

ISAUnited maintains a public GitHub repository for standards development. Practitioners may view and clone materials, but contributions require:

- ISAUnited registration and vetting
- Approved Contributor ID
- Valid GitHub username

All Sub-Standard contributions must follow the Defensible Standards Submission Schema (D-SSF) and are peer-reviewed by the Technical Fellow Society during the annual Open Season.

Obsolete and withdrawn documents should not be used; please use replacements.

Abstract

The ISAUnited Defensible 10 Standards provide a structured, engineering-grade framework for implementing robust and measurable cybersecurity architecture and engineering practices. The guide outlines the frameworks, principles, methods, and technical specifications required to design, build, verify, and operate reliable systems.

Developed under the ISAUnited methodology, the standards align with modern enterprise realities and integrate Security by Design, continuous technical validation, and resilience-based engineering to address emerging threats. The guide is written for security architects and engineers, IT and platform practitioners, software and product teams, governance and risk professionals, and technical decision-makers seeking a defensible approach that is testable, auditable, and scalable.

This document includes a series of Practitioner Guidance, Cybersecurity Students & Early-Career Guidance, and Quick Win Playbook callouts.

Practitioner Guidance- Actionable steps and patterns to apply the technical standards in real environments.

Cybersecurity Student & Early-Career Guidance- Compact, hands-on activities that turn each section's ideas into a small, verifiable artifact.

Quick Win Playbook- Immediate, evidence-driven actions that improve posture now while reinforcing good engineering discipline.

Together, these elements help organizations translate intent into engineered outcomes and sustain long-term protection and operational integrity.

Foreword

Message from ISAUnited Leadership

Cybersecurity is at a turning point. As digital systems scale, reactive and checklist-driven practices do not keep pace with adversaries. The ISAUnited position is clear: security must be practiced as engineered design, grounded in scientific principles, structured methods, and defensible evidence. Our mission is to professionalize cybersecurity architecture and engineering with standards that are actionable, testable, and auditable.

ISAUnited Defensible 10 Standards: First Edition is a practical framework for that shift. The standards in this book are not theoretical. They translate intent into measurable specifications, controls, and verification, and enable teams to design and operate resilient systems at enterprise scale.

About This First Edition

This edition publishes 10 Parent Standards, one for each core domain of security architecture and engineering. Sub-standards will follow in subsequent editions, contributed by ISAUnited members and reviewed by our Technical Fellow Society, to provide focused, technology-aligned detail. Adopting the Parent Standards now positions organizations for seamless integration of Sub Standards as they are released on the ISAUnited annual update cycle.

Why “Defensible Standards”

Defensible means the work can withstand technical, operational, and adversarial scrutiny. These standards are designed to be demonstrated with evidence, featuring clear architecture, measurable specifications, and verification, so that practitioners can confidently stand behind their designs.

Contents

Annex G (Normative): D07-Threat & Vulnerability Security Engineering

ISAUnited's Defensible 10 Standards
Parent Standard: D07-Threat & Vulnerability Security Engineering
Document: ISAU-DS-TVE-1000
Last Revision Date: January 2026
Peer-Reviewed By: ISAUnited Technical Fellow Society
Approved By: ISAUnited Standards Committee

Section 1. Standard Introduction

Threat and Vulnerability Security Engineering (TVE) is an architectural discipline focused on identifying, assessing, and mitigating exploitable weaknesses in enterprise systems before adversaries can exploit them. Modern enterprise estates span on-premises infrastructure, multi-cloud services, hybrid connectivity, and operational technology (OT). Exposure shifts constantly. A newly deployed API route, an overly permissive identity policy, an unpatched library, or a misconfigured storage service can expand the attack surface within minutes.

TVE does not operate as a periodic scan-and-patch program. It treats vulnerability reduction and threat mitigation as a single engineering function that spans design and operations. Teams instrument systems, correlate exposure with threat pressure, prioritize work with clear decision rules, and verify closure with evidence. Continuous security validation (CSV) is part of the work, not a separate event. The goal is simple: reduce exploitable conditions early, then prove that exploit paths remain blocked as the environment changes.

This Parent Standard (ISAU-DS-TVE-1000) defines the foundation for designing, implementing, and sustaining defensible TVE capabilities. It is written for cybersecurity architects and engineers, IT and cloud engineers, software engineers, and red, blue, and purple team leads. The guidance addresses continuous vulnerability assessment, threat-informed prioritization, remediation workflows, adversary emulation, and CSV as integrated parts of enterprise architecture. It also establishes Technical Corroboration as the closure requirement: remediation is not complete until validation results and supporting artifacts demonstrate that exploitation fails in practice.

Objective

This standard defines the architectural requirements, engineering controls, and validation criteria needed to:

1. Achieve authoritative visibility: maintain ASM↔CMDB parity and internet-facing flags for all in-scope assets, including ephemeral/container/serverless.
2. Prioritize with threat context: integrate KEV/EPSS and sector intelligence into vulnerability analysis so RBR decisions are traceable and timely.
3. Remediate on enforceable SLAs: drive automated workflows that meet KEV, internet-facing mitigation targets and record exceptions with compensating controls.
4. Prove exploit-block continuously: embed CSV/BAS so remediation (patch or virtual patch) is not complete without an authenticated rescan and CSV pass.

Obsolete and withdrawn documents should not be used; please use replacements.

5. Reduce exposure by design: minimize attack surface through default-deny exposure policies, hardened baselines as code, and SoD-enforced pipelines.
6. Produce audit-ready evidence: link every change to Test-IDs and an Evidence Pack ID to support verification, audit sampling, and learning.

It covers the full lifecycle—discover, analyze, prioritize, remediate, validate, and improve—so posture remains measurable, enforceable, and adaptable as adversary behavior and enterprise architecture evolve.

Justification

Adversaries weaponize vulnerabilities quickly after disclosure, and exploit chains often combine multiple weaknesses across identity, network paths, software dependencies, and configuration state. Periodic, policy-only programs cannot match that tempo. Foundational guidance, such as NIST SP 800-40 Rev. 4 for patch management and ISO IEC 27002:2022 for control practices, remains necessary. It is insufficient on its own for complex, distributed architectures. Engineering specificity, traceable decision logic, and continuous validation are required to keep exposure bounded over time.

TVE closes this gap by using threat modeling and adversary mapping frameworks as engineering inputs. STRIDE supports design-time analysis of standard weakness classes. MITRE ATT&CK supports mapping exposure to observed TTPs and shaping telemetry and validation plans. Cyber Kill Chain style models support layering and containment across campaign phases. Target lifecycle models support reprioritization and watchlists aligned to active actor objectives relevant to the sector.

TVE also requires risk-based remediation pipelines, zero-day preparedness (virtual patching, deception, and proactive threat hunting), and CSV (BAS, red teaming, and purple teaming). These mechanisms provide Technical Corroboration: after remediation, exploit attempts must fail, and the evidence must remain valid as systems, dependencies, and configurations change.

By adopting this standard, enterprise programs equip engineering teams with clear structure and disciplined practices to detect, prioritize, and mitigate vulnerabilities at adversary speed, with proof that defenses perform as intended.

Section 2. Definitions

These definitions ensure consistent interpretation across ISAUnited members, contributors, and implementers. Terms are framed for architecture and infrastructure design, not policy operations.

Adversary Emulation – A structured testing approach that reproduces specific threat-actor behaviors (campaign objectives, TTPs, tooling) to evaluate detections, preventions, and response playbooks.

Asset Criticality – A classification of assets based on business impact, sensitivity, and mission dependence, used to weight vulnerability prioritization and remediation SLAs.

Attack Surface Management (ASM) – A continuous process of discovering, classifying, and monitoring all digital assets, services, and infrastructure components that adversaries, internal or external, could target.

ASM↔CMDB Parity – The degree of match between assets discovered by ASM and assets recorded in the CMDB; used as a visibility quality metric.

Authenticated Scanning – Vulnerability scanning performed with valid credentials or agents to enable deeper assessment of installed software, configurations, and local weaknesses.

Breach and Attack Simulation (BAS) – An automated or semi-automated security validation technique that emulates real-world adversary TTPs to test the effectiveness of security controls and defenses.

CMDB (Configuration Management Database) – The authoritative repository for configuration items and their relationships; used to govern inventory, ownership, and change.

Compensating Controls – Temporary or alternative safeguards (e.g., WAF rules, IPS signatures, access restrictions) that reduce risk when a primary remediation (e.g., a vendor patch) is not yet available or cannot be applied.

Configuration Drift – Deviation of systems from approved, hardened baselines over time due to changes, patches, or manual interventions.

Continuous Security Validation (CSV) – The engineering discipline of continuously verifying that implemented security controls perform as intended under live or simulated adversarial conditions, ensuring ongoing resilience.

Crown Jewel Assets – Systems, applications, data stores, or processes whose compromise would cause unacceptable business impact; typically receive the strictest SLOs/SLAs.

CVE (Common Vulnerabilities and Exposures) – A standardized identifier for publicly known cybersecurity vulnerabilities.

CWE (Common Weakness Enumeration) – A community-curated list of common software and hardware weakness types that can lead to vulnerabilities.

CVSS (Common Vulnerability Scoring System) – An industry scoring method for rating the severity of CVEs; useful but insufficient alone for prioritization.

EPSS (Exploit Prediction Scoring System) – A data-driven probability estimate that a CVE will be exploited in the wild, used to enhance risk-based remediation.

Ephemeral Assets – Short-lived compute resources (for example, containers, serverless functions, burst VMs) that appear and disappear quickly and must be included in discovery, scanning, and validation.

Evidence Pack (EP) – The tamper-evident collection of artifacts (plans, configs, logs, test outputs, screenshots, approvals) that proves a control or remediation was implemented and validated.

Evidence Pack ID – A unique identifier linking a change/closure to its Evidence Pack for audit and traceability.

Exploit Chain – A sequence of vulnerabilities and weaknesses that an attacker can link to escalate privileges, pivot laterally, or achieve strategic objectives.

Exposure Management (EM) – A risk-driven process of assessing, prioritizing, and mitigating vulnerabilities and misconfigurations based on exploitability, business impact, and threat intelligence.

Internet-facing – Assets or services directly reachable from untrusted networks (for example, the public internet); typically subject to the fastest detection and remediation SLO/SLAs.

KEV (Known Exploited Vulnerabilities) – A designation for CVEs that are actively exploited in the wild; often maintained in authoritative catalogs used to accelerate remediation.

Lockheed Martin Cyber Kill Chain – A model describing the phases of a targeted cyberattack—from reconnaissance to actions on objectives—used to identify detection and prevention opportunities across the campaign lifecycle.

Obsolete and withdrawn documents should not be used; please use replacements.

Mandiant Target Lifecycle – A threat-intelligence framework describing the progression of an adversary's campaign, providing insight into intrusion objectives, operational phases, and exploitation patterns.

Mean Time to Detect (MTTD) – The average elapsed time between the introduction or exposure of a vulnerability and its detection by organizational controls.

Mean Time to Remediate (MTTR) – The average elapsed time from detection to verified closure of a vulnerability, misconfiguration, or exposure.

Microsoft STRIDE – A design-time threat-modeling framework (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) used to anticipate and prevent classes of weaknesses in systems and software.

MITRE ATT&CK Framework – A globally recognized knowledge base of adversary TTPs based on real-world observations are used for threat mapping, defense gap analysis, and adversary emulation.

Patch Management – A structured engineering process for acquiring, testing, and deploying software and firmware updates to remediate vulnerabilities, enhance functionality, or maintain compliance.

Policies as Code / Controls as Code – The practice of expressing security policies and control logic (scan profiles, risk models, baselines, virtual-patch rules, CSV tests) as version-controlled code with peer review and automated validation.

Purple Teaming – A collaborative exercise where red and blue teams share context and iterate in real time to improve detections, preventions, and response playbooks.

Red Teaming – An independent, goal-oriented exercise that safely simulates adversary operations to test an organization's people, processes, and technology.

Risk-Based Remediation (RBR) – A remediation strategy that prioritizes fixes based on exploit likelihood, asset criticality, threat intelligence, and business impact rather than severity scores alone.

Separation of Duties (SoD) – A control requiring that detection (scanning/ASM), remediation (patch/config), and validation (CSV/BAS) are performed by distinct roles/pipelines/identities to prevent conflicts of interest.

Service Level Agreement (SLA) – A time-bound performance target (for example, remediation windows for actively exploited vulnerabilities) that engineering and operations teams must meet.

Service Level Objective (SLO) – A measurable performance target used to manage reliability or security quality (for example, authenticated scan coverage ≥ 95 percent, exposure alert latency ≤ 60 minutes).

Threat & Vulnerability Security Engineering (TVE) – The ISAUnited domain that engineers the continuous discovery, prioritization, remediation, and validation of exploitable exposure across on-premises, multi-cloud, SaaS, and OT/ICS environments.

Threat Intelligence (TI) Enrichment – The addition of external and internal threat context (KEV status, EPSS probability, active campaigns, ATT&CK technique tags) to findings to guide prioritization and control tuning.

Threat Intelligence Fusion – The aggregation, normalization, and correlation of threat intelligence from multiple sources to produce actionable insights for vulnerability and risk mitigation.

Unauthenticated Scanning – Vulnerability scanning performed without credentials to identify externally visible exposures and misconfigurations.

Virtual Patching – The application of network, application, or host-level rules and signatures (for example, WAF/IPS) that mitigate a vulnerability's exploitability without changing the underlying code or binaries.

Vulnerability Management – The end-to-end process of discovering, assessing, prioritizing, and remediating security vulnerabilities in systems, applications, and infrastructure to reduce organizational risk.

Zero-Day – A vulnerability that is unknown to the vendor or has no available patch at the time of discovery.

Zero-Day Preparedness – An organization's capability to detect, mitigate, and respond to unknown or undisclosed vulnerabilities before public exploits are available, leveraging proactive detection and defensive engineering.

Section 3. Scope

Threat and Vulnerability Security Engineering (TVE) includes the architectural methods, engineering workflows, and technical controls used to identify, assess, prioritize, remediate, and validate exploitable weaknesses before adversaries exploit them. Modern enterprise environments span on-premises infrastructure, multi-cloud services, software-as-a-service ecosystems, and operational technology and industrial control systems (OT and ICS). The speed and complexity of current threats require a shift from periodic scanning to continuous, threat-informed engineering.

Obsolete and withdrawn documents should not be used; please use replacements.

This Parent Standard defines the architectural expectations and technical guardrails needed to build and sustain a defensible vulnerability and threat mitigation capability. TVE is treated as a design and operations function. Teams integrate attack-surface management, vulnerability assessment, threat-intelligence correlation, risk-based remediation, and continuous security validation (CSV) into system design, delivery pipelines, and live operations. These activities are planned, instrumented, measured, and evidenced. They are not treated as afterthoughts.

A threat-informed engineering approach shortens remediation cycles, reduces exploitable exposure, and validates posture against real-world tactics, techniques, and procedures (TTPs). Technical Corroboration is within scope: closure requires evidence that exploitation fails in practice, and that the result remains valid as the environment changes.

Applicability

- **Asset types:** endpoints, servers, applications, APIs, network devices, containers, serverless workloads, data stores, cloud workloads, software as a service, and OT and ICS components.
- **Enterprise functions:** security architecture and engineering, security operations, vulnerability analysis, DevSecOps, IT and cloud engineering, and software engineering.
- **Deployment models:** on premises, private cloud, public cloud, hybrid, and edge.

Key Focus Areas

- **Attack Surface Management (ASM):** continuous discovery, classification, monitoring, and exposure reduction for reachable services and entry points.
- **Vulnerability Assessment and Risk-Based Remediation (RBR):** scanning with authenticated and unauthenticated methods, exploitability analysis, threat pressure inputs, and remediation workflows with enforceable SLAs.
- **Threat Intelligence Integration:** ingestion, normalization, correlation, and operational use to guide prioritization and defensive design decisions.
- **Continuous Security Validation (CSV):** breach and attack simulation (BAS), red team and purple team exercises, and controlled exploit testing to verify control performance.
- **Zero Day Preparedness:** rapid containment and mitigation through virtual patching, deception, targeted hardening, and compensating controls when vendor patches are not available.

- **Adversary Mapping:** use of design-time threat modeling frameworks, ATT&CK style technique mapping, kill chain style campaign phasing, and target lifecycle models to inform detection, containment, and eradication planning.

Evidence and Measurability

TVE scope requires measurable artifacts, including:

- Asset and attack surface inventories, coverage maps, scan configurations, and coverage reports.
- Correlation artifacts that show how exposure is mapped to adversary techniques, active exploitation signals, and reprioritization decisions.
- Remediation workflows with SLA metrics, including time to detect, time to remediate, and exception handling outcomes.
- CSV and BAS outputs that demonstrate exploit block verification and the absence of residual exposure under defined test conditions.
- Evidence packs that link requirements to controls as code to test outcomes, with traceability suitable for audit and internal review.

Outcomes

TVE implementations within this scope are:

- **Defensible:** bounded by clear architectural scope, validated against adversary models, and reinforced by measurable controls and evidence.
- **Measurable:** demonstrated through continuous monitoring, repeatable testing, and independent verification artifacts.
- **Adaptive:** able to evolve with TTP changes, newly disclosed vulnerabilities, and architectural shifts.
- **Aligned:** consistent with organizational risk appetite, business priorities, and recognized best practices for vulnerability and threat management.

This scope establishes the foundation for an enterprise-wide TVE capability that reduces exposure, accelerates remediation, and strengthens resilience against modern threats.

Section 4. Use Case

Resilient threat and vulnerability defense requires more than policy statements and vulnerability scanners. It requires practiced application in real enterprise environments where identity, network paths, APIs, and dependencies change daily. The use case below reflects a hybrid organization operating across on-premises infrastructure, multi-cloud services, and OT and ICS segments. It surfaces common exposure patterns, links

them to adversary behavior models, and maps weaknesses to targeted technical defenses using Attack Surface Management (ASM), Risk-Based Remediation (RBR), and Continuous Security Validation (CSV).

The goal is an engineering playbook that ties day-to-day TVE actions, including discovery, prioritization, remediation, and validation, to measurable reductions in exploitability and time at risk. Technical Corroboration is explicit: closure requires evidence that the exploit path fails in practice, not only that a patch was applied.

Table G 1:

Use Case Name	Proactive vulnerability mitigation under active exploitation conditions through threat-informed engineering
Objective	Detect, prioritize, and mitigate an actively exploited vulnerability across a hybrid enterprise by integrating threat signals, ASM, RBR, and CSV. Minimize exposure time, prevent exploitation, and produce evidence suitable for internal review and audit.
Scenario	A financial services enterprise operating on-premises and across multiple cloud environments receives an advisory about an actively exploited vulnerability affecting a widely deployed application server component. Threat reporting indicates sector targeting, observed exploitation in the wild, and known-exploited-vulnerability status in a widely referenced exploitation catalog. ASM and internal scanning identify hundreds of reachable instances on vulnerable builds, including services reachable from the public internet. Historical remediation performance averages 30–45 days for critical issues.
Actors	Threat intelligence analyst; vulnerability management engineer; SOC analyst; red team lead; purple team lead; DevSecOps engineer; infrastructure security architect; change and release coordinator; OT site lead when OT and ICS segments are in scope.
Adversary Mapping	Design time: Focus areas for the STRIDE method relevant to the scenario, including tampering, elevation of privilege, and information disclosure. Technique mapping examples: exploit of a public-facing application; command and script execution; exploitation of remote services; valid account abuse; ingress tool transfer. Campaign phasing: delivery; exploitation; installation; command and control; actions on objectives. Target lifecycle view: sector-specific objectives and campaign watchlists used to adjust prioritization and validation focus.

Challenges Identified	Delayed patch deployment; incomplete cross-environment visibility; limited coverage of ephemeral workloads; CVSS only prioritization without threat pressure; weak proof of remediation effectiveness; constrained OT maintenance windows.
Technical Solution	1) Attack Surface Discovery and Prioritization Maintain continuous ASM. Classify assets by reachability, business criticality, and data sensitivity. Cover ephemeral workloads using orchestration inventory, image registries, and deployment records, not only host-based scans. Prioritize public, internet-reachable instances and systems on crown-jewel paths. 2) Threat Signal Integration Ingest multiple threat sources and internal telemetry. Record active exploitation signals, exploit availability indicators, and sector targeting signals. Apply EPSS as an exploitability signal input when appropriate. Map the vulnerability to technique patterns relevant to exploitation and post-compromise behavior. Maintain a sector watchlist to drive reprioritization as threat pressure changes. 3) Risk-Based Remediation (RBR) Define enforceable mitigation targets for active exploitation conditions. Example targets: public internet-reachable and actively exploited; mitigated within 48 hours via patching, configuration changes, or virtual patching; internal high-critical mitigated within 7 days; medium mitigated within 30 days. Automate mitigation through controlled workflows, including virtual patching rules, safe configuration changes, package updates, and gated release pipelines with rollback testing. Require authenticated reassessment after mitigation and enforce drift prevention through policy-as-code. 4) Continuous Security Validation (CSV) Execute BAS scenarios or controlled exploit tests against defined exploit paths. The red team validates that the exploit path fails and that lateral movement routes are no longer viable. Purple team tunes detection logic and telemetry coverage based on validation results, with explicit false-negative reduction targets. 5) Containment and Compensating Controls Apply temporary network containment when needed, including firewall restrictions and exposure reduction. Enforce least-privileged service identities and reduce the blast radius through segmentation and scoped access policies. Increase telemetry and add deception triggers on high-value routes to quickly detect exploitation attempts. 6) OT and ICS Safety Use read-only discovery methods. Apply mitigations approved for the environment and validate in a testbed when feasible before production changes. Schedule changes within maintenance windows and document compensating controls used to bridge timing constraints.

Expected Outcome (Targets)	≥ 80% reduction in time at risk for public internet-reachable vulnerable instances within 48 hours through remediation or compensating mitigations. 100% inventory coverage of vulnerable instances is demonstrated by ASM–CMDB parity outputs and orchestration and registry coverage records for ephemeral workloads. Prioritization decisions reflect threat pressure inputs, asset criticality, and reachability, not CVSS alone. CSV outputs demonstrate unsuccessful exploit attempts after mitigation, and no viable lateral movement under defined test conditions. No unplanned service disruption, with a change success rate ≥ 99 %, supported by tested rollback procedures.
Evidence Artifacts	ASM exports, coverage maps, and parity records; asset inventory reconciliation outputs, including orchestration and registry evidence for ephemeral workloads; threat correlation records showing technique mapping, active exploitation signals, and reprioritization decisions; ticket and change records with timestamps; mitigation diffs for virtual patching and configuration changes; authenticated reassessment results; CSV and BAS reports showing before and after results; detection tuning notes; OT and ICS testbed validation records when applicable. Evidence Pack IDs: EP-07.1, EP-07.2, EP-07.3, EP-07.4, EP-07.5

Key Takeaways

- Treat threat and vulnerability work as an engineered system: define scope boundaries, instrument coverage, and require closure evidence through EP-07.x artifacts.
- Replace periodic scanning with continuous exposure visibility: reconcile ASM to the asset inventory, and prove coverage parity for internet reachable services and crown jewel paths.
- Prioritize by threat pressure, not severity alone: combine reachability, asset criticality, exploit availability indicators, and technique mapping to drive mitigation targets and change windows.
- Make remediation deterministic: use controlled workflows with gated promotion, tested rollback, and authenticated reassessment before closure.
- Require Technical Corroboration for closure: validate exploit block with BAS or controlled exploit tests, then revalidate after material change events such as configuration drift, dependency updates, and infrastructure redeployments.
- Engineer for zero-day conditions: pre-plan compensating controls, including virtual patching, exposure reduction, and segmented blast radius containment when patches are not available.

- Protect constrained environments explicitly: for OT and ICS segments, use read-only discovery, approved mitigations, testbed validation when feasible, and documented compensating controls during maintenance windows.
- Preserve defensibility through evidence discipline: store logs, correlation outputs, validation results, and change records in a tamper-resistant evidence repository aligned to EP-07.x.

Practitioner Guidance:

- Validate authoritative visibility first: reconcile ASM and the asset inventory, then close coverage gaps before tuning prioritization rules.
- Enforce threat-informed mitigation targets for active exploitation conditions: require authenticated reassessment and CSV pass criteria before closure.
- Automate as code: scan profiles, baselines, virtual patching rules, and remediation workflows should be version-controlled, peer reviewed, and delivered through gated pipelines with rollback testing.
- Tune detections through purple team workflows: use CSV outputs to reduce false negatives and confirm detection depth for mapped techniques.
- Protect safety-critical environments: for OT and ICS segments, use read-only discovery, approved mitigations, and testbeds when feasible prior to production changes.

Section 5. Requirements (Inputs)

To implement Threat and Vulnerability Security Engineering (TVE), the following baseline architectural, operational, and environmental conditions MUST be met. These inputs ensure conformance with the Technical Specifications in Section 6 and the aligned sub-standards. They MUST exist before teams design, implement, or validate TVE controls. Each input MUST have an owner, an operational status, and a proof reference aligned to the Evidence Pack system (EP-07.x).

5.1 Authoritative Asset and Attack Surface Inventory

An authoritative, continuously updated inventory MUST exist for endpoints, servers, network devices, applications, APIs, containers, serverless functions, cloud workloads, software-as-a-service services, data stores, and OT and ICS systems. Assets MUST be tagged with business-criticality, data sensitivity, ownership, environment, and reachability. ASM outputs and the system of record for asset inventory MUST be kept in automated synchronization, and sync integrity MUST be tested on a defined cadence.

5.2 Continuous Vulnerability Assessment Coverage

Obsolete and withdrawn documents should not be used; please use replacements.

Scanning MUST operate across all in-scope asset classes using agent-based and agentless methods, with authenticated and unauthenticated modes where feasible. Coverage MUST include ephemeral workloads through orchestration inventory, image registries, and deployment records, not only host-based scans. Scan windows, rates, and safe operating limits MUST be engineered, approved, and documented. Coverage reports and automated gap alerts MUST be produced.

5.3 Threat Intelligence Integration and Correlation
Threat signals MUST be ingested from multiple sources, normalized, and correlated to vulnerabilities and exposed services. CVEs MUST be mapped to technique models, and to sector-relevant campaigns when such information is available. Known Exploited Vulnerability Indicators (KEV) and EPSS should be used as inputs for exploitability pressure. The source, freshness, and confidence of threat signals MUST be recorded.

5.4 Risk-Based Remediation Pipeline and SLAs
Prioritization MUST incorporate exploitability pressure inputs, asset criticality, reachability, and business impact. CVSS MUST not be the sole driver of prioritization or closure. SLAs MUST be defined, enforced, and measured. The workflow MUST integrate with change and release management, and it MUST include exception tracking with time bounds and compensating controls.

5.5 Continuous Security Validation Capability
A continuous security validation capability MUST exist using approved validation playbooks and controlled test methods. Validation activities MUST include adversary emulation workflows, controlled exploit-path testing, and confirmation of detection efficacy. Remediation items MUST remain open until exploit block and detection outcomes meet defined pass criteria. Failed validations MUST trigger rollback, compensating controls, or risk acceptance with time-bound revalidation.

5.6 Patch and Configuration Management
Centralized patch and configuration management MUST be available for in-scope assets. Hardened baselines MUST be defined and managed as versioned artifacts, using recognized benchmarks such as CIS Benchmarks where applicable. Post-change verification and drift detection MUST be enforced. Rollback procedures MUST be documented, tested, and auditable.

5.7 Centralized Telemetry and Correlation
Events from vulnerability assessment, ASM, patch workflows, and validation activities MUST be centralized in a correlation platform suitable for security operations. Parsing and normalization MUST be tested, and rule logic MUST be validated for expected inputs and failure behavior. Correlation logic MUST detect

newly exploitable paths and MUST support verification of control performance. Time synchronization MUST be maintained across sources.

5.8 Incident Response and Containment Playbooks
TVE detections and active-exploitation signals MUST be linked to incident response runbooks that include pre-approved containment actions. Containment actions may include reducing exposure, applying virtual patches, imposing temporary network restrictions, and isolating services. Automated or semi-automated execution should be available for active exploitation conditions, with appropriate change control and safety safeguards.

5.9 Least Privilege Identities and Secrets Hygiene for TVE Tooling
Dedicated service identities MUST exist for scanning, validation, automation, and integration functions. These identities MUST be least privileged, time-bound where feasible, and subject to monitoring. Credentials, tokens, and keys MUST be vaulted, rotated, and auditable. Tool-to-tool trust MUST be established explicitly using strong authentication and integrity controls appropriate to the environment.

5.10 CI and CD Integration and Release Evidence
Delivery pipelines should include security gates relevant to TVE outcomes, such as policy checks, dependency vulnerability checks, secret detection, deployment posture checks, and, where applicable, post-deployment validation hooks. Pipeline artifacts MUST be versioned and linked to release identifiers, including configurations, results, approvals, and validation outcomes.

5.11 Software Supply Chain Signals
SBOM and provenance signals should be ingested and maintained for in-scope software components. Advisories and ecosystem alerts MUST be monitored. Transitive dependency exposure MUST inform risk scoring, mitigation planning, and validation scope.

5.12 Exception Management and Compensating Controls
Risk acceptances MUST be time-bound and require documented justification, documented compensating controls, and scheduled revalidation. Exceptions MUST be tracked in the same system of record as remediation, and exception closure MUST require verification that controls remain effective within the approved time window.

5.13 Staging, Testbeds, and OT and ICS Safety
Staging environments or testbeds should be used to validate patches and mitigations before production deployment for high-risk changes. For OT and ICS segments, discovery MUST default to read-only methods. Mitigations MUST respect safety constraints, and changes MUST be scheduled within approved windows and rehearsed when feasible.

5.14 Metrics Ownership and Targets

Owners and targets MUST be defined for time to detect, time to remediate, SLA compliance, validation pass rates, and inventory and coverage percentages. Dashboards MUST be published to engineering and risk governance forums. Variance from targets MUST trigger corrective action plans with defined timelines and accountable owners.

Additional Architectural Prerequisites (supporting §6)

5.15 Policy, Configuration, and Detection as Code

Scan profiles, baselines, virtual patching rules, and detection logic should be managed as versioned artifacts. Changes MUST be peer reviewed, tested, and deployed through controlled workflows. Artifact integrity should be protected through signing or equivalent integrity controls.

5.16 Authenticated and Unauthenticated Modes and Safe Windows

Scanning modes and rates MUST be tuned per asset class, and safe windows MUST be defined to prevent service degradation. Negative testing should verify scanner behavior on authentication failures and network errors. If coverage is reduced due to safety constraints, compensating controls and alternate assessment methods MUST be documented.

5.17 Virtual Patching and Edge Enforcement

Virtual patching capability MUST exist to deploy compensating mitigations for active exploitation conditions when patches are not immediately available. Mitigations may include request filtering, protocol restrictions, exposure reduction, egress restrictions, and rate controls. Virtual patching MUST be validated through CSV before remediation closure.

5.18 Adversary Mapping Services

Services or workflows MUST exist to maintain mappings between vulnerabilities, technique models, campaign phasing models, and sector-relevant watchlists. These mappings MUST be available for risk scoring, validation planning, and detection engineering.

5.19 Tamper-Evident Evidence Repositories

Logs, scan configurations, change records, correlation outputs, validation results, and approvals MUST be stored in tamper-evident repositories with integrity verification and retention aligned to validation and verification needs and audit expectations.

Practitioner Guidance:

- Validate visibility first: reconcile ASM outputs to the asset inventory system of record, then close coverage gaps before tuning prioritization rules.
- Make prioritization threat-informed: use known exploited vulnerability indicators, EPSS, reachability, and asset criticality to drive SLAs and change windows. Do not rely on CVSS alone.
- Automate as versioned artifacts: manage scan profiles, baselines, virtual patching rules, and remediation workflows through controlled delivery paths with peer review and tested rollback.
- Treat remediation as unverified until validation passes: require authenticated reassessment, exploit block confirmation, and detection efficacy checks, then record closure evidence.
- Protect operations: respect maintenance windows, service objectives, and OT and ICS safety constraints. Use testbeds, phased rollout, and pre-approved compensating controls to avoid disruption.

Section 6. Technical Specifications (Outputs)

Technical specifications define the concrete, defensible outputs that must be implemented to satisfy this Parent Standard. Each output is a required engineering area that transforms policy into measurable, auditable security outcomes across on-premises, multi-cloud, hybrid, and OT/ICS environments.

Traceability note: Each item references representative Inputs (§5.x) it depends on and produces Evidence consumed by §12 V&V.

Outputs must be:

- **Measurable:** validated by scans, logs, audits, or tests
- **Actionable:** implementation-ready, not policy slogans
- **Aligned:** traceable to §5 Requirements and sub-standards

6.1 Asset and Attack Surface Management (ASM)

Objective. Maintain authoritative visibility into assets and exposed services. Remove blind spots, including ephemeral workloads and OT and ICS environments.

- **Continuous asset discovery:** Maintain automated discovery for endpoints, servers, network devices, applications, APIs, containers, serverless functions, cloud workloads, software as a service services, data stores, and OT and ICS systems. For ephemeral assets, include orchestration inventory, registry data, and deployment records as authoritative sources.
- **Internet reachable and internal surface mapping:** Maintain service maps for reachable ports, protocols, endpoints, and identity exposure points.

Obsolete and withdrawn documents should not be used; please use replacements.

Capture reachability changes driven by routing, policy, and configuration changes.

- **Asset criticality tagging:** Tag assets for business impact, data sensitivity, environment, and ownership. Explicitly identify crown jewel systems and crown jewel paths.
- **Exposure change alerting:** Detect and alert on new services, unauthorized deployments, and reachability expansion. Target MTTD ≤ 60 minutes for internet reachable exposure and ≤ 4 hours for internal exposure.
- **Inventory reconciliation:** Reconcile ASM outputs to the asset inventory system of record on a defined cadence. Resolve discovery deltas within 24 hours.
- **Coverage integrity checks:** Test discovery pipelines for failure behavior. If discovery data is stale or incomplete, trigger gap alerts and do not treat coverage reports as authoritative until reconciled.

6.2 Vulnerability Assessment and Risk-Based Remediation (RBR)

Objective. Comprehensively detect vulnerabilities and remediate them in line with threat-informed mitigation targets.

- **Vulnerability assessment coverage:** Run agent-based and agentless assessments using authenticated and unauthenticated methods where feasible. Define safe windows and tuned rates per asset class to avoid service degradation.
- **Ephemeral workload assessment:** Assess images and deployments through registries and orchestration records, not only host scans. When runtime scanning is infeasible, document alternate assessment methods and compensating controls.
- **Threat signal correlation:** Enrich findings with technique mapping, exploit availability indicators, active exploitation signals, and EPSS where appropriate. Record the source, freshness, and confidence of threat inputs.
- **Risk scoring beyond CVSS:** Maintain a documented risk scoring method that includes exploitability pressure, reachability, asset criticality, lateral movement potential, and business impact. Calibrate the method at least quarterly and preserve decision trails for sampled cases.
- **Mitigation targets and enforcement:** Define mitigation targets that reflect threat pressure and reachability. Example targets include internet-reachable and actively exploited, mitigated within 48 hours; internal high within 7 days; and medium within 30 days. Track exceptions as time-bound risk acceptances with compensating controls and scheduled revalidation.
- **Virtual patching and compensating controls:** Maintain the ability to reduce exploitability before patch availability through request filtering, exposure reduction, protocol restrictions, throttling, egress restrictions, and host rules. Closure requires proof that exploit paths fail under defined test conditions.
- **Closure discipline:** Remediation is not complete when a change is applied. Closure requires reassessment and validation results showing an exploit block for the defined exploit path.

6.3 Threat Intelligence and Adversary Simulation

Objective. Use adversary behavior to drive prioritization and prove defensive performance.

- **Multi-source ingestion:** Ingest threat signals from multiple sources, normalize formats, and de-duplicate indicators. Track feed health and ingestion latency. Target median ingest to enrichment latency ≤ 30 minutes for high-priority signals.
- **Campaign watchlists:** Maintain sector-relevant watchlists using a campaign lifecycle view. Link watchlists to crown jewel paths and high-risk exposure classes. Reprioritize within 24 hours when watchlists change for relevant technologies or services.
- **Technique mapping:** Map exposure and vulnerability classes to technique models to guide validation planning, detection engineering, and containment strategy.
- **Breach and attack simulation or controlled exploit testing:** Run scheduled and on-demand tests that validate mitigation effectiveness for patched and virtually patched paths. Validate closure of internet-reachable, actively exploited findings within 7 days.
- **Red and purple team exercises:** Conduct adversary emulation aligned to campaign phases. Use purple teaming to tune detections, reduce false negatives, and improve telemetry fidelity.
- **Tuning workflow discipline:** Manage tuning changes through versioned, peer-reviewed workflows with tested rollback for material changes.

6.4 Continuous Security Validation (CSV) and Control Effectiveness

Objective. Treat remediation as incomplete until the exploit block is proven, and keep it proven as systems change.

- **Automated validation checks:** Run continuous checks for defined high-risk scopes that validate patches, configurations, compensating controls, and detection outcomes. Target daily checks for high-risk scopes.
- **Exploit block verification:** Validate exploit block for patched and virtually patched assets using controlled exploit path tests or BAS scenarios. Re-test after material change events, including configuration updates, dependency updates, and infrastructure redeployments.
- **Configuration drift detection:** Monitor baseline deviations and respond within defined service objectives. Target drift MTTD ≤ 4 hours. Where auto revert is used, target revert success ≥ 95 % and preserve evidence of revert outcomes and exceptions.
- **Traceability mapping:** Maintain a traceability mapping between controls, required inputs, technique mappings, and covered vulnerability classes. Update on a defined cadence and store in the evidence repository used for V and V.

- **Validation reporting:** Produce structured validation reports with pass and fail results, residual risk, and corrective actions. Track corrective actions to closure.

6.5 Patch and Configuration Management Integration

Objective. Patch quickly, safely, and verifiably, with engineered rollback and OT and ICS safety constraints.

- **Patch deployment workflow:** Maintain centralized deployment with an emergency mitigation channel for active exploitation conditions. Use staged rollout patterns with recorded health checks.
- **Baseline enforcement:** Apply hardened baselines as versioned artifacts across system classes. Track exceptions and revalidate exceptions on schedule.
- **Post change verification:** Validate installation success and service stability using configuration checks and functional checks. Use authenticated reassessment where feasible.
- **Rollback discipline:** Maintain documented rollback procedures per platform. Execute rollback drills at least twice per year. Target rollback success ≥ 99 % in drills, with RTO aligned to system criticality.
- **Change alignment:** Track remediation actions through change and release workflows. Link mitigation, validation outcomes, and closure decisions to the evidence repository used for V and V. Orphan remediation changes are not acceptable.

Practitioner Guidance:

- Establish visibility first. Do not prioritize until ASM parity and assessment coverage are proven.
- Prioritize using threat pressure and reachability. CVSS alone does not represent exploitability in context.
- Automate as versioned artifacts. Scan profiles, baselines, mitigation rules, and workflows belong under peer review and controlled delivery.
- Validate early, then keep validating. Closure requires an exploit block proof, and drift monitoring keeps that proof up to date.
- Coordinate across domains. TVE depends on telemetry, identity controls, and segmentation. Keep owners and handoffs explicit.

Quick Win Playbook:

Title: Fail Closed Remediation Closure Gate for One Internet Reachable Service

Objective: Implement a repeatable closure gate for one high-value, internet-reachable service so that vulnerability remediation cannot be closed until:

1. reassessment confirms the vulnerable condition no longer exists, and
2. Validation confirms the exploit path fails under defined test conditions.

This playbook establishes the minimum engineering loop for Threat and Vulnerability Security Engineering: visibility, prioritization, remediation, and proof.

Target: Require a closure gate for one internet reachable service so remediation cannot close without reassessment and exploit block validation (6.1, 6.2, 6.4).

Component and System: ASM, vulnerability assessment workflow, change workflow, compensating control point, validation runner, evidence repository.

Protects: Public-facing service paths from rapid exploitation of newly disclosed vulnerabilities, and prevents false closure based on patch status alone.

Stops and Detects: Untracked exposed instances, unauthenticated assessment gaps, closure without proof, and drift that reopens exposure.

Action:

1. Select one high-value internet-reachable service and define its scope, owners, and crown jewel dependencies.
2. Reconcile ASM to the asset inventory system of record for this scope, then close parity gaps.
3. Require authenticated assessment for eligible assets and document mode exceptions with compensating controls.
4. Define a mitigation target for active exploitation conditions and pre-approve compensating controls for the scope.
5. Implement a closure gate: ticket closure requires reassessment, plus validation results demonstrating the exploit block for the defined exploit path.
6. Add a drift trigger: configuration change or redeploy requires retest before prior closure remains valid.

Proof: Parity reconciliation outputs, assessment coverage reports, change records, and mitigation diffs, validation outputs linked to closure, retest evidence for material change events.

Metric: 100 % of scoped remediation closures include reassessment and exploit block validation. 0 closures without linked proof artifacts.

	Rollback: Remove the closure gate only through time-bound exception handling. Preserve the superseded workflow artifacts for audit continuity.

Section 7. Cybersecurity Core Principles

The following ISAUnited Cybersecurity Core Principles are foundational to the design, implementation, and ongoing management of a Defensible Threat & Vulnerability Security Engineering (TVE) program. These principles guide architectural decisions, technical specifications, and operational practices to ensure that threat detection, vulnerability remediation, and attack-surface reduction are resilient, measurable, and engineered to withstand real-world adversaries.

Purpose and Function

Security principles are not slogans; they are operating constraints that enforce discipline, clarity, and foresight. Grounding technical specifications and implementation strategies in these principles ensures sub-standards under this domain deliver a long-term, engineering-based defense posture rather than one-off reactions.

Table G-2: Principles and TVE-Domain Applicability:

Principle Name	Code	Applicability to Threat & Vulnerability Security Engineering
Least Privilege	ISAU-RP-01	Scanning, CSV/BAS, and remediation automations operate with only required permissions to reduce abuse of agents, scripts, and privileged interfaces.
Zero Trust	ISAU-RP-02	All inputs used for prioritization and automation (scan results, TI, KEV/EPSS) are authenticated, authorized, and integrity-checked before use.
Defense in Depth	ISAU-RP-04	Layered hardening, allowlists, WAF/IPS, EDR, segmentation, and identity controls jointly prevent exploitation and lateral movement.
Secure by Design	ISAU-RP-05	

Principle Name	Code	Applicability to Threat & Vulnerability Security Engineering
		Detection, TI fusion, risk-based remediation, and CSV are integrated into architectures and pipelines from the start.
Minimize Attack Surface	ISAU-RP-06	Unnecessary services, network exposure, and legacy components are continuously removed to reduce exploitable attack surfaces.
Evidence Production	ISAU-RP-15	Every scan, exploit test, remediation, and validation produces immutable, audit-ready evidence for Verification and Validation (V&V) and learning.
Make Compromise Detection Easier	ISAU-RP-16	TVE telemetry and alerts are engineered for high-fidelity, rapid triage of exploited or at-risk assets.
Protect Integrity	ISAU-RP-19	Scan results, threat intelligence data, and remediation artifacts are protected against tampering to ensure trustworthy decisions.
Protect Availability	ISAU-RP-20	TVE capabilities are highly available, so detection, prioritization, and remediation continue during incidents or outages.

***Note on traceability**: A future engineering matrix will map each principle to its associated technical outputs (§6), security controls (§9), and V&V requirements (§12), providing complete principle→specification→evidence traceability.*

Practitioner Guidance:

Embed these principles directly into change requests, code reviews, and evidence packs. Any control or process that cannot produce evidence (RP-15) or operate safely under degraded conditions (RP-20) does not meet TVE requirements. Treat violations of RP-01 or RP-02 as design defects, not operational exceptions.

Section 8. Foundational Standards Alignment

Internationally recognized frameworks from NIST and ISO/IEC establish baseline expectations for vulnerability handling, security assessment, patch/configuration management, and systems security engineering. Threat & Vulnerability Security Engineering (TVE) builds on these foundations, integrating them into a defensible, engineering-focused model that addresses hybrid architectures, adversary-informed prioritization, and measurable implementation and validation.

Purpose and Function

- Demonstrate alignment with globally accepted NIST/ISO practices for vulnerability handling, assessment, remediation, and continuous monitoring.
- Bridge compliance baselines to ISAUnited's architecture-and-engineering methodology (ASM, RBR, CSV, patch/config baselines, TI fusion).
- Enhance credibility and traceability for adoption and audit-readiness.
- Provide a consistent baseline for clause-level mapping in sub-standards.

Table G-3. Applicable Foundational Standards:

Framework	Standard ID	Reference focus
NIST	SP 800-40	Enterprise patch management processes and risk-based deployment.
NIST	SP 800-53 Rev. 5	Security and privacy controls relevant to TVE (e.g., RA, SI, CM, CA).
NIST	SP 800-115	Technical security assessment methods, including vulnerability scanning and penetration testing.
NIST	SP 800-137	Information Security Continuous Monitoring (ISCM) for ongoing control effectiveness.
NIST	SP 800-30 Rev. 1	Risk assessment methodology to inform prioritization and remediation.
NIST	SP 800-61 Rev. 2	Computer Security Incident Handling Guide for linking TVE detections to IR/containment.

Obsolete and withdrawn documents should not be used; please use replacements.

Framework	Standard ID	Reference focus
NIST	SP 800-160 Vol. 1	Systems Security Engineering practices to design and verify trustworthy TVE capabilities.
NIST	SP 800-82	OT/ICS security practices for safe discovery, change windows, and control validation in industrial environments.
ISO/IEC	27001:2022	ISMS requirements encompass vulnerability and change control within risk management.
ISO/IEC	27002:2022	Code of practice for implementing controls for vulnerability management and logging/monitoring.
ISO/IEC	30111	Vulnerability handling processes (receive, analyze, remediate, coordinate).
ISO/IEC	29147	Vulnerability disclosure practices for coordinated reporting and advisories.
ISO/IEC	27035 (series)	Information security incident management aligned to TVE-driven containment actions.
ISO/IEC	27005	Information security risk management supporting RBR scoring, prioritization, and exception rationale.
ISO/IEC	27004	Measurement and metrics for program performance (coverage %, MTTD/MTTR, SLA/CSV rates) and evidence quality.

NOTE: ISAUnited Charter Adoption of Foundational Standards.

Per the ISAUnited Charter, the institute formally adopts the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) and the National Institute of Standards and Technology (NIST) as its foundational standards bodies, consistent with their public encouragement of organizational adoption. Parent Standards align with ISO/IEC and NIST for architectural grounding and auditability, and this alignment cascades down to Sub-Standards as invariant, minimum requirements that may be tightened but not weakened. ISAUnited does not restate or speak on behalf of ISO/IEC or NIST; practitioners shall consult the official publications and terminology

of these organizations, verify scope and version currency against the latest materials, and implement controls in a manner consistent with ISAUnited security invariants and the requirements of this standard.

Sub-Standard Expectations

Sub-standards under ISAU-DS-TVE-1000 must:

- Cite specific clauses from Table G-3 (e.g., NIST SP 800-40; NIST SP 800-53 SI/RA; ISO/IEC 30111/29147) for each normative output they extend.
- Convert those clauses into testable engineering behaviors (policy-as-code / control-as-code) with defined verification/validation in §12.
- Document any divergence with compensating controls, a risk-based rationale, and a sunset date; store passing artifacts under the Evidence Pack ID.
- Include a concise mapping table: §6 Output → Framework → Clause → Test-ID(s) → Evidence Pack ID.

Practitioner Guidance:

- Map at the clause level only. For each §6 output (e.g., 6.2 RBR, 6.3 TI & Adversary Simulation, 6.4 CSV, 6.5 Patch/Config), add a row: Spec → NIST/ISO clause → how enforced (policy/code) → Evidence Pack ID.
- Keep mappings current. When any control or policy changes (SLA targets, risk model inputs, scan modes, CSV gates), update the NIST/ISO citation in the same change and store the diff in the Evidence Pack.
- In multi-regime environments, adopt the strictest applicable clause. Record the rationale once in the mapping sheet.
- Maintain scope discipline. Do not list ATT&CK/CSA/CIS here; place them in §9 with their testable control mappings.

Section 9. Security Controls

This section identifies the technical control families and external control references directly supported or enforced by the Threat & Vulnerability Security Engineering (TVE) Parent Standard. These mappings translate architectural intent into actionable safeguards and ensure traceability, auditability, and consistency across enterprise environments.

Purpose and Function

Security controls convert this Parent Standard's requirements into measurable, testable safeguards. The mappings below reinforce timely vulnerability detection, threat-intel fusion, risk-based remediation (RBR), and continuous security validation (CSV).

Implementation Guidance

Authors and practitioners should:

- Reference specific, implementation-level controls from authoritative frameworks.
- Provide framework acronym, control ID, and concise description tied to §6 outputs and §7 principles.
- Prefer concrete technical safeguards over policy abstractions.

Table G-4. Control Mappings for Threat & Vulnerability Security Engineering:

Framework	Control ID	Control Name / Description
CSA CCM	TVM-01	Threat & Vulnerability Management — Continuous identification, assessment, and prioritization of vulnerabilities across all assets (agent/agentless; authenticated/unauthenticated; includes ephemeral/container/serverless).
CSA CCM	TVM-02	Threat & Vulnerability Management — Integrate real-time threat intelligence into vulnerability workflows to enable RBR and SLA enforcement.
CSA CCM	SEF-02	Segregation of Duties — Separate scanning, remediation, and validation (CSV/BAS) functions to preserve control integrity and avoid conflicts of interest.
CIS Controls v8	7.1	Establish & Maintain a Vulnerability Management Process — Documented program with roles, cadence, SLAs, exception handling, and evidence capture.
CIS Controls v8	7.3	Remediate Detected Vulnerabilities — Prioritize and address them using exploit likelihood, asset criticality, and exposure context (not just CVSS).
CIS Controls v8	7.6	Perform Authenticated Vulnerability Scans of Internal Assets — Authenticated scanning at defined intervals with coverage reporting and gap alerts.
OWASP ASVS	V14.2	Security Architecture — Ensure vulnerability processes include threat modeling and mapping of exposures to attacker behaviors to prioritize and place controls.

Framework	Control ID	Control Name / Description
OWASP ASVS	V14.3	Security Architecture — Conduct regular penetration testing and security reviews to validate that remediation and compensating controls are effective (CSV evidence).

***NOTE**: NIST and ISO/IEC are Foundational Standards in §8. Use CSA/CIS/OWASP here in §9 for control implementation. Adversary-technique mapping (e.g., ATT&CK) belongs in §12 and sub-standards' test plans.*

NOTE: Use of External Control Frameworks.

ISAUnited maps to external control frameworks to provide alignment and traceability, but does not speak on behalf of those organizations. Practitioners shall consult and follow the official practices, recommendations, and implementation guidance of the Center for Internet Security (CIS), the Cloud Security Alliance (CSA), and the Open Worldwide Application Security Project (OWASP) when applying controls. Always verify control identifiers, scope, and version currency against the publishers' latest materials. Where wording differs, use the framework's official documentation while maintaining consistency with ISAUnited security invariants and this standard's requirements.

Sub-Standard Expectations

Sub-standards developed under the TVE Parent Standard must:

- Select and enforce explicit technical controls relevant to their focus (e.g., ASM, RBR, zero-day preparedness, CSV).
- Provide clause-level mapping tables: §6 Output → Framework → Control/Technique → Test-ID(s) → Evidence Pack ID.
- Justify and document any deviation from the Parent-level control families, including compensating controls and a sunset date.

Practitioner Guidance:

- Map once, enforce forever. For each §6 output, add a row: Spec → Framework (CSA/CIS/OWASP) → Control ID → How enforced (policy/code) → Test-ID(s) → Evidence Pack ID—and keep it beside the code that implements it.

	• Treat controls as code. Keep scan profiles, SoD rules, CSV test suites, and WAF/IPS patterns in version control with peer review and signed releases; link change tickets to control IDs. • Prove coverage, not intent. Maintain dashboards for scan coverage (authenticated vs unauthenticated), remediation SLA compliance, CSV pass rates, and exception counts; review them on a fixed cadence. • Close the loop with CSV. Remediation isn't complete until a CSV/pen-test aligned with OWASP ASVS V14.3 passes, and before- and after results are attached to the Evidence Pack.

Section 10. Engineering Discipline

This section defines the architectural thinking, rigorous engineering processes, and disciplined operational behaviors required to implement Threat & Vulnerability Security Engineering (ISAU-DS-TVE-1000). ISAUnited's Defensible Standards are engineered systems, grounded in systems thinking, critical reasoning, and Verification & Validation (V&V), that produce measurable, auditable, defensible outcomes across discovery, prioritization, remediation, and continuous validation.

10.1 Purpose & Function

Purpose. Establish a repeatable, auditable way of working that integrates systems thinking, lifecycle controls, adversary-aware design, and measurable outcomes for threat and vulnerability engineering.

Function in D10S. Parent Standards set expectations and invariants. Sub-Standards convert them into policies-as-code/controls-as-code, test specifications, and evidence artifacts embedded in delivery and operations.

10.2 Systems Thinking

Goal: Make the TVE system legible end-to-end, boundaries, flows, interfaces, and dependencies, so controls bind where risk actually manifests.

10.2.1 System Definition & Boundaries

- Declare system purpose, scope, stakeholders, and in/out-of-scope assets (ASM, VM scanners, image/registry scanners, ticket/change system, patch/config platforms, BAS/CSV, SIEM/telemetry, TI ingestion, CMDB, evidence store; OT/ICS where applicable).
- Model trust zones and boundary crossings (internet/extranet → exposed services; admin/workload → control planes; CI/CD →

environments; scanner/BAS → targets; TI feeds → correlation; IR runbooks → containment).

10.2.2 Interfaces & TVE Contracts

- Maintain Interface Control Documents (ICDs) for discovery, scanning, prioritization, remediation, and validation paths.
- For each interface specify: principal type (human vs service identity), required privileges, scan modes (auth/unauth), safe windows/rates, risk model inputs (KEV/EPSS/criticality), SLA tiers, rollback criteria, telemetry fields (asset_id, exposure_id, finding_id, risk_score, internet_facing_flag, evidence_pack_id), retention/time-sync requirements, and invariants ("KEV, internet-facing ≤ 48 hours," "closure requires CSV pass," "TI ingestion fails closed").

10.2.3 Dependencies & Emergent Behavior

- Map shared services (time sync/NTP, vault/keys, SIEM, orchestrators/registries, CI/CD, evidence store).
- Identify emergent risks from composition (for example, unauthenticated scans + permissive allowlists → blind spots; long change queues + no virtual patching → extended exposure; single operator controlling scan + fix + verify → SoD failure; image drift + missing promotion gates → reintroduced vulnerabilities).

10.2.4 Failure Modes & Safeguards

- For critical paths, document failure modes (feed outage, scan auth failure, enrichment mismatch, risk model error, patch rollback, CSV failure, drift reintroductions) and safeguards (deny-by-default exposure policy, negative tests, virtual patching, SoD enforcement, immutable logging, scheduled failover with no fail open).

Required Artifacts (minimum): context diagram with trust boundaries; TVE flow map (discover → prioritize → remediate → validate); ICD set; invariants register.

10.3 Critical Thinking

Goal: Replace assumptions with explicit reasoning that survives review, attack, and audit.

10.3.1 Decision Discipline

- **Use Architecture Decision Records (ADRs)**: problem → options → constraints/assumptions → trade-offs → decision → invariants → test/evidence plan (who/when/how measured).

10.3.2 Engineering Prompts

- **Boundaries**: What are the exposure and control boundaries, and why? Where do ASM, scanning, and CSV sit?

- **Interfaces**: What must always be true at each TVE interface (invariants)? How is it tested (positive/negative)?
- **Adversary pressure**: Which exploit paths are credible here (public-facing, remote services, lateral movement)? What is the shortest attack path, and how is it interrupted?
- **Evidence**: Which objective signals prove this control works today and after change (coverage, SLA timestamps, CSV pass, drift checks)?
- **Failure**: When this fails, does it fail safe (deny/contain, virtual patch, immutable log)? What is the operator's next action?

Required Artifacts (minimum): ADRs; assumptions and constraints log; evidence plan per decision.

10.4 Domain-Wide Engineering Expectations

Secure System Design

- Define TVE boundaries (ASM/VM, TI fusion, risk model, patch/config, CSV/BAS, SIEM, evidence store; OT/ICS specifics).
- Validate boundaries and trust relationships via structured reviews using §10.2 artifacts; ensure protections bind to SLA tiers, risk model inputs, and privilege boundaries at each hop.

Implementation Philosophy — "Built-in, not bolted-on."

- Integrate ASM, TI enrichment, RBR, virtual patching, and CSV at design time.
- Express controls as policy-as-code/control-as-code bound to §10.2.4 invariants ("KEV, internet-facing ≤ 48 hours," "closure requires CSV pass," "TI ingestion fails closed").

Lifecycle Integration

- Embed TVE controls into design review, backlog, build/test, deploy, and operations; keep delivery mechanics in Annex J.
- Enforce version-controlled reviews with required ADRs and Evidence Pack ID updates on every change.

Verification Rigor (V&V)

- Combine automated checks (coverage SLOs, enrichment/latency tests, risk model unit tests, CSV suites, drift detection) with targeted probes (virtual patch bypass, rollback drills, noise injection).
- Require continuous validation in pipelines and scheduled runtime checks tied to invariants (for example, coverage %, SLA attainment, CSV pass rate, drift MTTD). Run CSV regression after any material change to scanning, risk scoring, virtual patching, or baselines.

Operational Discipline

- Monitor for drift and unauthorized change (policy diffs, disabled scans, stale TI feeds, SLA breaches, SoD violations); auto-remediate where safe with time-bounded exceptions.
- Maintain runbooks/SOPs for KEV events, high-risk findings, rollback failure, CSV failures, and OT/ICS windows; record outcomes in the Evidence Pack.

10.5 Engineering Implementation Expectations

- Policies/Controls as Code. Manage scan profiles, exposure rules, risk scoring, baselines, WAF/IPS rules, and CSV suites as code with peer review and provenance.
- Structured Remediation Path. Build → validate risk model → canary → staged rollout with health gates → promote/rollback (execution detail in Annex J; semantics here).
- Automated Security Testing. Integrate negative tests for TI ingest (fails closed), SLA timers, virtual patch efficacy, CSV regression after changes, and drift auto-revert assertions before production.
- Explicit Coverage Mapping. Maintain diagrams/metrics for discovery/scan coverage (auth/unauth), internet-facing flags, remediation steps, CSV checkpoints, including ephemeral/container/serverless.
- Traceable Architecture Decisions. Link ADRs to controls, tests, and evidence; update ADRs and evidence on each change request.

Required Artifacts (minimum): policies-as-code repo; enforcement/test gates; boundary/ICD set; automated test results; evidence ledger (see §10.7 and §12).

10.6 Sub-Standard Alignment (inheritance rules)

Sub-Standards must operationalize this discipline with TVE-specific detail:

- Attack Surface Management (for example, ISAU-DS-TVE-1010). Continuous automated discovery, exposure classification logic, version-controlled detection policies; Tests: exposure alert latency, coverage SLOs, CMDB parity.
- Patch & Secure Baselines (for example, ISAU-DS-TVE-1020). SLA tiers, emergency channels, baseline-as-code, rollback drills; Tests: KEV, internet-facing ≤ 48 hours, compliance snapshots, rollback success.
- Threat Intelligence & Risk-Based Prioritization (for example, ISAU-DS-TVE-1030). Multi-source ingest, KEV/EPSS use, scoring function; Tests: enrichment latency, scoring traceability, reprioritization within 24 hours.
- CSV & Adversary Simulation (for example, ISAU-DS-TVE-1040). BAS/CSV suites, pass criteria, regression after change; Tests: exploit-block verification, detection tuning outcomes.
- Exposure Management & Zero-Day Preparedness (for example, ISAU-DS-TVE-1050). Virtual patching patterns, containment playbooks, deception where appropriate; Tests: virtual patch efficacy, containment MTTR.

Obsolete and withdrawn documents should not be used; please use replacements.

10.7 Evidence & V&V (what proves it works)

Establish a TVE Evidence Pack per system containing:

- Design Evidence: boundary/flow diagrams; ICDs; invariants register; ADRs.
- Build Evidence: policies-as-code history (scan profiles, risk model, baselines, WAF/IPS); ingest/latency tests; CSV suite definitions; CI outcomes.
- Operate Evidence: coverage dashboards (auth/unauth split), TI correlation logs (including KEV/EPSS flags), SLA timestamps, virtual-patch diffs, authenticated rescans, CSV/BAS before/after reports, drift alerts/reverts, SIEM correlations.
- Challenge Evidence: targeted probes (virtual patch bypass), red/purple outcomes, rollback drills, incident timelines with automated containment, remediation closure with re-test.

Each control requires objective pass/fail criteria, specified test frequency, a responsible owner, and a defined retention policy. Map Evidence Pack IDs into §12 traceability.

10.8 Example: Sub-Standard Discipline Alignment (Risk-Based Remediation)

Scope: ISAU-DS-TVE-1030 (Threat Intelligence & Risk-Based Prioritization).
Design: Define invariants (“KEV, internet-facing ≤ 48 hours,” “closure requires CSV pass,” “TI ingestion fails closed”). Place enrichment, scoring, and SLA timers on the path.
Implement: Express risk scoring, SLA tiers, and exception policy as code; integrate TI feeds; emit traceable decisions; link changes to Evidence Pack ID.
V&V: Unit-test scoring function; measure ingest-to-enrich latency; run BAS/CSV on patched and virtually patched items; assert denial of closure without CSV pass; run weekly regression.
Operate: Evidence Pack includes scoring model diffs, enrichment logs, SLA dashboards, CSV results, and rollback/exception records.

Practitioner Guidance:

- Maintain a living Controls → Outputs → Tests sheet per TVE scope; update it in the same change that modifies policies or pipelines, and attach proofs (coverage reports, enrichment logs, CSV results, diffs).
- Favor controls expressed as code and verified automatically by §12 tests; reserve exceptions for time-bounded, owner-approved waivers with compensating controls and explicit Test-IDs/Evidence Pack IDs.

Section 11. Associate Sub-Standards Mapping

Purpose of Sub-Standards

ISAUnited Defensible Sub-Standards are detailed, domain-specific extensions of the Threat & Vulnerability Security Engineering Parent Standard (ISAU-DS-TVE-1000).

Each Sub-Standard delivers:

- Granular technical guidance tailored to a specific TVE focus area (e.g., ASM, RBR, CSV).
- Actionable implementation strategies that translate architectural intent into operational controls and validation procedures.
- Precise verification methodologies ensuring outputs are measurable, auditable, and defensible under adversary pressure.
- Alignment with ISAUnited core principles and the Technical Specifications of the Parent Standard (§6).

Sub-Standards bridge the gap between the Parent's architectural expectations and the detailed engineering required for robust discovery, prioritization, remediation, and validation across on-premises, multi-cloud, SaaS, and OT/ICS environments.

Scope and Focus of Threat & Vulnerability Sub-Standards

Automated Vulnerability Scanning & Attack Surface Reduction
Identifier (example): ISAU-DS-TVE-1010

- Authenticated and unauthenticated scanning requirements for internal, external, ephemeral/container/serverless assets.
- Continuous attack surface monitoring with asset classification and prioritization.
- Alerting thresholds for new exposures and unauthorized service changes.
- Mandatory integration with asset inventory/CMDB for coverage parity.

Patch Management & Secure Configuration Baselines
Identifier (example): ISAU-DS-TVE-1020

- SLA-based patching windows for critical/high/medium classes (KEV/EPSS-aware).
- Hardened baseline configurations (e.g., CIS/STIG) as code, with drift detection.
- Post-patch verification must include authenticated rescans before closure.
- Automated post-patch verification and validated rollback procedures.
- Version-controlled configuration management with change evidence.

Threat Intelligence & Adaptive Risk-Based Prioritization
Identifier (example): ISAU-DS-TVE-1030

- Multi-source TI ingestion/correlation into VM workflows.

- CVE mapping to adversary models (e.g., ATT&CK, Mandiant Target Lifecycle, Lockheed Cyber Kill Chain) for prioritization and detection engineering.
- Scoring model incorporating exploit likelihood (KEV/EPSS), asset criticality, exposure context, and sector activity.
- Continuous re-prioritization based on live intelligence.

Red Teaming, Penetration Testing & Continuous Security Validation (CSV)
Identifier (example): ISAU-DS-TVE-1040

- Annual red teaming and targeted quarterly **penetration testing** requirements.
- Continuous BAS to validate exploit-block for patched and virtually patched paths.
- Exploit validation playbooks tied to remediation closure.
- Independent verification for high-severity vulnerability closures.

Exposure Management & Zero-Day Preparedness
Identifier (example): ISAU-DS-TVE-1050

- Architecture for proactive zero-day mitigation (virtual patching, containment patterns).
- Deception technologies (honeypots/decoys) for early detection where appropriate.
- Continuous monitoring of vendor advisories and TI alerts.
- Integrated IR workflows for high-priority zero-day events.

Table G-5. Example Sub-Standards:

Identifier	Sub-Standard Name	Focus Area
ISAU-DS-TVE-1010	Automated Vulnerability Scanning & Attack Surface Reduction	Continuous Scanning & ASM
ISAU-DS-TVE-1020	Patch Management & Secure Configuration Baselines	Patching & Baselines
ISAU-DS-TVE-1030	Threat Intelligence & Adaptive Risk-Based Prioritization	Threat-Aligned Remediation
ISAU-DS-TVE-1040	Red Teaming, Pen Testing & Continuous Security Validation (CSV)	CSV & Adversary Simulation
ISAU-DS-TVE-1050	Exposure Management & Zero-Day Preparedness	Zero-Day Defense & EM

Note: Future identifiers under TVE continue the 1xxx series to maintain consistency with ISAUnited numbering.

Development and Approval Process

ISAUnited uses an open, peer-driven annual process to propose, review, and publish sub-standards:

- Open Season Submission: Contributors submit proposed sub-standards aligned to ISAU-DS-TVE-1000 objectives.
- Technical Peer Review: The Technical Fellow Society evaluates accuracy, validity, and applicability.
- Approval & Publication: Approved sub-standards receive formal versioning and publication as authoritative extensions of ISAU-DS-TVE-1000.

Sub-Standard Deliverables (normative)

Each sub-standard must include:

- **Inputs (Requirements):** Preconditions from Annex F §5 it depends on.
- **Outputs (Specifications):** Concrete identity-layer behaviors and thresholds (for example, AAL targets, token TTL/rotation, JIT windows) tied to §6.
- **Verification/Validation:** Named tests and acceptance criteria tied to §12 (for example, replay denial, elevation denial without approval, certification closure).
- **Evidence:** Artifact list and storage location (EP-06.xx).
- **Standards Mapping:** Spec → NIST/ISO clause (§8) → Controls (§9) → Test-ID (§12) → Evidence Pack ID.
- **Interfaces:** Clear delineation of what is enforced at IdP/STS/PDP/PEP/PAM (Annex F) vs. delivery mechanics (Annex J) and crypto parameters (Annex I).

Practitioner Guidance:

- Start with a one-page map. For each sub-standard, create a single sheet: Scope → §5 Inputs → §6 Outputs → Test-IDs (§12) → Evidence Pack ID. Build from this to prevent drift and guesswork.
- Bind every output to a test and owner. No output without a named Test-ID, pass/fail criteria, cadence, a responsible owner, and a destination for results in the Evidence Pack.
- Treat controls as code with traceability. Keep scan profiles, risk scoring, baselines, virtual-patch rules, and CSV suites in version control; require peer review; link commits to the ISAU-DS-TVE-1xxx ID and the Evidence Pack entry.

	• Publish coverage SLOs and show trend. For each sub-standard, surface a small set of SLOs (e.g., authenticated scan %, ASM↔CMDB parity, SLA compliance, CSV pass rate) and a rolling 90-day trend dashboard. • Enforce exception hygiene. Any SLA/control exception must include compensating controls, a re-validation date, an owner, and a Test-ID—recorded with the sub-standard's ID in the same system of record.

Section 12. Verification and Validation (Tests)

This section defines the structured evaluation methods that prove Threat & Vulnerability Security Engineering (TVE) controls, architecture, and operations align with the intent of this Parent Standard. It mandates measurable, repeatable procedures so implementations are technically defensible and fully consistent with ISAUnited's engineering discipline.

Verification confirms that the capabilities were implemented in accordance with §5 Requirements (Inputs) and §6 Technical Specifications (Outputs).

Validation demonstrates that the capabilities perform under real-world conditions, withstand adversary testing, and remain resilient as threats and environments evolve.

Core Verification Activities

- Confirm all §6 outputs are deployed and configured in the target environment with coverage against the declared scope (on-premises, multi-cloud, SaaS, OT/ICS).
- Review hardened baselines and safe scan windows for scanners, patch/config platforms, ASM, and TI ingestion; compare configurations to engineering benchmarks (e.g., CIS baselines; NIST SP 800-40 process expectations).
- Verify system integration paths (e.g., ASM → VM → RBR → Patch/Config → CSV/BAS → SIEM) have no fail-open states and preserve data integrity, identity, and timing.
- Conduct peer review of diagrams, workflows, SLA logic, and scoring models; ensure traceability from requirement → control → test → evidence.

Core Validation Activities

- Execute adversary-informed testing (targeted pen-tests, BAS, red/purple team) to validate exploitability and confirm exploit-block after remediation or virtual patching.
- Exercise the RBR pipeline on KEV/EPSS-pressured items and measure detection-to-closure against SLAs; verify exception handling and compensating controls.
- Validate alignment with adversary models (e.g., MITRE ATT&CK technique simulators), relevant campaign phases (Mandiant Target Lifecycle), and interruption points (Kill Chain).
- Assess operational resilience via fault drills (scanner outage, TI feed loss, patch rollback) to confirm continuity, fail-closed behavior, and recovery times.
- Measure performance against defined metrics, including:
  - MTTD for newly introduced vulnerabilities/exposures.
  - MTTR to verified closure by severity/SLA tier.
  - SLA compliance rate for remediation and validation.
  - CSV pass rate and percentage of fixes re-tested and confirmed closed.

Required Deliverables

All Verification & Validation efforts must produce documented outputs that include:

1. Test Plans & Procedures — Scope, cases, data sets, simulators/tools, positive/negative criteria, and safety constraints (esp. OT/ICS).
2. Validation Reports — Results, pass/fail, residual risk, reprioritization outcomes, and ranked backlog of unremediated items.
3. Evidence Artifacts — Logs, scan configs and results, enrichment records, scoring decisions, CSV/BAS outputs, screenshots, and ticket/change links.
4. Corrective Action Plans (CAPs) — Remediation steps, owners, deadlines, exception records, and follow-up test IDs.

Common Pitfalls to Avoid

- Closing without proof - Remediation (patch or virtual patch) marked “done” without an authenticated rescan and a CSV/BAS exploit-block pass attached to an Evidence Pack ID.
- Testing success only - Skipping negative tests (e.g., feed outage, scanner auth failure, policy rollback) that prove the system fails closed and maintains continuity.
- Unauthenticated-only verification - Re-testing with unauthenticated scans that miss local/config weaknesses—use authenticated modes for closure evidence.
- No regression after change - Failing to rerun CSV/BAS upon material changes (signatures, risk model weights, baselines, virtual patch rules).

Obsolete and withdrawn documents should not be used; please use replacements.

- Threat-blind validation - V&V not incorporating KEV/EPSS and sector campaigns; results do not reflect current adversary pressure.
- Evidence gaps - Test outputs are not linked to Test-IDs and an Evidence Pack; artifacts lack timestamps, hashes, or chain-of-custody.
- Scope blind spots - Ephemeral/container/serverless or OT/ICS assets excluded from V&V; staging not representative of production.
- Uncalibrated SLA metrics - MTTD/MTTR/SLA/CSV metrics undefined, unowned, or not reviewed—no CAPs triggered on breach.
- Integration not exercised - Validating components in isolation but skipping end-to-end paths (ASM → VM → RBR → Patch/Config → CSV/BAS → SIEM).
- SoD collapse during testing - Same identity/pipeline conducts scan, remediation, and validation—undermines trust in the results.
- Foundational drift - Baselines, safe windows, or TI ingest settings changed without V&V updates or ETM refresh.
- OT/ICS safety not proven - Running invasive tests in production OT or skipping testbed validation and vendor-approved procedures.

Table G-6. Traceability Matrix — Requirements (§5) → V&V (§12) → Related Technical Specs (§6):

Requirement ID	Requirement (summary)	Verification (build-correct)	Validation (works-right)	Related Technical Specs
5.1	Comprehensive asset & attack-surface inventory	ASM jobs cover all environments; CMDB sync verified; asset criticality and internet-facing tags present	Targeted scans locate all asset classes; surprise asset drops are detected within SLA	§6.1 Asset & ASM
5.2	Continuous vulnerability assessment capability	Authenticated/unauthenticated modes configured (agent/agentless); safe windows approved; coverage reports generated	Re-scan proves parity on critical classes; cadence targets met for internet-facing and high-critical assets	§6.2 Vulnerability Assessment & RBR
5.3	Threat-intel integration framework	TI feeds ingested/normalized; enrichment rules active; KEV/EPSS flags present	Active-exploit CVEs auto-prioritized; simulations emulate current actor techniques to confirm prioritization efficacy	§6.2; §6.3 TI & Adversary Simulation
5.4	Risk-based remediation (RBR) pipeline	Risk model implemented;		§6.2; §6.5 Patch &

Requirement ID	Requirement (summary)	Verification (build-correct)	Validation (works-right)	Related Technical Specs
		ticketing/change linkage active; SLA tiers defined and visible	Detection-to-closure meets SLA for KEV/highs; exceptions documented with compensating controls and retest date	Config Mgmt
5.5	CSV capability	BAS platform configured; red/purple cadence defined; approved test playbooks	BAS/pen tests confirm exploits blocked post-remediation (incl. virtual patch); no residual lateral movement	§6.3; §6.4 CSV & Control Effectiveness
5.6	Patch & configuration management infrastructure	Patch channels, baselines, compliance monitors configured; rollback procedures documented/tested	Emergency patches deploy within SLA; post-patch re-tests clean; rollbacks succeed when invoked	§6.5 Patch & Config Mgmt
5.7	Centralized logging & SIEM correlation	VM/ASM/patch/CSV events centralized; parsers/dashboards verified; time sync in place	Correlations detect newly exploitable paths within MTTD target; reports satisfy audit sampling	§6.4 (reporting); §6.5 (post-patch logs)
5.8	IR & containment workflow integration	Runbooks link high-risk CVEs to IR playbooks; WAF/IPS controls registered for virtual patching	Live drills show containment executed within MTTR target; service impact minimized and documented	§6.2 (virtual patching); §6.3 (adaptive tuning); §6.4 (validation)
5.9	Least-privilege identities & secrets hygiene (TVE tooling)	Dedicated scanner/CSV/automation identities exist; scopes reviewed; secrets vaulted/rotated with audit trails	Attempts outside scoped roles are denied; key/secret rotation does not	§6.2; §6.4; §6.5

Requirement ID	Requirement (summary)	Verification (build-correct)	Validation (works-right)	Related Technical Specs
			break scanning/CSV	
5.10	CI/CD integration & release evidence	Pipeline gates for policies/controls-as-code, coverage checks, CSV tests, and evidence capture are enabled	Changes without passing gates are blocked; post-deploy CSV regression passes, and artifacts are attached to the ticket	§6.2; §6.4; §6.5
5.11	Software supply-chain inputs (SBOM/provenance)	SBOM/provenance ingested; approved registries/artifacts enforced; 3rd-party plugins quarantined until verified	Unsigned/unknown artifacts denied at admission; signed image update proceeds and CSV smoke passes	§6.1; §6.2; §6.4
5.12	Exception & compensating control process	Exceptions are time-bounded with the owner and compensating controls; records link to tickets	Exceptions expire on schedule or are re-approved; CSV proves compensating control efficacy until closure	§6.2; §6.4; §6.5
5.13	Staging/testbeds & OT/ICS safety	Read-only discovery for OT; vendor-approved procedures; testbed validations documented	Safe rollout in OT window succeeds; drills show no service impact; recovery meets stated RTO/RPO	§6.1; §6.5
5.14	Metrics ownership & targets	Owners and SLO/SLA targets set for coverage, MTTD, MTTR, CSV pass rate; dashboards live	Weekly reviews meet thresholds or trigger CAPs; trends improve QoQ; breaches auto-escalate	§6.1; §6.2; §6.4; §6.5
5.15	Policy/Configuration/Detection as Code	Repos exist; protected branches/CODEOWNERS; CI lint/unit tests for scan profiles, risk model,	Promotion of signed bundles changes controls as intended;	§6.1; §6.2; §6.4; §6.5

Requirement ID	Requirement (summary)	Verification (build-correct)	Validation (works-right)	Related Technical Specs
		baselines, CSV suites; signed artifacts	rollback succeeds and is evidenced	
5.16	Auth/unauth modes & safe windows	Modes set per asset class; rate limits/safe windows documented; negative tests present; approvals “as code.”	Scans run within windows without SLO breach; auth failures alert within target; ingest/auth errors fail closed	§6.2; §6.1
5.17	Virtual patching & edge enforcement	WAF/IPS/edge rules cataloged; KEV playbooks defined; linkage to tickets/SLA timers	BAS confirms exploit-block until vendor patch; closure requires rescan + CSV pass; no residual lateral movement	§6.2; §6.4; §6.5
5.18	Adversary mapping services	TI enrichment emits ATT&CK/KEV/EPSS fields; sector watchlists are active; reprioritization rules are deployed	Active campaigns trigger reprioritization within 24h; detections tuned via purple teaming with improved hit rate	§6.3; §6.2
5.19	Tamper-evident evidence repositories	WORM/append-only enabled; hash verification/time sync configured; access controls and audit trails verified	Independent review reproduces findings from the Evidence Pack; integrity checks pass; chain of custody intact	§6.4; §6.5

How to use the matrix

- **Plan**: For each §5 requirement, schedule ≥1 Verification and ≥1 Validation activity and link to a §6 spec.
- **Execute**: Run the activities and record an Evidence Pack ID for each row.
- **Maintain**: When requirements, controls, or specs change, update tests/evidence and re-run CSV/BAS as applicable.

Practitioner Guidance:

- Bind every fix to proof. A remediation (patch or virtual patch) is not complete until a CSV/BAS test aligned to the risk scenario passes, and evidence is attached to the ticket.
- Measure what matters. Track and review weekly coverage %, MTTD, MTTR, SLA compliance, CSV pass rate, and exception count; set thresholds and escalate on breaches.
- Test the failure, not just the success. Include negative tests (feed outage, scanner auth failure, patch rollback) to verify fail closed behavior and continuity.
- Keep tests with the code. Store plans, suites, and results with policies/controls-as-code; require a Test-ID and an Evidence Pack ID for every change.

Quick Win Playbook:

Title: Authenticated Scanning Coverage Closeout

Objective: Achieve and sustain authenticated scanning coverage of at least 95 percent for eligible in-scope assets, eliminating blind spots and producing audit-ready evidence for §12 Tests.

Target: Close the authenticated scanning gap to ≥95 percent coverage across in-scope assets (§6.2).

Component/System: VM scanners (agent/agentless), credential vaults, orchestrators/registries (containers, serverless), and change/ticket system.

Protects: Depth of detection for local and configuration vulnerabilities missed by unauthenticated scans.

Stops/Detects: Blind spots from credential failures, missed agents, and unenrolled ephemeral/container/serverless nodes.

Action: Onboard service credentials via vault; enable and verify authenticated modes; enumerate eligible assets (record justified ineligible cases); schedule authenticated rescans in safe windows; alert on authentication failures; include ephemeral/container/serverless targets via orchestrator/registry discovery; re-test after secrets rotation.

Proof: Coverage report (authenticated vs unauthenticated) + vault role/config and rotation log + orchestrator/registry inventory snapshot + authenticated rescan diffs + alert samples for auth-failure events → Evidence Pack EP-07.01.

	Metric: ≥95 percent authenticated coverage for eligible assets; authentication-failure MTTD ≤ 24 hours; authenticated coverage trend improving week over week; zero unexplained gaps in the eligible set. **Rollback**: Temporarily revert specific authenticated scopes causing instability; document exception, compensating controls, owner, and re-validation date in EP-07.01.

Section 13. Implementation Guidelines

This section does not prescribe vendor-specific tactics. Parent Standards are stable, long-lived architectural foundations. Here, we define how sub-standards and delivery teams must translate the Parent's intent (ISAU-DS-TVE-1000) into operational behaviors that are testable, automatable, and auditable for Threat & Vulnerability Security Engineering (TVE). Delivery mechanics (pipeline orchestration, artifact signing/attestation, promotion/rollback) are governed by Annex J.

Purpose of This Section in Sub-Standards

Sub-standards must use Implementation Guidelines to:

- Translate Parent expectations into enforceable TVE behaviors (e.g., authenticated scan coverage SLOs, KEV/EPSS-aware SLA gates, virtual-patching patterns, CSV exploit-block proofs, SoD enforcement).
- Provide stack-agnostic practices that improve adoption, reduce failure, and align with ISAUnited's defensible design philosophy.
- Highlight common failure modes and how to prevent them with measurable gates and checks.
- Offer repeatable patterns (as code) that enforce controls, risk models, and engineering discipline across ASM, vulnerability scanning, TI fusion, RBR pipelines, patch/config management, CSV/BAS, SIEM, and evidence repositories.

Open Season Guidance for Contributors

Contributors developing sub-standards Must:

- Align all guidance with the Parent's strategic posture and §6 outputs (e.g., KEV internet-facing ≤ 48 hours; closure requires CSV pass; "no fail-open" on TI ingest; SoD between scan/remediate/validate).

Obsolete and withdrawn documents should not be used; please use replacements.

- Avoid vendor/product terms; express controls as requirements, tests, and evidence with an Evidence Pack ID.
- Include lessons learned (what fails, why, and how the test proves it).
- Focus on repeatable engineering patterns (policies-as-code / controls-as-code), not one-offs.
- Provide a minimal Standards Mapping: Spec/Control → NIST/ISO clause from §8 → Evidence Pack ID (keep CSA/CIS/OWASP mapping in §9).

Technical Guidance

A. Organizing Principles (normative)

1. Everything as Code - Scan profiles, target scopes, safe windows/rates, TI enrichment rules, risk-scoring functions (KEV/EPSS/criticality), SLA tiers, virtual-patch policies, admission policies (images/registries), CSV/BAS suites, promotion gates, and SIEM parsers Must be version-controlled, peer-reviewed, and released on protected branches.
2. Non-bypassable Security Gates - Every merge/release Must pass gates tied to §6 and §12, e.g.:
   - Authenticated scan coverage ≥ 95% of eligible assets;
   - Internet-facing exposure MTTD ≤ 60 min;
   - KEV internet-facing mitigation ≤ 48 h;
   - Closure requires authenticated rescan + CSV exploit-block pass;
   - TI ingestion fails-closed
   - SoD checks pass (distinct identities/pipelines).
3. Immutable, Reproducible Releases - No manual policy changes post-build. Risk models, rules, and signatures are pinned and signed; deployment verifies integrity at enforcement points.
4. Least Privilege & SoD (TVE context) - Scanners, CSV, and remediation automations use scoped identities and separate pipelines. Secrets are vaulted/rotated. SoD violations are alertable and release-blocking.
5. Environment Parity - Staging mirrors production for scan modes, TI enrichment, risk scoring, SLA gates, virtual-patch policies, and CSV suites so tests are predictive; drift is monitored and reconciled.

B. Guardrails by Pipeline Stage (normative)

1. **Pre-commit / local**
   • Signed commits; secrets scanning.
   • Lint scan profiles/risk models; reject CVSS-only scoring; require KEV/EPSS inputs present.
   • Generate CSV test stubs for any new remediation rule.
2. **Pull request (PR) / code review**
   • CODEOWNERS approval for changes to scopes, SLAs, or risk weights.
   • Coverage gate on changed asset classes (authenticated vs unauthenticated split); critical findings = 0.

Obsolete and withdrawn documents should not be used; please use replacements.

• TI enrichment diff must show KEV/EPSS/linkage; PR includes planned §12 Test-IDs and Evidence Pack ID stub.

3. **Build & package**
   • Deterministic artifacts (pinned rule bundles, signed risk model, scan profiles).
   • Package CSV/BAS suites that correspond to changed controls.
4. **Pre-deploy / release**
   • Drift check against approved policies; approvals “as code.”
   • Progressive rollout (canary/staged) for remediation and virtual-patch bundles with health SLOs and automatic rollback.
   • Positive/negative tests: coverage, KEV gates, SLA timers, CSV exploit-block, TI fail-closed behavior.
5. **Deploy & runtime**
   • Enforce SLA timers and exception rules; block closure without rescan+CSV pass.
   • Auto-block unapproved new exposures from ASM until linked to a change record.
   • SIEM correlation and SoD monitors are continuously active.
6. **Post-deploy validation & operations**
   • Continuous validation: authenticated re-scans, CSV regressions after change, KEV replay tests, exposure latency checks.
   • Track Security SLOs: authenticated coverage ≥ 95%; exposure MTTD ≤ 60 min; KEV mitigation ≤ 48 h; CSV pass rate = 100% for closures; exceptions ≤ 5% and time-bounded.
   • Auto-generate a TVE Evidence Pack per release (policy diffs, validation results, SLA timers, CSV/BAS outcomes, exception records, ADR links).

C. Identity, Access, and Secrets (normative alignment to §6.1–§6.5)

- Dedicated identities for scanners/CSV/automation; mTLS/signed tokens between components; secrets injected via approved services with audit trails.
- Deterministic error/deny semantics (no fail-open); telemetry includes asset_id/finding_id/risk_score/trace_id/policy_version/time.

D. TVE Supply-Chain Integrity (normative; mechanics in Annex J)

- Only deploy signed policy/risk bundles and CSV suites whose tests passed gates; restrict artifact sources/namespaces.
- Quarantine third-party scanner plugins/signatures until verified; enforce integrity and license checks.
- Separate build/deploy identities; forbid production writes from build jobs; treat rule/matrix tamper as release-blocking.

E. Measurement & Acceptance (aligned to §6 and §12)

- **Visibility**: ASM↔CMDB parity ≥ 99%; exposure alert latency ≤ 60 min (internet-facing).
- **Assessment**: Authenticated scan coverage ≥ 95% of eligible assets; cadence met by class.

Obsolete and withdrawn documents should not be used; please use replacements.

- **Prioritization**: KEV/EPSS enrichment present for 100% high/critical; risk decisions traceable.
- **Remediation**: KEV internet-facing mitigation ≤ 48 h; SLA compliance ≥ 95%; exceptions time-bounded with compensating controls.
- **Validation**: 100% closures have authenticated rescan + CSV pass; regression run after material change.
- **Evidence**: Every change links §5 → §6 → §12 via an Evidence Pack ID.

Common Pitfalls (and the engineered countermeasure)

1. CVSS-only prioritization - Countermeasure: enforce KEV/EPSS presence and weighting in risk model gates; reject builds missing threat context.
2. Unauthenticated-only scanning - Countermeasure: authenticated coverage SLOs with alerts on auth failure; block release if coverage < target.
3. Closing without proof - Countermeasure: release gate that denies ticket closure due to missing an authenticated rescan, CSV pass, and Evidence Pack link.
4. New exposures shipped by accident - Countermeasure: ASM-to-change linkage; auto-block unapproved exposures; require CSV reachability proof post-approval.
5. SoD collapse - Countermeasure: distinct pipelines/identities for scan vs remediate and validate; SoD monitor that blocks if any single identity holds all three roles.

Practitioner Guidance:

- Codify the gates that matter. Put authenticated coverage, KEV/EPSS gates, SLA timers, and CSV exploit-block pass into checks that cannot be bypassed in the pipeline.
- Bind every closure to proof. Do not close remediation without an authenticated rescan, a CSV exploit-block pass, and an Evidence Pack ID.
- Watch four SLOs weekly. Coverage percentage, exposure MTTD, KEV/SLA attainment, and CSV pass rate—trend them and escalate on breach.
- Keep SoD real. Separate people, identities, and pipelines; alert on violations and treat them as release blocking.

Quick Win Playbook:

Title: Auto-Block New Internet-Facing Exposures

	Objective: Prevent unapproved internet-facing services from reaching production by enforcing change linkage and CSV reachability tests before allowing, with auditable evidence for §12 Tests. **Target**: Auto-block unapproved new internet-facing exposures and require change linkage + CSV reachability before allowing (§6.1, §6.4). **Component/System**: ASM platform, change/ticket system, perimeter ACLs or reverse proxy, SIEM. **Protects**: Prevents accidental exposure and shadow IT from reaching production. Stops/Detects: Unapproved listeners/ports, rogue deployments, and test endpoints promoted to production. **Action**: Wire ASM “new exposure” events to the change system; if no matching change ticket exists, push an automatic deny rule at the edge and page the owner; once linked to an approved change, require CSV reachability and negative tests, then lift the block. **Proof**: ASM alert + change-mismatch log + auto-block rule diff + CSV reachability and negative-test results → Evidence Pack EP-07.02. **Metric**: ≥ 90 percent of new exposures blocked within 60 minutes; 100 percent of exposures mapped to an approved change or remediated; CSV reachability conformance = 100 percent before unblock. **Rollback**: Lift the temporary block when the approved change is verified; retain artifacts as superseded in the Evidence Pack.

Appendices

Appendix A: Engineering Traceability Matrix (ETM)

Req ID	Requirement (Inputs) (§5)	Technical Specifications (Outputs) (§6)	Core Principles (§7)	Control Mappings (§9)	Verification – Build Correct (§12)	Validation – Works Right (§12)	Evidence Pack ID
5.1	Comprehensive asset and attack-surface inventory	§6.1 Asset & ASM	RP-05 Secure by Design; RP-06 Minimize Attack Surface; RP-15 Evidence Production	CSA TVM-01; CIS 7.1	ASM jobs cover all environments; CMDB sync verified; criticality and internet-facing tags present	Targeted scans find all classes; surprise asset drops are detected within SLA	EP-07.04
5.2	Continuous vulnerability assessment capability	§6.2 Vulnerability Assessment & RBR	RP-02 Zero Trust; RP-04 Defense in Depth; RP-16 Make Compromise Detection Easier	CIS 7.6; CSA TVM-01	Auth/unauth modes configured; safe windows approved; coverage reports generated	Authenticated re-scan parity on critical classes; cadence targets met	EP-07.01
5.3	Threat-intel integration framework	§6.2 VA & RBR; §6.3 TI & Adversary Simulation	RP-02 Zero Trust; RP-04 Defense in Depth; RP-15 Evidence Production	CSA TVM-02; CIS 7.1; OWASP ASVS V14.2	Feeds ingested/normalized; KEV/EPSS flags present; correlation rules active	Active-exploit CVEs auto-prioritized; campaign sims confirm prioritization	EP-07.07

Req ID	Requirement (Inputs) (§5)	Technical Specifications (Outputs) (§6)	Core Principles (§7)	Control Mappings (§9)	Verification – Build Correct (§12)	Validation – Works Right (§12)	Evidence Pack ID
5.4	Risk-based remediation (RBR) pipeline	§6.2 VA & RBR; §6.5 Patch & Config Mgmt	RP-05 Secure by Design; RP-01 Least Privilege; RP-15 Evidence Production	CIS 7.1, 7.3; CSA TVM-02	Risk model implemented; ticket/change linkage active; SLA tiers visible	Detection-to-closure meets SLA for KEV/highs; exceptions have compensating controls and retest dates	EP-07.05
5.5	CSV capability	§6.3 TI & Adversary Simulation; §6.4 CSV & Control Effectiveness	RP-04 Defense in Depth; RP-15 Evidence Production	OWASP ASVS V14.3	BAS platform configured; red/purple cadence defined; test playbooks approved	BAS/pen tests confirm exploit-block (including virtual patch); no residual lateral movement	EP-07.08
5.6	Patch & configuration management infrastructure	§6.5 Patch & Config Mgmt	RP-10 Secure Defaults; RP-20 Protect Availability; RP-15 Evidence Production	CIS 7.1; CSA TVM-01	Patch channels/baselines set; rollback procedures documented/tested	Emergency patches meet SLA; post-patch authenticated re-tests clean; rollbacks succeed when invoked	EP-07.06
5.7	Centralized logging & SIEM correlation	§6.4 CSV (reporting); §6.5 Patch & Config (post-patch logs)	RP-15 Evidence Production; RP-16 Make Compromi	OWASP ASVS V14.2 (eventing aspects)	Parsers/dashboards verified; time sync present	Correlations detect new exploitable paths within MTTD target; audit samples pass	EP-07.09

Req ID	Requirement (Inputs) (§5)	Technical Specifications (Outputs) (§6)	Core Principles (§7)	Control Mappings (§9)	Verification – Build Correct (§12)	Validation – Works Right (§12)	Evidence Pack ID
			se Detection Easier				
5.8	IR & containment workflow integration	§6.2 (virtual patching); §6.3 (adaptive tuning); §6.4 (control validation)	RP-04 Defense in Depth; RP-20 Protect Availability	CIS 7.3; CSA TVM-02	Runbooks link high-risk CVEs to IR playbooks; WAF/IPS controls are registered	Drills show containment within MTTR target; service impact minimized	EP-07.10
5.9	Least-privilege identities & secrets hygiene (TVE tooling)	§6.2; §6.4; §6.5	RP-01 Least Privilege; RP-19 Protect Integrity	CSA SEF-02; CIS 7.1	Dedicated identities/scopes verified; secrets vaulted/rotated with audit trails	Out-of-scope actions denied; rotations do not break scanning/CSV	EP-07.11
5.10	CI/CD integration & release evidence	§6.2; §6.4; §6.5	RP-05 Secure by Design; RP-15 Evidence Production; RP-11 SoD*	CSA SEF-02; CIS 7.1	Gates for policies/controls-as-code, coverage, CSV tests, and evidence capture enabled	Changes without passing gates are blocked; post-deploy CSV regression passes, and artifacts are attached to the ticket	EP-07.11
5.11	Software supply-chain inputs	§6.1; §6.2; §6.4	RP-19 Protect Integrity; RP-06 Minimize	CIS 7.1; OWASP ASVS V14.2	SBOM/provenance ingested; approved registries/artifacts enforced;	Unsigned/unknown artifacts denied at admission; signed update	EP-07.12

Req ID	Requirement (Inputs) (§5)	Technical Specifications (Outputs) (§6)	Core Principles (§7)	Control Mappings (§9)	Verification – Build Correct (§12)	Validation – Works Right (§12)	Evidence Pack ID
	(SBOM/provenance)		Attack Surface		unverified plugins quarantined	proceeds; CSV smoke passes	
5.12	Exception & compensating control process	§6.2; §6.4; §6.5	RP-05 Secure by Design; RP-15 Evidence Production	CIS 7.3	Exceptions are time-bounded with the owner and compensating controls; records linked to tickets	Exceptions expire or are re-approved on schedule; CSV proves compensating control efficacy	EP-07.05
5.13	Staging/testbeds & OT/ICS safety	§6.1; §6.5	RP-20 Protect Availability; RP-10 Secure Defaults	—	Read-only discovery for OT; vendor procedures and testbed validations documented	Safe rollout in OT window; drills show no service impact; recovery meets RTO/RPO	EP-07.13
5.14	Metrics ownership & targets	§6.1; §6.2; §6.4; §6.5	RP-15 Evidence Production; RP-16 Make Compromise Detection Easier	CIS 7.1	Owners and SLO/SLA targets set; dashboards live	Weekly reviews meet thresholds or trigger CAPs; trends improve; breaches auto-escalate	EP-07.15
5.15	Policy/Configuration/Detection as Code	§6.1; §6.2; §6.4; §6.5	RP-05 Secure by Design; RP-19	CSA SEF-02; CIS 7.1	Repos, protected branches/CODEOWNERS; lint/unit tests; signed artifacts	Promotion changes controls as intended; rollback succeeds and is evidenced	EP-07.11

Req ID	Requirement (Inputs) (§5)	Technical Specifications (Outputs) (§6)	Core Principles (§7)	Control Mappings (§9)	Verification – Build Correct (§12)	Validation – Works Right (§12)	Evidence Pack ID
			Protect Integrity				
5.16	Auth/unauth modes & safe windows	§6.2; §6.1	RP-02 Zero Trust; RP-10 Secure Defaults	CIS 7.6; CSA TVM-01	Modes set per asset class; safe windows/ratelimits documented; negative tests present	Scans run within windows; auth failures alert within target; ingest/auth errors fail closed	EP-07.01
5.17	Virtual patching & edge enforcement	§6.2; §6.4; §6.5	RP-04 Defense in Depth; RP-06 Minimize Attack Surface	CIS 7.3; CSA TVM-02	WAF/IPS/edge rules cataloged; KEV playbooks linked to tickets/SLA timers	BAS confirms exploit-block until vendor patch; closure requires rescan + CSV pass	EP-07.03
5.18	Adversary mapping services	§6.3; §6.2	RP-04 Defense in Depth; RP-16 Make Compromise Detection Easier	OWASP ASVS V14.2	TI enrichment emits ATT&CK/KEV/EPSS fields; active sector watchlists	Active campaigns trigger reprioritization ≤24h; purple-team tuning improves hit rate	EP-07.07
5.19	Tamper-evident evidence repositories	§6.4; §6.5	RP-15 Evidence Production; RP-19 Protect Integrity	—	WORM/append-only and hash verification configured; access controls/time sync verified	Independent review reproduces findings; integrity checks	EP-07.14

Req ID	Requirement (Inputs) (§5)	Technical Specifications (Outputs) (§6)	Core Principles (§7)	Control Mappings (§9)	Verification – Build Correct (§12)	Validation – Works Right (§12)	Evidence Pack ID
						pass; chain of custody intact	

Notes

- Sub-EP entries represent future IAM sub-standards to be developed; each will inherit this EP structure and include §6/§12 mappings and Quick Win artifacts.
- For every row, practitioners should record the Test-ID(s) executed and the exact EP-06.xx link in the project's register to keep traceability current.

Appendix B: EP-07 Summary Matrix – Evidence Pack Overview

Layer	EP Identifier	Purpose	Evidence Categories Included
Parent EP	EP-07.00	Stores annex-wide TVE evidence supporting §§5, 6, 10, and 12. Acts as the index/readme for all EP-07.xx sub-packs.	• TVE boundary/flow maps (discover → prioritize → remediate → validate) • Invariants register ("KEV, internet-facing ≤ 48 hours," "closure requires CSV pass," "TI ingestion fails closed") • Policies/controls-as-code repo pointers (scan profiles, risk model, baselines, virtual patch bundles, CSV suites) • Unified telemetry schema fields (asset_id, exposure_id, finding_id, risk_score, internet_facing_flag, trace_id, policy_version, time) • Quick Win index and pass/fail summaries (refs to EP-07.01/02/03)
Sub-EP	EP-07.01	Authenticated scanning coverage closeout for one in-scope estate (§§6.2, 12).	• Coverage reports (authenticated vs unauthenticated) • Vault role/config + rotation logs • Orchestrator/registry inventory snapshots (containers/serverless) • Authenticated rescan diffs • Alert samples for auth-failure events • Quick Win: "Authenticated Scanning Coverage Closeout" results
Sub-EP	EP-07.02	Auto-block new internet-facing exposures until linked to a change + CSV reachability (§§6.1, 6.4, 12).	• ASM "new exposure" alerts • Change-mismatch logs • Edge deny rule diffs • CSV reachability + negative-test outputs • Unblock approvals • Quick Win: "Auto-Block New Internet-Facing Exposures" evidence
Sub-EP	EP-07.03	KEV virtual patching on one public service; exploit-block verified (§§6.2, 6.4, 6.5, 12).	• TI enrichment logs (KEV/EPSS flags) • WAF/IPS rule bundles + diffs • Authenticated rescan results • BAS before/after runs validating exploit-block • Closure ticket with SLA timestamps

Layer	EP Identifier	Purpose	Evidence Categories Included
Sub-EP	EP-07.04	ASM ↔ CMDB parity and exposure latency validation (§§6.1, 12).	• ASM exports + CMDB sync/ parity reports • Exposure alert latency logs • Tagging completeness (criticality, internet-facing) • Reconciliation tickets and closure proofs
Sub-EP	EP-07.05	Risk-based remediation (RBR) pipeline operation with SLA enforcement (§§6.2, 6.5, 12).	• Risk model spec (KEV/EPSS/criticality weighting) • SLA policy files • Ticket/change links with detection-to-closure timestamps • Exception register with compensating controls + re-validation dates
Sub-EP	EP-07.06	Patch & secure baseline rollouts with rollback drills (§§6.5, 12).	• Baseline-as-code compliance snapshots • Patch deployment logs (including emergency channels) • Post-patch verification + authenticated rescans • Rollback drill reports (success + RTO/RPO)
Sub-EP	EP-07.07	Threat-intel ingestion/enrichment and reprioritization latency (§§6.2, 6.3, 12).	• Feed health + ingest→enrich latency metrics • CVE→ATT&CK/KEV/EPSS correlation outputs • Reprioritization events with timestamps • Purple-team tuning notes and detection diffs
Sub-EP	EP-07.08	CSV/BAS suite results for one high-risk scope (patched + virtually patched) (§§6.3, 6.4, 12).	• CSV control-pass/fail reports • BAS exploit attempts (before/after) • Regression suite outputs after material change • Residual-risk notes and CAPs
Sub-EP	EP-07.09	Centralized logging & SIEM correlation for TVE telemetry (§§6.4, 6.5, 12).	• Parser/normalization tests • Correlation rule packs and alert timelines • Audit sampling exports • Time-sync/NTP proofs
Sub-EP	EP-07.10	IR & containment drill for a high-priority vuln path (§§6.2, 6.3, 12).	• Runbooks + drill scripts • WAF/IPS deployment logs

Layer	EP Identifier	Purpose	Evidence Categories Included
			• Containment MTTR attainment • Service-impact notes and approvals
Sub-EP	EP-07.11	CI/CD gates + SoD enforcement for TVE changes (§§6.2, 6.4, 6.5, 12).	• Gate logs (coverage, KEV/EPSS, SLA timers, CSV pass) • Blocked change examples • Distinct pipeline/identity proofs (scan vs remediate vs validate) • Evidence links per commit (Spec → Test-ID → EP-07.xx)
Sub-EP	EP-07.12	Supply-chain integrity for scanning/policy bundles (§§6.1, 6.2, 6.4, 12).	• SBOM/provenance manifests • Signature/attestation verification logs • Admission/promotion policy results (approved registries/artifacts) • Quarantine/approval evidence for third-party plugins
Sub-EP	EP-07.13	OT/ICS safety controls for discovery and remediation windows (§§6.1, 6.5, 12).	• Read-only discovery configs • Vendor-approved procedures • Testbed validations • Window execution logs with “no service impact” attestations
Sub-EP	EP-07.14	Immutable evidence & integrity (WORM/append-only) with reconstruction checks (§§6.4, 12).	• Evidence store retention configs • Hash manifests + access controls • Random reconstruction samples (end-to-end fix timelines)
Sub-EP	EP-07.15	Traceability exports (ETM snapshots) Inputs (§5) → Tests (§12) → Outputs (§6).	• ETM/matrix snapshots • Change-set diffs linking Spec → Test-ID → EP-07.xx • Quarterly review sign-offs

Notes for editors

- Each EP-07.xx row should reference the exact §6 outputs and §12 Test-IDs exercised by its artifacts, and record the invariant(s) proven (“KEV, internet-facing ≤ 48 hours,” “closure requires CSV pass,” “TI ingestion fails closed,” “SoD separation”).

- Parent EP-07.00 must include a human-readable index to every sub-EP, its location, checksum manifest, and the latest pass/fail status for associated Quick Wins.
- Sub-EP entries represent present and future TVE sub-standards; each inherits this EP structure and includes §6/§12 mappings and Quick Win artifacts.

Adoption References

NOTE: ISAUnited Charter Adoption of External Organizations.

ISAUnited formally adopts the work of the International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) and the National Institute of Standards and Technology (NIST) as foundational standards bodies, and the Center for Internet Security (CIS), the Cloud Security Alliance (CSA), and the Open Worldwide Application Security Project (OWASP) as security control–framework organizations. This adoption aligns with each organization's public mission and encourages use by practitioners and institutions. ISAUnited incorporates these organizations into its charter so that every Parent Standard and Sub-Standard is grounded in a common, defensible foundation.

a) **Foundational Standards (Parent level).**
ISAUnited adopts *ISO/IEC* and *NIST* as foundational standards organizations. Parent Standards align with these bodies for architectural grounding and auditability, and extend that foundation through ISAUnited's normative, testable specifications. This alignment does not supersede *ISO/IEC* or *NIST*.
b) **Security Control Frameworks (Control level).**
ISAUnited adopts *CIS, CSA*, and *OWASP* as control framework organizations. Control mappings translate architectural intent into enforceable technical controls within Parent Standards and Sub-Standards. These frameworks provide alignment at the implementation level rather than at the foundational level.
c) **Precedence and scope.**
Foundational alignment (*ISO/IEC, NIST*) establishes the architectural baseline. Control frameworks (*CIS, CSA, OWASP*) provide enforceable mappings. ISAUnited's security invariants and normative requirements govern implementation details while remaining consistent with the adopted organizations.
d) **Mapping.**
Each cited control mapping is tied to a defined output, an associated verification and validation activity, and an Evidence Pack ID to maintain end-to-end traceability from requirement to control, test, and evidence.
e) **Attribution.**
ISAUnited cites organizations by name, respects attribution requirements, and conducts periodic alignment reviews. Updates are recorded in the Change Log with corresponding evidence.
f) **Flow-downs.**

Obsolete and withdrawn documents should not be used; please use replacements.

(Parent → Sub-Standard). Parent alignment to the International *ISO/IEC* and *NIST* flows down as architectural invariants and minimum requirements that Sub-Standards must uphold or tighten. Parent-level mappings to *CIS, CSA*, and *OWASP* flow down as implementation control intents that Sub-Standards must operationalize as controls-as-code, tests, and evidence. Each flow-down MUST reference the Parent clause, the adopted organization name, the Sub-Standard clause that implements it, the associated verification/validation test, and an Evidence Pack ID for traceability. Any variance requires a written rationale, compensating controls, and a time-bounded expiry recorded with an Evidence Pack ID.

Change Log and Revision History

Review Date	Changes	Committee	Action	Status
January 2026	Standards Revision	Standards Committee	Publication	Draft v1 published
November 2025	Standards Submitted	Technical Fellow Society	Peer review	Pending
October 2025	Standards Revision	Task Group ISAU-TG39-2024	Draft submitted	Complete
December 2024	Standards Development (Parent D01)	Task Group ISAU-TG39-2024	Draft complete	Complete

End of Document
IO.