OWASP | GenAI SECURITY PROJECT
TOP 10 FOR LLM AND GENERATIVE AI

Artificial Intelligence Security Verification Standard

Initial Version Work In Progress

_____ ___, 2026

v5.1.0 20260101-002106

Table of Contents

Frontispiece **1**
About the Standard 1
Copyright and License 1
Project Leads 2
Contributors and Reviewers 2
Preface **4**
Introduction 4
Key Objectives for AISVS Version 1.0 4
Well-Defined Scope 4
Using the AISVS **5**
Artificial Intelligence Security Verification Levels 5
Definition of the Levels 5
Role (D/V) 6
C1 Training Data Governance & Bias Management **7**
Control Objective 7
C1.1 Training Data Provenance 7
C1.2 Training Data Security & Integrity 7
C1.3 Training Data Labeling Quality, Integrity, and Security 8
C1.4 Training Data Quality and Security Assurance 8
C1.5 Data Lineage and Traceability 9
References 9
C2 User Input Validation **10**
Control Objective 10
C2.1 Prompt Injection Defense 10
C2.2 Adversarial-Example Resistance 10
C2.3 Prompt Character Set 11
C2.4 Schema, Type & Length Validation 11
C2.5 Content & Policy Screening 12
C2.6 Input Rate Limiting & Abuse Prevention 12
C2.7 Multi-Modal Input Validation 13
C2.8 Real-Time Adaptive Threat Detection 14
References 14
C3 Model Lifecycle Management & Change Control **15**

Control Objective . . . 15
C3.1 Model Authorization & Integrity . . . 15
C3.2 Model Validation & Testing . . . 15
C3.3 Controlled Deployment & Rollback . . . 16
C3.4 Secure Development Practices . . . 16
C3.5 Model Retirement & Decommissioning . . . 17
References . . . 17
C4 Infrastructure, Configuration & Deployment Security . . . 18
Control Objective . . . 18
C4.1 Runtime Environment Isolation . . . 18
C4.2 Secure Build & Deployment Pipelines . . . 18
C4.3 Network Security & Access Control . . . 19
C4.4 Secrets & Cryptographic Key Management . . . 19
C4.5 AI Workload Sandboxing & Validation . . . 20
C4.6 AI Infrastructure Resource Management, Backup and Recovery . . . 20
C4.7 AI Hardware Security . . . 21
C4.8 Edge & Distributed AI Security . . . 21
References . . . 22
C5 Access Control & Identity for AI Components & Users . . . 23
Control Objective . . . 23
C5.1 Identity Management & Authentication . . . 23
C5.2 Authorization & Policy . . . 23
C5.3 Query-Time Security Enforcement . . . 24
C5.4 Output Filtering & Data Loss Prevention . . . 24
C5.5 Multi-Tenant Isolation . . . 25
C5.6 Autonomous Agent Authorization . . . 25
References . . . 26
C6 Supply Chain Security for Models, Frameworks & Data . . . 27
Control Objective . . . 27
C6.1 Pretrained Model Vetting & Origin Integrity . . . 27
C6.2 Framework & Library Scanning . . . 27
C6.3 Dependency Pinning & Verification . . . 28
C6.4 Trusted Source Enforcement . . . 28
C6.5 Third-Party Dataset Risk Assessment . . . 29
C6.6 Supply Chain Attack Monitoring . . . 29
C6.7 ML-BOM for Model Artifacts . . . 29
References . . . 30
C7 Model Behavior, Output Control & Safety Assurance . . . 31
Control Objective . . . 31
C7.1 Output Format Enforcement . . . 31
C7.2 Hallucination Detection & Mitigation . . . 31
C7.3 Output Safety & Privacy Filtering . . . 32
C7.4 Output & Action Limiting . . . 32

C7.5 Explainability & Transparency . . . 33
C7.6 Monitoring Integration . . . 33
7.7 Generative Media Safeguards . . . 33
References . . . 34
C8 Memory, Embeddings & Vector Database Security . . . 35
Control Objective . . . 35
C8.1 Access Controls on Memory & RAG Indices . . . 35
C8.2 Embedding Sanitization & Validation . . . 36
C8.3 Memory Expiry, Revocation & Deletion . . . 36
C8.4 Prevent Embedding Inversion & Leakage . . . 36
C8.5 Scope Enforcement for User-Specific Memory . . . 37
C8.6 Advanced Memory System Security . . . 37
References . . . 38
9 Autonomous Orchestration & Agentic Action Security . . . 39
Control Objective . . . 39
9.1 Agent Task-Planning & Recursion Budgets . . . 39
9.2 Tool Plugin Sandboxing . . . 40
9.3 Autonomous Loop & Cost Bounding . . . 40
9.4 Protocol-Level Misuse Protection . . . 40
9.5 Agent Identity & Tamper-Evidence . . . 41
9.6 Multi-Agent Swarm Risk Reduction . . . 41
9.7 User & Tool Authentication / Authorization . . . 42
9.8 Agent-to-Agent Communication Security . . . 42
9.9 Intent Verification & Constraint Enforcement . . . 43
9.10 Agent Reasoning Strategy Security . . . 43
9.11 Agent Lifecycle State Management & Security . . . 44
9.12 Tool Integration Security Framework . . . 44
C9.13 Model Context Protocol (MCP) Security . . . 45
Component Integrity & Supply Chain Hygiene . . . 45
Authentication & Authorization . . . 45
Secure Transport & Network Boundary Protection . . . 46
Schema, Message, and Input Validation . . . 46
Outbound Access & Agent Execution Safety . . . 46
Transport Restrictions & High-Risk Boundary Controls . . . 47
References . . . 47
10 Adversarial Robustness & Privacy Defense . . . 49
Control Objective . . . 49
10.1 Model Alignment & Safety . . . 49
10.2 Adversarial-Example Hardening . . . 49
10.3 Membership-Inference Mitigation . . . 50
10.4 Model-Inversion Resistance . . . 50
10.5 Model-Extraction Defense . . . 50
10.6 Inference-Time Poisoned-Data Detection . . . 51

10.7 Dynamic Security Policy Adaptation . . . 51
10.8 Reflection-Based Security Analysis . . . 52
10.9 Evolution & Self-Improvement Security . . . 52
References . . . 53
11 Privacy Protection & Personal Data Management . . . 54
Control Objective . . . 54
11.1 Anonymization & Data Minimization . . . 54
11.2 Right-to-be-Forgotten & Deletion Enforcement . . . 54
11.3 Differential-Privacy Safeguards . . . 55
11.4 Purpose-Limitation & Scope-Creep Protection . . . 55
11.5 Consent Management & Lawful-Basis Tracking . . . 55
11.6 Federated Learning with Privacy Controls . . . 55
References . . . 56
C12 Monitoring, Logging & Anomaly Detection . . . 57
Control Objective . . . 57
C12.1 Request & Response Logging . . . 57
C12.2 Abuse Detection and Alerting . . . 57
C12.3 Model Drift Detection . . . 58
C12.4 Performance & Behavior Telemetry . . . 58
C12.5 AI Incident Response Planning & Execution . . . 59
C12.6 AI Performance Degradation Detection . . . 59
C12.7 DAG Visualization & Workflow Security . . . 59
C12.8 Proactive Security Behavior Monitoring . . . 60
References . . . 60
C13 Human Oversight, Accountability & Governance . . . 61
Control Objective . . . 61
C13.1 Kill-Switch & Override Mechanisms . . . 61
C13.2 Human-in-the-Loop Decision Checkpoints . . . 61
C13.3 Chain of Responsibility & Auditability . . . 62
C13.4 Explainable-AI Techniques . . . 62
C13.5 Model Cards & Usage Disclosures . . . 62
C13.6 Uncertainty Quantification . . . 63
C13.7 User-Facing Transparency Reports . . . 63
References . . . 63
Appendix A: Glossary . . . 65
Appendix B: References . . . 70
TODO . . . 70
Appendix C: AI Security Governance & Documentation (Reorganized) . . . 71
Objective . . . 71
AC.1 AI Risk Management Framework Adoption . . . 71
AC.2 AI Security Policy & Procedures . . . 71
AC.3 Roles & Responsibilities for AI Security . . . 71
AC.4 Ethical AI Guidelines Enforcement . . . 72

AC.5 AI Regulatory Compliance Monitoring . . . 72
AC.6 Training Data Governance, Documentation & Process . . . 72
AC.6.1 Data Sourcing & Due Diligence . . . 72
AC.6.2 Bias & Fairness Management . . . 73
AC.6.3 Labeling & Annotation Governance . . . 73
AC.6.4 Dataset Quality Gates & Quarantine . . . 74
AC.6.5 Threat/Poisoning Detection & Drift . . . 74
AC.6.6 Deletion, Consent, Rights, Retention & Compliance . . . 74
AC.6.7 Versioning & Change Management . . . 75
AC.6.8 Synthetic Data Governance . . . 75
AC.6.9 Access Monitoring . . . 75
AC.6.10 Adversarial Training Governance . . . 76
AC.7 Model Lifecycle Governance & Documentation . . . 76
AC.8 Prompt, Input, and Output Safety Governance . . . 76
AC.8.1 Prompt Injection Defense . . . 76
AC.8.2 Adversarial-Example Resistance . . . 77
AC.8.3 Content & Policy Screening . . . 77
AC.8.4 Input Rate Limiting & Abuse Prevention . . . 77
AC.8.5 Input Provenance & Attribution . . . 77
AC.9 Multimodal Validation, MLOps & Infrastructure Governance . . . 77
AC.9.1 Multimodal Security Validation Pipeline . . . 77
AC.9.2 CI/CD & Build Security . . . 78
AC.9.3 Container & Image Security . . . 78
AC.9.4 Monitoring, Alerting & SIEM . . . 78
AC.9.5 Vulnerability Management . . . 78
AC.9.6 Configuration & Drift Control . . . 78
AC.9.7 Production Environment Hardening . . . 78
AC.9.8 Release Promotion Gates . . . 78
AC.9.9 Workload, Capacity & Cost Monitoring . . . 79
AC.9.10 Approvals & Audit Trails . . . 79
AC.9.11 IaC Governance . . . 79
AC.9.12 Data Handling in Non-Production . . . 79
AC.9.13 Backup & Disaster Recovery . . . 79
AC.9.14 Compliance & Documentation . . . 80
AC.9.15 Hardware & Supply Chain . . . 80
AC.9.16 Cloud Strategy & Portability . . . 80
AC.9.17 GitOps & Self-Healing . . . 80
AC.9.18 Zero-Trust, Agents, Provisioning & Residency Attestation . . . 80
AC.9.19 Access Control & Identity . . . 81
New Items to be Integrated Above . . . 82
Appendix D: AI-Assisted Secure Coding Governance & Verification . . . 83
Objective . . . 83
AD.1 AI-Assisted Secure-Coding Workflow . . . 83

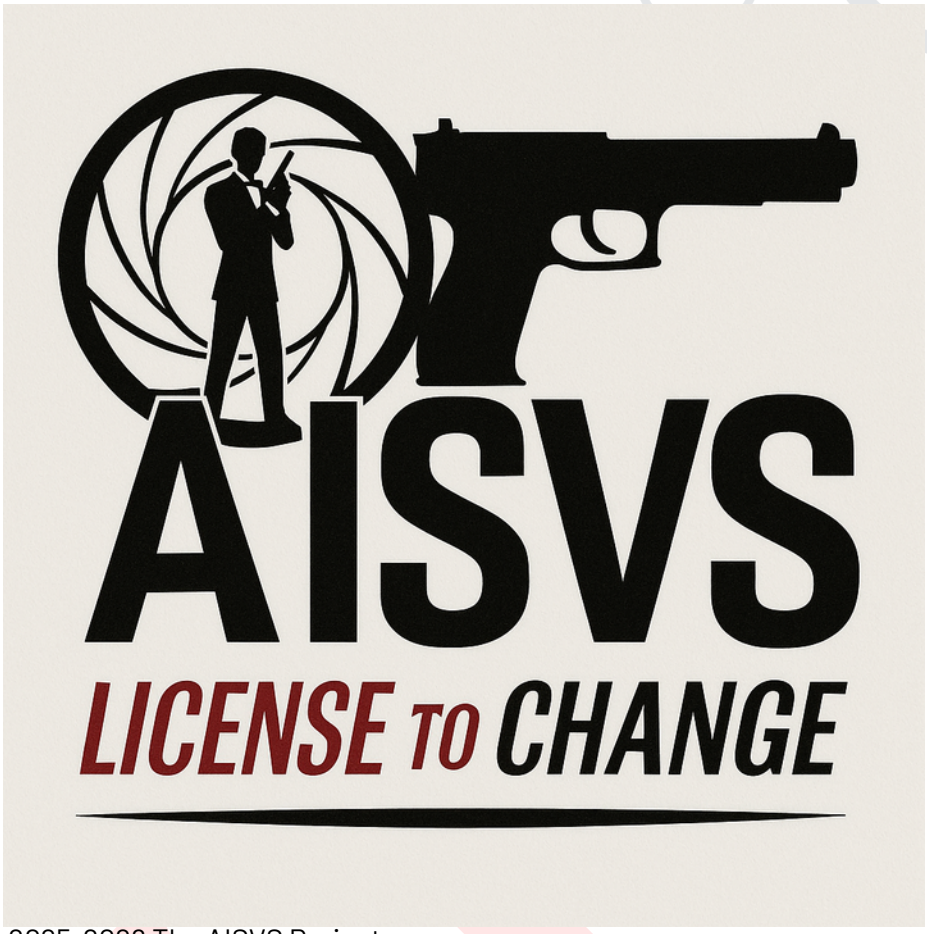
AD.2 AI Tool Qualification & Threat Modeling . . . 83
AD.3 Secure Prompt & Context Management . . . 84
AD.4 Validation of AI-Generated Code . . . 84
AD.5 Explainability & Traceability of Code Suggestions . . . 84
AD.6 Continuous Feedback & Model Fine-Tuning . . . 85
References . . . 85
Appendix E: Example Tools and Frameworks . . . 86
Objective . . . 86
AE.1 Training Data Governance & Bias Management . . . 86
AE.2 User Input Validation . . . 86
Appendix B: Strategic Controls . . . 87
C4.15 Quantum-Resistant Infrastructure Security . . . 87
C4.17 Zero-Knowledge Infrastructure . . . 87
C4.18 Side-Channel Attack Prevention . . . 88
C4.19 Neuromorphic & Specialized AI Hardware Security . . . 88
C4.20 Privacy-Preserving Compute Infrastructure . . . 89

Frontispiece

About the Standard

The Artificial Intelligence Security Verification Standard (AISVS) is a community-driven catalogue of security requirements that data scientists, MLOps engineers, software architects, developers, testers, security professionals, tool vendors, regulators, and consumers can use to design, build, test, and verify trustworthy AI-enabled systems and applications. It provides a common language for specifying security controls across the AI lifecycle—from data collection and model development to deployment and ongoing monitoring—so that organizations can measure and improve the resilience, privacy, and safety of their AI solutions.

Copyright and License



Project Leads

Jim Manico
Aras “Russ” Memisyazici

Contributors and Reviewers

https://github.com/ottosulin
https://github.com/mbhatt1
https://github.com/vineethsai
https://github.com/cciprofm
https://github.com/deepakrpandey12

AISVS is a brand-new standard created specifically to address the unique security challenges of artificial-intelligence systems. While it draws inspiration from broader security best practices, every requirement in AISVS has been developed from the ground up to reflect the AI threat landscape and to help organizations build safer, more resilient AI solutions.

Preface

Welcome to the Artificial Intelligence Security Verification Standard (AISVS) version 1.0!

Introduction

Established in 2025 through a collaborative community effort, AISVS defines the security requirements to consider when designing, developing, deploying, and operating modern AI models, pipelines, and AI-enabled services.

AISVS v1.0 represents the combined work of its project leads, working group, and wider community contributors to produce a pragmatic, testable baseline for securing AI systems.

Our goal with this release is to make AISVS straightforward to adopt while staying laser-focused on its defined scope and addressing the rapidly evolving risk landscape unique to AI.

Key Objectives for AISVS Version 1.0

Version 1.0 will be created with several guiding principles.

Well-Defined Scope

Each requirement must align with AISVS's name and mission:

- Artificial Intelligence – Controls operate at the AI/ML layer (data, model, pipeline, or inference) and are the responsibility of AI practitioners.
- Security – Requirements directly mitigate identified security, privacy, or safety risks.
- Verification – Language is written so conformance can be objectively validated.
- Standard – Sections follow a consistent structure and terminology to form a coherent reference.

By following AISVS, organizations can systematically evaluate and strengthen the security posture of their AI solutions, fostering a culture of secure AI engineering.

Using the AISVS

The Artificial Intelligence Security Verification Standard (AISVS) defines security requirements for modern AI applications and services, focusing on aspects within the control of application developers.

The AISVS is intended for anyone developing or evaluating the security of AI applications, including developers, architects, security engineers, and auditors. This chapter introduces the structure and use of the AISVS, including its verification levels and intended use cases.

Artificial Intelligence Security Verification Levels

The AISVS defines three ascending levels of security verification. Each level adds depth and complexity, enabling organizations to tailor their security posture to the risk level of their AI systems.

Organizations may begin at Level 1 and progressively adopt higher levels as security maturity and threat exposure increase.

Definition of the Levels

Each requirement in AISVS v1.0 is assigned to one of the following levels:

Level 1 requirements

Level 1 includes the most critical and foundational security requirements. These focus on preventing common attacks that do not rely on other preconditions or vulnerabilities. Most Level 1 controls are either straightforward to implement or essential enough to justify the effort.

Level 2 requirements

Level 2 addresses more advanced or less common attacks, as well as layered defenses against widespread threats. These requirements may involve more complex logic or target specific attack prerequisites.

Level 3 requirements

Level 3 includes controls that are typically harder to implement or situational in applicability. These often represent defense-in-depth mechanisms or mitigations against niche, targeted, or high-complexity attacks.

Role (D/V)

Each AISVS requirement is marked according to the primary audience:

- D – Developer-focused requirements
- V – Verifier/auditor-focused requirements
- D/V – Relevant to both developers and verifiers

C1 Training Data Governance & Bias Management

Control Objective

Training data must be sourced, handled, and maintained in a way that preserves provenance, security, quality, and fairness. Doing so fulfils legal duties and reduces risks of bias, poisoning, or privacy breaches that show up during training that could effect the entire AI lifecycle.

C1.1 Training Data Provenance

Maintain a verifiable inventory of all datasets, accept only trusted sources, and log every change for auditability.

#1.1.1 Level: 1 Role: D/V
Verify that an up-to-date inventory of every training-data source (origin, steward/owner, licence, collection method, intended use constraints, and processing history) is maintained.

#1.1.2 Level: 1 Role: D/V
Verify that training data processes exclude unnecessary features, attributes, or fields (e.g., unused metadata, sensitive PII, leaked test data).

#1.1.3 Level: 2 Role: D/V
Verify that all dataset changes are subject to a logged approval workflow.

#1.1.4 Level: 3 Role: D/V
Verify that datasets or subsets are watermarked or fingerprinted where feasible.

C1.2 Training Data Security & Integrity

Restrict access to training data, encrypt it at rest and in transit, and validate its integrity to prevent tampering, theft, or data poisoning.

#1.2.1 Level: 1 Role: D/V
Verify that access controls protect training data storage and pipelines.

#1.2.2 Level: 2 Role: D/V

Verify that all access to training data is logged, including user, time, and action.

#1.2.3 Level: 2 Role: D/V

Verify that training datasets are encrypted in transit and at rest, using industry-standard cryptographic algorithms and key management practices.

#1.2.4 Level: 2 Role: D/V

Verify that cryptographic hashes or digital signatures are used to ensure data integrity during training data storage and transfer.

#1.2.5 Level: 2 Role: D/V

Verify that that automated detection techniques are applied to guard against unauthorized modifications or corruption of training data.

#1.2.6 Level: 2 Role: D/V

Verify that obsolete training data is securely purged or anonymized.

#1.2.7 Level: 3 Role: D/V

Verify that all training dataset versions are uniquely identified, stored immutably, and auditable to support rollback and forensic analysis.

C1.3 Training Data Labeling Quality, Integrity, and Security

Protect labels and require technical review for critical data.

#1.3.1 Level: 2 Role: D/V

Verify that cryptographic hashes or digital signatures are applied to label artifacts to ensure their integrity and authenticity.

#1.3.2 Level: 2 Role: D/V

Verify that labeling interfaces and platforms enforce strong access controls, maintain tamper-evident audit logs of all labeling activities, and protect against unauthorized modifications.

#1.3.3 Level: 3 Role: D/V

Verify that sensitive information in labels is redacted, anonymized, or encrypted at the data field level at rest and in transit.

C1.4 Training Data Quality and Security Assurance

Combine automated validation, manual spot-checks, and logged remediation to guarantee dataset reliability.

#1.4.1 Level: 1 Role: D

Verify that automated tests catch format errors and nulls on every ingest or significant data transformation.

#1.4.2 Level: 2 Role: D/V

Verify that LLM training and fine-tuning pipelines implement poisoning detection & data integrity valida-

tion (e.g., statistical methods, outlier detection, embedding analysis) to identify potential poisoning attacks (e.g., label flipping, backdoor trigger insertion, role-switching commands, influential instance attacks) or unintentional data corruption in training data.

#1.4.3 Level: 2 Role: D/V

Verify that automatically generated labels (e.g., via LLMs or weak supervision) are subject to confidence thresholds and consistency checks to detect hallucinated, misleading, or low-confidence labels.

#1.4.4 Level: 3 Role: D/V

Verify that appropriate defenses, such as adversarial training (using generated adversarial examples), data augmentation with perturbed inputs, or robust optimization techniques, are implemented and tuned for relevant models based on risk assessment.

#1.4.5 Level: 3 Role: D

Verify that automated tests catch label skews on every ingest or significant data transformation.

C1.5 Data Lineage and Traceability

Track the full journey of each data point from source to model input for auditability and incident response.

#1.5.1 Level: 2 Role: D/V

Verify that the lineage of each data point, including all transformations, augmentations, and merges, is recorded and can be reconstructed.

#1.5.2 Level: 2 Role: D/V

Verify that lineage records are immutable, securely stored, and accessible for audits.

#1.5.3 Level: 2 Role: D/V

Verify that lineage tracking covers synthetic data generated via privacy-preserving or generative techniques and that all synthetic data is clearly labeled and distinguishable from real data throughout the pipeline.

References

- NIST AI Risk Management Framework
- EU AI Act – Article 10: Data & Data Governance
- CISA Advisory: Securing Data for AI Systems
- OpenAI Privacy Center – Data Deletion Controls

C2 User Input Validation

Control Objective

Robust validation of user input is a first-line defense against some of the most damaging attacks on AI systems. Prompt injection attacks can override system instructions, leak sensitive data, or steer the model toward behavior that is not allowed. Unless dedicated filters and other validation is in place, research shows that jailbreaks that exploit context windows will continue to be effective.

C2.1 Prompt Injection Defense

Prompt injection is one of the top risks for AI systems. Defenses against this tactic employ a combination of pattern filters, data classifiers and instruction hierarchy enforcement.

#2.1.1 Level: 1 Role: D/V
Verify that any external or derived input that may steer behavior, including user prompts, RAG results, plugin or MCP outputs, agent to agent messages, API or webhook responses, configuration or policy files, memory reads and memory writes, is treated as untrusted, made inert by quoting or tagging and active content removal, and screened by a maintained prompt injection detection ruleset or service before concatenation into prompts or execution of actions.

#2.1.2 Level: 1 Role: D/V
Verify that the system enforces an instruction hierarchy in which system and developer messages override user instructions and other untrusted inputs, even after processing user instructions.

#2.1.3 Level: 2 Role: D
Verify that prompts originating from third-party content (web pages, PDFs, emails) are sanitized in isolation (for example, stripping instruction-like directives and neutralizing HTML, Markdown, and script content) before being concatenated into the main prompt.

C2.2 Adversarial-Example Resistance

Natural Language Processing (NLP) models are still vulnerable to subtle character or word-level

perturbations that humans often miss but models tend to misclassify.

#2.2.1 Level: 1 Role: D
Verify that basic input normalization steps (Unicode NFC, homoglyph mapping, whitespace trimming, removal of control and invisible Unicode characters) are run before tokenization or embedding and before parsing into tool or MCP arguments.

#2.2.2 Level: 2 Role: D/V
Verify that statistical anomaly detection flags inputs with unusually high edit distance to language norms or abnormal embedding distances and that flagged inputs are gated before concatenation into prompts or execution of actions.

#2.2.3 Level: 2 Role: D
Verify that the inference pipeline supports adversarial-training-hardened model variants or defense layers (e.g., randomization, defensive distillation, alignment checks) for high-risk endpoints.

#2.2.4 Level: 2 Role: V
Verify that suspected adversarial inputs are quarantined, and logged with full payloads and trace metadata (source, tool or MCP server, agent ID, session).

#2.2.5 Level: 2 Role: D/V
Verify that encoding and representation smuggling in both inputs and outputs (e.g., invisible Unicode/control characters, homoglyph swaps, or mixed-direction text) are detected and mitigated. Approved mitigations include canonicalization, strict schema validation, policy-based rejection, or explicit marking.

C2.3 Prompt Character Set

Restricting the character set of user inputs to only allow characters that are necessary for business requirements can help prevent various types of attacks.

#2.3.1 Level: 1 Role: D
Verify that the system implements a character set limitation for user inputs, allowing only characters that are explicitly required for business purposes.

#2.3.2 Level: 1 Role: D
Verify that an allow-list approach is used to define the permitted character set.

#2.3.3 Level: 1 Role: D/V
Verify that inputs containing characters outside of the allowed set are rejected and logged with trace metadata (source, tool or MCP server, agent ID, session).

C2.4 Schema, Type & Length Validation

AI attacks featuring malformed or oversized inputs can cause parsing errors, prompt spillage across fields, and resource exhaustion. Strict schema enforcement is also a prerequisite when performing deterministic tool calls.

#2.4.1 Level: 1 Role: D
Verify that every API, tool or MCP endpoint defines an explicit input schema (JSON Schema, Protobuf or multimodal equivalent) rejects extra or unknown fields and implicit type coercion, and validates inputs server-side before prompt assembly or tool execution.

#2.4.2 Level: 1 Role: D/V
Verify that inputs exceeding maximum token or byte limits are rejected with a safe error and never silently truncated.

#2.4.3 Level: 2 Role: D/V
Verify that type checks (e.g., numeric ranges, enum values, MIME types for images/audio) are enforced server-side including for tool or MCP arguments.

#2.4.4 Level: 2 Role: D
Verify that semantic validators, that understand NLP input, run in constant time and avoid external network calls to prevent algorithmic DoS.

#2.4.5 Level: 3 Role: V
Verify that validation failures are logged with redacted payload snippets and unambiguous error codes and include trace metadata (source, tool or MCP server, agent ID, session) to aid security triage.

C2.5 Content & Policy Screening

Developers should be able to detect syntactically valid prompts that request disallowed content (such as illicit instructions, hate speech, and/or copyrighted text) then prevent them from propagating.

#2.5.1 Level: 1 Role: D
Verify that a content classifier (zero shot or fine tuned) scores every input and output for violence, self-harm, hate, sexual content and illegal requests, with configurable thresholds.

#2.5.2 Level: 1 Role: D/V
Verify that inputs which violate policies will be rejected so they will not propagate to downstream LLM or tool/MCP calls.

#2.5.3 Level: 2 Role: D
Verify that screening respects user-specific policies (age, regional legal constraints) via attribute-based rules resolved at request time, including agent-role attributes.

#2.5.4 Level: 3 Role: V
Verify that screening logs include classifier confidence scores and policy category tags with applied stage (pre-prompt or post-response) and trace metadata (source, tool or MCP server, agent ID, session) for SOC correlation and future red-team replay.

C2.6 Input Rate Limiting & Abuse Prevention

Developers should prevent abuse, resource exhaustion, and automated attacks against AI sys-

tems by limiting input rates and detecting anomalous usage patterns.

#2.6.1 Level: 1 Role: D/V
Verify that per-user, per-IP, per-API-key, and per-agent and per-session/task rate limits are enforced for all input and tool/MCP endpoints.

#2.6.2 Level: 2 Role: D/V
Verify that burst and sustained rate limits are tuned to prevent DoS and brute force attacks, and that per-task budgets (for example tokens, tool/MCP calls, and cost) are enforced for agent planning loops.

#2.6.3 Level: 2 Role: D/V
Verify that anomalous usage patterns (e.g., rapid-fire requests, input flooding, repetitive failing tool/MCP calls or recursive agent loops) trigger automated blocks or escalations.

#2.6.4 Level: 3 Role: V
Verify that abuse prevention logs are retained and reviewed for emerging attack patterns, with trace metadata (source, tool or MCP server, agent ID, session).

C2.7 Multi-Modal Input Validation

AI systems should include robust validation for non-textual inputs (images, audio, files) to prevent injection, evasion, or resource abuse.

#2.7.1 Level: 1 Role: D
Verify that all non-text inputs (images, audio, files) are validated for type, size, and format before processing, and that any extracted text (image-to-text or speech-to-text) and any hidden or embedded instructions (metadata, layers, alt text, comments) are treated as untrusted per 2.1.1.

#2.7.2 Level: 2 Role: D/V
Verify that files are scanned for malware and steganographic payloads before ingestion, and that any active content (like scripts or macros) is removed or the file is quarantined.

#2.7.3 Level: 2 Role: D/V
Verify that image/audio inputs are checked for adversarial perturbations or known attack patterns, and detections trigger gating (block or degrade capabilities) before model use.

#2.7.4 Level: 3 Role: V
Verify that multi-modal input validation failures are logged and trigger alerts for investigation, with trace metadata (source, tool or MCP server, agent ID, session).

#2.7.5 Level: 2 Role: D/V
Verify that cross-modal attack detection identifies coordinated attacks spanning multiple input types (e.g., steganographic payloads in images combined with prompt injection in text) with correlation rules and alert generation, and that confirmed detections are blocked or require HITL (human-in-the-loop) approval.

#2.7.6 Level: 3 Role: D/V
Verify that multi-modal validation failures trigger detailed logging including all input modalities, validation results and threat scores, and trace metadata (source, tool or MCP server, agent ID, session).

C2.8 Real-Time Adaptive Threat Detection

Developers should employ advanced threat detection systems for AI that adapt to new attack patterns and provide real-time protection with compiled pattern matching.

#2.8.1 Level: 1 Role: D/V
Verify that pattern matching (e.g., compiled regex) runs on all inputs and outputs (including tool/MCP surfaces) with minimal latency impact.

#2.8.3 Level: 2 Role: D/V
Verify that adaptive detection models adjust sensitivity based on recent attack activity and are updated with new patterns in real time, and trigger risk-adaptive responses (for example disable tools, shrink context, or require HITL approval).

#2.8.4 Level: 3 Role: D/V
Verify that detection accuracy is improved via contextual analysis of user history, source, and session behavior, including trace metadata (source, tool or MCP server, agent ID, session).

#2.8.5 Level: 3 Role: D/V
Verify that detection performance metrics (detection rate, false positive rate, processing latency) are continuously monitored and optimized, including time-to-block and stage (pre-prompt/post-response).

References

- OWASP LLM01:2025 Prompt Injection
- LLM Prompt Injection Prevention Cheat Sheet
- MITRE ATLAS : Adversarial Input Detection
- Mitigate jailbreaks and prompt injections

C3 Model Lifecycle Management & Change Control

Control Objective

AI systems must implement change control processes that prevent unauthorized or unsafe model modifications from reaching production. This control ensures model integrity through the entire lifecycle--from development through deployment to decommissioning--which enables rapid incident response and maintains accountability for all changes.

Core Security Goal: Only authorized, validated models reach production by employing controlled processes that maintain integrity, traceability, and recoverability.

C3.1 Model Authorization & Integrity

Only authorized models with verified integrity reach production environments.

#3.1.1 Level: 1 Role: D/V
Verify that all model artifacts (weights, configurations, tokenizers, base models, fine-tunes, adapters such as LoRA, and safety/policy models) are cryptographically signed by authorized entities and verified at deployment admission (and on load), blocking any unsigned or tampered artifact.

#3.1.2 Level: 2 Role: V
Verify that dependency tracking maintains a real-time inventory via a model registry and lineage/dependency graph, and produces a machine-readable Model/AI Bill of Materials (MBOM/AIBOM) (e.g., SPDX or CycloneDX) that enables identification of all consuming services/agents per environment (e.g., dev, staging, prod, region).

#3.1.3 Level: 3 Role: D/V
Verify that model origin integrity and trace records include an authorizing entity's identity, training data checksums, validation test results with pass/fail status, signature fingerprint/certificate chain ID, a creation timestamp, and approved deployment environments.

C3.2 Model Validation & Testing

Models must pass defined security and safety validations before deployment.

#3.2.1 Level: 1 Role: D/V
Verify that models undergo automated security testing that includes input validation, output sanitization, and safety evaluations with pre-agreed organizational pass/fail thresholds before deployment, covering agent workflows (planning, tool or MCP calls, RAG/memory, multimodal) and guardrails (policy/safety models or detection services) with a versioned evaluation harness.

#3.2.2 Level: 1 Role: V
Verify that all model changes (deployment, configuration, retirement) generate immutable audit records including a timestamp, an authenticated actor identity, a change type, and before/after states, with trace metadata (environment and consuming services/agents) and a model identifier (version/digest/signature).

#3.2.3 Level: 2 Role: D/V
Verify that validation failures automatically block model deployment unless an explicit override approval from pre-designated authorized personnel with documented business justifications.

C3.3 Controlled Deployment & Rollback

Model deployments must be controlled, monitored, and reversible.

#3.3.1 Level: 1 Role: D/V
Verify that deployment processes validate cryptographic signatures and compute integrity checksums before model activation or load, failing deployment on any mismatch.

#3.3.2 Level: 1 Role: D
Verify that production deployments implement gradual rollout mechanisms (canary deployments, blue-green deployments) with automated rollback triggers based on pre-agreed error rates, latency thresholds, guardrail/jailbreak alerts, or tool/MCP failure rates.

#3.3.3 Level: 2 Role: D/V
Verify that rollback capabilities restore the complete model state (weights, configurations, dependencies including adapters and safety/policy models) atomically.

#3.3.4 Level: 3 Role: D/V
Verify that emergency model shutdown capabilities can disable model endpoints and deactivate agent tools or MCP access, RAG/connectors and database/API credentials, and memory-store bindings within a pre-defined response time.

C3.4 Secure Development Practices

Model development and training processes must follow secure practices to prevent compromise.

#3.4.1 Level: 1 Role: D/V
Verify that model development, testing, and production environments are physically or logically separated.

They have no shared infrastructure, distinct access controls, and isolated data stores, and agent orchestration and tool or MCP servers are also isolated.

#3.4.2 Level: 1 Role: D

Verify that model development artifacts (hyperparameters, training scripts, configuration files, prompt templates, agent policies/routing graphs, tool or MCP contracts/schemas, and action catalogs or capability allow-lists) are stored in version control and require peer review approval before use in training.

#3.4.3 Level: 2 Role: D/V

Verify that model training and fine-tuning occur in isolated environments with controlled network access using egress allow-lists and no access to production tools or MCP resources.

#3.4.4 Level: 2 Role: D

Verify that training data sources are validated through integrity checks and authenticated via trusted sources with documented chain of custody before use in model development, including RAG indexes, tool logs, and agent-generated data used for fine-tuning.

C3.5 Model Retirement & Decommissioning

Models must be securely retired when they are no longer needed or when security issues are identified.

#3.5.1 Level: 1 Role: D/V

Verify that retired model artifacts (including adapters and safety/policy models) are securely wiped using secure cryptographic erasure.

#3.5.2 Level: 2 Role: V

Verify that model retirement events are logged with timestamp and actor identity, model identifier (version/digest/signature), and trace metadata (environment and consuming services/agents); model signatures are revoked, and registry/serving deny-lists plus loader cache invalidation prevent agents from loading retired artifacts.

References

- MITRE ATLAS
- MLOps Principles
- Reinforcement fine-tuning
- What is AI adversarial robustness? – IBM Research

C4 Infrastructure, Configuration & Deployment Security

Control Objective

AI infrastructure must be hardened against privilege escalation, supply chain tampering, and lateral movement through secure configuration, runtime isolation, trusted deployment pipelines, and comprehensive monitoring. Only validated and authorized infrastructure components reach production through controlled processes that ensure security, integrity, and auditability.

C4.1 Runtime Environment Isolation

Prevent container escapes and privilege escalation through OS-level isolation primitives.

#4.1.1 Level: 1 Role: D/V
Verify that all AI workloads run with minimal permissions needed on the operating system, by e.g. dropping unnecessary Linux capabilities in case of a container.

#4.1.2 Level: 1 Role: D/V
Verify that workloads are protected by technologies limiting exploitation such as sandboxing, seccomp profiles, AppArmor, SELinux or similar, and that the configuration is appropriate.

#4.1.3 Level: 2 Role: D/V
Verify that workloads run with a read-only root filesystem, and that any writable mounts are explicitly defined and hardened with restrictive options (e.g., noexec, nosuid, nodev).

#4.1.4 Level: 2 Role: D/V
Verify that runtime monitoring detects privilege-escalation and container-escape behaviors and automatically terminates offending processes.

#4.1.5 Level: 3 Role: D/V
Verify that high-risk AI workloads run in hardware-isolated environments (e.g., TEEs, trusted hypervisors, or bare-metal nodes) only after successful remote attestation.

C4.2 Secure Build & Deployment Pipelines

Ensure cryptographic integrity and supply chain security through reproducible builds and signed

artifacts.

#4.2.1 Level: 1 Role: D/V
Verify that builds are reproducible and produce signed provenance metadata as appropriate for the build artifacts that can be independently verified.

#4.2.2 Level: 2 Role: D/V
Verify that builds produce a software bill of materials (SBOM) and are signed before being accepted for deployment.

#4.2.3 Level: 2 Role: D/V
Verify that build artifact (e.g., container images) signatures and provenance metadata are validated at deployment, and unverified artifacts are rejected.

C4.3 Network Security & Access Control

Implement zero-trust networking with default-deny policies and encrypted communications.

#4.3.1 Level: 1 Role: D/V
Verify that network policies enforce default-deny ingress and egress, with only required services explicitly allowed.

#4.3.2 Level: 1 Role: D/V
Verify that administrative access protocols (e.g., SSH, RDP) and access to cloud metadata services are restricted and require strong authentication.

#4.3.3 Level: 2 Role: D/V
Verify that egress traffic is restricted to approved destinations and all requests are logged.

#4.3.4 Level: 2 Role: D/V
Verify that inter-service communication uses mutual TLS with certificate validation and regular automated rotation.

#4.3.5 Level: 2 Role: D/V
Verify that AI workloads and environments (dev, test, prod) run in isolated network segments (VPCs/VNets) with no direct internet access and no shared IAM roles, security groups, or cross-environment connectivity.

C4.4 Secrets & Cryptographic Key Management

Protect secrets and cryptographic keys with secure storage, automated rotation, and strong access controls.

#4.4.1 Level: 1 Role: D/V
Verify that secrets are stored in a dedicated secrets management system with encryption at rest and isolated from application workloads.

#4.4.2 Level: 1 Role: D/V
Verify that cryptographic keys are generated and stored in hardware-backed modules (e.g., HSMs, cloud KMS).

#4.4.3 Level: 1 Role: D/V
Verify that access to production secrets requires strong authentication.
#4.4.4 Level: 1 Role: D/V
Verify that secrets are deployed to applications at runtime through a dedicated secrets management system. Secrets must never be embedded in source code, configuration files, build artifacts, container images, or environment variables.
#4.4.5 Level: 2 Role: D/V
Verify that secrets rotation is automated.

C4.5 AI Workload Sandboxing & Validation

Isolate untrusted AI models in secure sandboxes and protect sensitive AI workloads using trusted execution environments (TEEs) and confidential computing technologies.

#4.5.1 Level: 1 Role: D/V
Verify that external or untrusted AI models execute in isolated sandboxes.
#4.5.2 Level: 1 Role: D/V
Verify that sandboxed workloads have no outbound network connectivity by default, with any required access explicitly defined.
#4.5.3 Level: 2 Role: D/V
Verify that workload attestation is performed before model or workload loading, ensuring cryptographic proof of a trusted execution environment.
#4.5.4 Level: 3 Role: D/V
Verify that confidential workloads execute within a trusted execution environment (TEE) that provides hardware-enforced isolation, memory encryption, and integrity protection.
#4.5.5 Level: 3 Role: D/V
Verify that confidential inference services prevent model extraction through encrypted computation with sealed model weights and protected execution.
#4.5.6 Level: 3 Role: D/V
Verify that orchestration of trusted execution environments includes lifecycle management, remote attestation, and encrypted communication channels.
#4.5.7 Level: 3 Role: D/V
Verify that secure multi-party computation (SMPC) enables collaborative AI training without exposing individual datasets or model parameters.

C4.6 AI Infrastructure Resource Management, Backup and Recove

Prevent resource exhaustion attacks and ensure fair resource allocation through quotas and monitoring. Maintain infrastructure resilience through secure backups, tested recovery procedures, and disaster recovery capabilities.

#4.6.1 Level: 2 Role: D/V
Verify that workload's resource consumption is limited appropriately with e.g. Kubernetes ResourceQuotas or similar to mitigate Denial of Service attacks.

#4.6.2 Level: 2 Role: D/V
Verify that resource exhaustion triggers automated protections (e.g., rate limiting or workload isolation) once defined CPU, memory, or request thresholds are exceeded.

#4.6.3 Level: 2 Role: D/V
Verify that backup systems run in isolated networks with separate credentials, and the storage system is either run in an air-gapped network or implements WORM (write-once-read-many) protection against unauthorized modification.

C4.7 AI Hardware Security

Secure AI-specific hardware components including GPUs, TPUs, and specialized AI accelerators.

#4.7.1 Level: 2 Role: D/V
Verify that before workload execution, AI accelerator integrity is validated using hardware-based attestation mechanisms (e.g., TPM, DRTM, or equivalent).

#4.7.2 Level: 2 Role: D/V
Verify that accelerator (GPU) memory is isolated between workloads through partitioning mechanisms with memory sanitization between jobs.

#4.7.3 Level: 3 Role: D/V
Verify that hardware security modules (HSMs) protect AI model weights and cryptographic keys with certification to FIPS 140-3 Level 3 or Common Criteria EAL4+.

#4.7.4 Level: 2 Role: D/V
Verify that accelerator firmware (GPU/TPU/NPUs) is version-pinned, signed, and attested at boot; unsigned or debug firmware is blocked.

#4.7.5 Level: 2 Role: D/V
Verify that VRAM and on-package memory are zeroed between jobs/tenants and that device reset policies prevent cross-tenant data remanence.

#4.7.6 Level: 2 Role: D/V
Verify that partitioning/isolation features (e.g., MIG/VM partitioning) are enforced per tenant and prevent peer-to-peer memory access across partitions.

#4.7.7 Level: 3 Role: D/V
Verify that accelerator interconnects (NVLink/PCIe/InfiniBand/RDMA/NCCL) are restricted to approved topologies and authenticated endpoints; plaintext cross-tenant links are disallowed.

#4.7.8 Level: 3 Role: D
Verify that accelerator telemetry (power, temps, ECC, perf counters) is exported to SIEM/OTel and alerts on anomalies indicative of side-channels or covert channels.

C4.8 Edge & Distributed AI Security

Secure distributed AI deployments including edge computing, federated learning, and multi-site architectures.

#4.8.1 Level: 2 Role: D/V
Verify that edge AI devices authenticate to central infrastructure using mutual TLS.

#4.8.2 Level: 2 Role: D/V
Verify that edge devices implement secure boot with verified signatures and rollback protection to prevent firmware downgrade attacks.

#4.8.3 Level: 3 Role: D/V
Verify that distributed AI coordination uses Byzantine fault-tolerant consensus mechanisms with participant validation and malicious node detection.

#4.8.4 Level: 3 Role: D/V
Verify that edge-to-cloud communication supports bandwidth throttling, data compression, and secure offline operation with encrypted local storage.

#4.8.5 Level: 3 Role: D/V
Verify that mobile or edge inference applications implement platform-level anti-tampering protections (e.g., code signing, verified boot, runtime self-integrity checks) that detect and block modified binaries, repackaged apps, or attached instrumentation frameworks.

#4.8.6 Level: 3 Role: D/V
Verify that models deployed to edge or mobile devices are cryptographically signed during packaging, and that the on-device runtime validates these signatures or checksums before loading or inference; unverified or altered models must be rejected.

#4.8.7 Level: 3 Role: D/V
Verify that on-device inference runtimes enforce process, memory, and file access isolation to prevent model dumping, debugging, or extraction of intermediate embeddings and activations.

#4.8.8 Level: 3 Role: D/V
Verify that model weights and sensitive parameters stored locally are encrypted using hardware-backed key stores or secure enclaves (e.g., Android Keystore, iOS Secure Enclave, TPM/TEE), with keys inaccessible to user space.

#4.8.9 Level: 3 Role: D/V
Verify that models packaged within mobile, IoT, or embedded applications are encrypted or obfuscated at rest, and decrypted only inside a trusted runtime or secure enclave, preventing direct extraction from the app package or filesystem.

References

- NIST Cybersecurity Framework 2.0
- CIS Controls v8
- Kubernetes Security Best Practices
- Cloud Security Alliance: Cloud Controls Matrix
- ENISA: Secure Infrastructure Design
- NIST AI Risk Management Framework

C5 Access Control & Identity for AI Components & Users

Control Objective

Effective access control for AI systems requires robust identity management, context-aware authorization, and runtime enforcement following zero-trust principles. These controls ensure that humans, services, and autonomous agents only interact with models, data, and computational resources within explicitly granted scopes, with continuous verification and audit capabilities.

C5.1 Identity Management & Authentication

Establish cryptographically-backed identities for all entities with multi-factor authentication.

#5.1.1 Level: 1 Role: D/V
Verify that all human users and service principals authenticate through a centralized enterprise identity provider (IdP) using OIDC and/or SAML protocols.

#5.1.2 Level: 1 Role: D/V
Verify that high-risk operations (model deployment, weight export, training data access, production configuration changes) require multi-factor authentication or step-up authentication with session re-validation.

#5.1.3 Level: 3 Role: D/V
Verify that federated AI agents authenticate via signed JWT assertions that have a maximum lifetime of 24 hours and include cryptographic proof of origin.

C5.2 Authorization & Policy

Implement access controls for all AI resources with explicit permission models and audit trails.

#5.2.1 Level: 1 Role: D/V
Verify that every AI resource (datasets, models, endpoints, vector collections, embedding indices, compute instances) enforces role-based access controls with explicit allow-lists and default-deny policies.

#5.2.2 Level: 1 Role: V
Verify that all access control modifications are logged immutably with timestamps, actor identities, re-

source identifiers, and permission changes.

#5.2.3 Level: 2 Role: D
Verify that data classification labels (PII, PHI, proprietary, etc) automatically propagate to derived resources (embeddings, prompt caches, model outputs).

#5.2.4 Level: 2 Role: D/V
Verify that unauthorized access attempts and privilege escalation events trigger real-time alerts with contextual metadata.

#5.2.5 Level: 1 Role: D/V
Verify that authorization decisions are externalized to a dedicated policy engine (OPA, Cedar, or equivalent).

#5.2.6 Level: 1 Role: D/V
Verify that policies evaluate dynamic attributes at runtime including user role or group, resource classification, request context, tenant isolation, and temporal constraints.

#5.2.7 Level: 3 Role: D/V
Verify that policy cache time-to-live (TTL) values do not exceed 5 minutes for high-sensitivity resources and 1 hour for standard resources with cache invalidation capabilities.

C5.3 Query-Time Security Enforcement

Implement database-layer security controls with mandatory filtering and row-level security policies.

#5.3.1 Level: 1 Role: D/V
Verify that all vector database and SQL queries include mandatory security filters (tenant ID, sensitivity labels, user scope) enforced at the database engine level.

#5.3.2 Level: 1 Role: D/V
Verify that row-level security policies and field-level masking are enabled with policy inheritance for all vector databases, search indices, and training datasets.

#5.3.3 Level: 2 Role: D
Verify that failed authorization evaluations will immediately abort queries and return explicit authorization error codes.

#5.3.4 Level: 3 Role: D/V
Verify that query retry mechanisms re-evaluate authorization policies to account for dynamic permission changes within active user sessions.

C5.4 Output Filtering & Data Loss Prevention

Deploy post-processing controls to prevent unauthorized data exposure in AI-generated content.

#5.4.1 Level: 1 Role: D/V

Verify that post-inference filtering mechanisms scan and redact unauthorized PII, classified information, and proprietary data before delivering content to requestors.

#5.4.2 Level: 1 Role: D/V

Verify that citations, references, and source attributions in model outputs are validated against caller entitlements and removed if unauthorized access is detected.

#5.4.3 Level: 2 Role: D

Verify that output format restrictions (sanitized PDFs, metadata-stripped images, approved file types) are enforced based on user permission levels and data classifications.

C5.5 Multi-Tenant Isolation

Ensure cryptographic and logical isolation between tenants in shared AI infrastructure.

#5.5.1 Level: 1 Role: D/V

Verify that memory spaces, embedding stores, cache entries, and temporary files are namespace-segregated per tenant with secure purging on tenant deletion or session termination.

#5.5.2 Level: 1 Role: D/V

Verify that every API request includes an authenticated tenant identifier that is cryptographically validated against session context and user entitlements.

#5.5.3 Level: 2 Role: D

Verify that network policies implement default-deny rules for cross-tenant communication within service meshes and container orchestration platforms.

#5.5.4 Level: 3 Role: D

Verify that encryption keys are unique per tenant with customer-managed key (CMK) support and cryptographic isolation between tenant data stores.

C5.6 Autonomous Agent Authorization

Control permissions for AI agents and autonomous systems through scoped capability tokens and continuous authorization.

#5.6.1 Level: 1 Role: D/V

Verify that autonomous agents receive scoped capability tokens that explicitly enumerate permitted actions, accessible resources, time boundaries, and operational constraints.

#5.6.2 Level: 1 Role: D/V

Verify that high-risk capabilities (file system access, code execution, external API calls, financial transactions) are disabled by default and require explicit authorization.

#5.6.3 Level: 2 Role: D

Verify that capability tokens are bound to user sessions, include cryptographic integrity protection, and ensure that they cannot be persisted or reused in offline scenarios.

#5.6.4 Level: 2 Role: V
Verify that agent-initiated actions undergo authorization through an ABAC policy engine.

References

- NIST SP 800-162: Guide to Attribute Based Access Control (ABAC)
- NIST SP 800-207: Zero Trust Architecture
- NIST SP 800-63-3: Digital Identity Guidelines
- NIST IR 8360: Machine Learning for Access Control Policy Verification

C6 Supply Chain Security for Models, Frameworks & Data

Control Objective

AI supply-chain attacks exploit third-party models, frameworks, or datasets to embed backdoors, bias, or exploitable code. These controls provide end-to-end traceability, vulnerability management, and monitoring to protect the entire model lifecycle.

C6.1 Pretrained Model Vetting & Origin Integrity

Assess and authenticate third-party model origins, licenses, and hidden behaviors before any fine-tuning or deployment.

#6.1.1 Level: 1 Role: D/V
Verify that every third-party model artifact includes a signed origin record identifying source repository and commit hash.

#6.1.2 Level: 1 Role: D/V
Verify that models are scanned for malicious layers or Trojan triggers using automated tools before import.

#6.1.3 Level: 2 Role: D
Verify that transfer-learning fine-tunes pass adversarial evaluation to detect hidden behaviors.

#6.1.4 Level: 2 Role: V
Verify that model licenses, export-control tags, and data-origin statements are recorded in a ML-BOM entry.

#6.1.5 Level: 3 Role: D/V
Verify that high-risk models (publicly uploaded weights, unverified creators) remain quarantined until human review and sign-off.

C6.2 Framework & Library Scanning

Continuously scan ML frameworks and libraries for CVEs and malicious code to keep the runtime stack secure.

#6.2.1 Level: 1 Role: D/V

Verify that CI pipelines run dependency scanners on AI frameworks and critical libraries.

#6.2.2 Level: 1 Role: D/V

Verify that critical vulnerabilities (CVSS ≥ 7.0) block promotion to production images.

#6.2.3 Level: 2 Role: D

Verify that static code analysis runs on forked or vendored ML libraries.

#6.2.4 Level: 2 Role: V

Verify that framework upgrade proposals include a security impact assessment referencing public CVE feeds.

#6.2.5 Level: 3 Role: V

Verify that runtime sensors alert on unexpected dynamic library loads that deviate from the signed SBOM.

C6.3 Dependency Pinning & Verification

Pin every dependency to immutable digests and reproduce builds to guarantee identical, tamper-free artifacts.

#6.3.1 Level: 1 Role: D/V

Verify that all package managers enforce version pinning via lockfiles.

#6.3.2 Level: 1 Role: D/V

Verify that immutable digests are used instead of mutable tags in container references.

#6.3.3 Level: 2 Role: D

Verify that reproducible-build checks compare hashes across CI runs to ensure identical outputs.

#6.3.4 Level: 2 Role: V

Verify that build attestations are stored for 18 months for audit traceability.

#6.3.5 Level: 3 Role: D

Verify that expired dependencies trigger automated PRs to update or fork pinned versions.

C6.4 Trusted Source Enforcement

Allow artifact downloads only from cryptographically verified, organization-approved sources and block everything else.

#6.4.1 Level: 1 Role: D/V

Verify that model weights, datasets, and containers are downloaded only from approved domains or internal registries.

#6.4.2 Level: 1 Role: D/V

Verify that Sigstore/Cosign signatures validate publisher identity before artifacts are cached locally.

#6.4.3 Level: 2 Role: D

Verify that egress proxies block unauthenticated artifact downloads to enforce trusted-source policy.

#6.4.4 Level: 2 Role: V

Verify that repository allow-lists are reviewed quarterly with evidence of business justification for each entry.

#6.4.5 Level: 3 Role: V
Verify that policy violations trigger quarantining of artifacts and rollback of dependent pipeline runs.

C6.5 Third-Party Dataset Risk Assessment

Evaluate external datasets for poisoning, bias, and legal compliance, and monitor them throughout their lifecycle.

#6.5.1 Level: 1 Role: D/V
Verify that external datasets undergo poisoning risk scoring (e.g., data fingerprinting, outlier detection).

#6.5.2 Level: 1 Role: D
Verify that bias metrics (demographic parity, equal opportunity) are calculated before dataset approval.

#6.5.3 Level: 2 Role: V
Verify that origin, lineage, and license terms for datasets are captured in ML-BOM entries.

#6.5.4 Level: 2 Role: V
Verify that periodic monitoring detects drift or corruption in hosted datasets.

#6.5.5 Level: 3 Role: D
Verify that disallowed content (copyright, PII) is removed via automated scrubbing prior to training.

C6.6 Supply Chain Attack Monitoring

Detect supply-chain threats early through CVE feeds, audit-log analytics, and red-team simulations.

#6.6.1 Level: 1 Role: V
Verify that CI/CD audit logs stream to SIEM detections for anomalous package pulls or tampered build steps.

#6.6.2 Level: 2 Role: D
Verify that incident response playbooks include rollback procedures for compromised models or libraries.

#6.6.3 Level: 3 Role: V
Verify that threat-intel enrichment tags ML-specific indicators (e.g., model-poisoning IoCs) in alert triage.

C6.7 ML-BOM for Model Artifacts

Generate and sign detailed ML-specific SBOMs (ML-BOMs) so downstream consumers can verify

component integrity at deploy time.

#6.7.1 Level: 1 Role: D/V
Verify that every model artifact publishes a ML-BOM that lists datasets, weights, hyperparameters, and licenses.

#6.7.2 Level: 1 Role: D/V
Verify that ML-BOM generation and Cosign signing are automated in CI and required for merge.

#6.7.3 Level: 2 Role: D
Verify that ML-BOM completeness checks fail the build if any component metadata (hash, license) is missing.

#6.7.4 Level: 2 Role: V
Verify that downstream consumers can query ML-BOMs via API to validate imported models at deploy time.

#6.7.5 Level: 3 Role: V
Verify that ML-BOMs are version-controlled and diffed to detect unauthorized modifications.

References

- OWASP LLM03:2025 Supply Chain
- MITRE ATLAS : Supply Chain Compromise
- SBOM Overview – CISA
- CycloneDX – Machine Learning Bill of Materials

C7 Model Behavior, Output Control & Safety Assurance

Control Objective

This control category ensures that model outputs are technically constrained, validated, and monitored so that unsafe, malformed, or high-risk responses cannot reach users or downstream systems.

C7.1 Output Format Enforcement

Ensure the model outputs data in a way that helps prevents injection.

#7.1.1 Level: 1 Role: D/V
Verify that the application validates all model outputs against a strict schema (like JSON Schema) and rejects any output that does not match.

#7.1.2 Level: 1 Role: D/V
Verify that the system uses "stop sequences" or token limits to strictly cut off generation before it can overflow buffers or executes unintended commands.

#7.1.3 Level: 2 Role: D/V
Verify that components processing model output treat it as untrusted input (e.g., using parameterized queries or safe de-serializers).

#7.1.4 Level: 3 Role: V
Verify that the system logs the specific error type when an output is rejected for bad formatting.

C7.2 Hallucination Detection & Mitigation

Detect when the model is unsure or lying, and stop that information from reaching the user.

#7.2.1 Level: 1 Role: D/V
Verify that the system calculates a numerical confidence score (e.g., using log-probabilities) for generated answers.

#7.2.2 Level: 1 Role: D/V

Verify that the application automatically blocks answers or switches to a fallback message if the confidence score drops below a defined threshold.

#7.2.3 Level: 2 Role: D/V
Verify that hallucination events (low-confidence responses) are logged with input/output metadata for analysis.

C7.3 Output Safety & Privacy Filtering

Technical controls to detect and scrub bad content before it is shown to the user.

#7.3.1 Level: 1 Role: D/V
Verify that automated classifiers scan every response and block content that matches hate, harassment, or sexual violence categories.

#7.3.2 Level: 1 Role: D/V
Verify that the system scans every response for PII (like credit cards or emails) and automatically redacts it before display.

#7.3.3 Level: 2 Role: D
Verify that data labeled as "confidential" in the system remains blocked or redacted.

#7.3.4 Level: 3 Role: D/V
Verify that the system requires a human approval step or re-authentication if the model generates high-risk content.

#7.3.5 Level: 3 Role: D/V
Verify that safety filters can be configured differently based on the user's role or location (e.g., stricter filters for minors).

C7.4 Output & Action Limiting

Prevent the model from doing too much, too fast, or accessing things it should not.

#7.4.1 Level: 1 Role: D
Verify that the system enforces hard limits on requests and tokens per user to prevent cost spikes and denial of service.

#7.4.2 Level: 1 Role: D/V
Verify that the model cannot execute high-impact actions (like writing files, sending emails, or executing code) without explicit user confirmation.

#7.4.3 Level: 2 Role: D
Verify that the agent framework explicitly configures and enforces the maximum depth of recursive calls, delegation limits, and the list of allowed external tools.

C7.5 Explainability & Transparency

Ensure the user knows why a decision was made.

#7.5.1 Level: 2 Role: D/V
Verify that the UI displays a confidence score or "reasoning summary" to the user for critical decisions.
#7.5.2 Level: 2 Role: D/V
Verify that explanations provided to the user are sanitized to remove system prompts or backend data.
#7.5.3 Level: 3 Role: D
Verify that technical evidence of the model's decision (like attention maps or log-probs) are logged.

C7.6 Monitoring Integration

Ensure the application sends the right signals for security teams to watch.

#7.6.1 Level: 1 Role: D
Verify that the system logs real-time metrics for safety violations (e.g., "Hallucination Detected", "PII Blocked").
#7.6.2 Level: 1 Role: V
Verify that the system triggers an alert if safety violation rates exceed a defined threshold within a specific time window.
#7.6.3 Level: 2 Role: V
Verify that logs include the specific model version and other details necessary to investigate potential abuse.

7.7 Generative Media Safeguards

Prevent the creation of illegal or fake media.

#7.7.1 Level: 1 Role: D/V
Verify that the system refuses to generate media (images/audio) that depicts real people without verified consent.
#7.7.2 Level: 2 Role: D/V
Verify that input filters block prompts requesting explicit or deepfake content before the model processes them.
#7.7.3 Level: 2 Role: V
Verify that the system checks generated content for copyright violations before releasing it.
#7.7.4 Level: 3 Role: D/V
Verify that all generated media includes an invisible watermark or cryptographic signature to prove it was

AI-generated.

#7.7.5 Level: 3 Role: V

Verify that attempts to bypass filters are detected and logged as security events.

References

- OWASP Top 10 for LLMs: LLM07: Insecure Output Handling

C8 Memory, Embeddings & Vector Database Security

Control Objective

Embeddings and vector stores act as the "live memory" of contemporary AI systems, continuously accepting user-supplied data and surfacing it back into model contexts via Retrieval-Augmented Generation (RAG). If left ungoverned, this memory can leak PII, violate consent, or be inverted to reconstruct the original text. The objective of this control family is to harden memory pipelines and vector databases so that access is least-privilege, embeddings are privacy-preserving, stored vectors expire or can be revoked on demand, and per-user memory never contaminates another user's prompts or completions.

C8.1 Access Controls on Memory & RAG Indices

Enforce fine-grained access controls on every vector collection.

#8.1.1 Level: 1 Role: D/V
Verify that row/namespace-level access control rules restrict insert, delete, and query operations per tenant, collection, or document tag.

#8.1.2 Level: 1 Role: D/V
Verify that API keys or JWTs carry scoped claims (e.g., collection IDs, action verbs) and are rotated at least quarterly.

#8.1.3 Level: 2 Role: D/V
Verify that privilege-escalation attempts (e.g., cross-namespace similarity queries) are detected and logged to a SIEM within 5 minutes.

#8.1.4 Level: 2 Role: D/V
Verify that vector DB audits log subject-identifier, operation, vector ID/namespace, similarity threshold, and result count.

#8.1.5 Level: 3 Role: V
Verify that access decisions are tested for bypass flaws whenever engines are upgraded or index-sharding rules change.

C8.2 Embedding Sanitization & Validation

Pre-screen text for PII, redact or pseudonymise before vectorisation, and optionally post-process embeddings to strip residual signals.

#8.2.1 Level: 1 Role: D/V
Verify that PII and regulated data are detected via automated classifiers and masked, tokenised, or dropped pre-embedding.

#8.2.2 Level: 1 Role: D
Verify that embedding pipelines reject or quarantine inputs containing executable code or non-UTF-8 artifacts that could poison the index.

#8.2.3 Level: 2 Role: D/V
Verify that local or metric differential-privacy sanitization is applied to sentence embeddings whose distance to any known PII token falls below a configurable threshold.

#8.2.4 Level: 2 Role: V
Verify that sanitization efficacy (e.g., recall of PII redaction, semantic drift) is validated at least semi-annually against benchmark corpora.

#8.2.5 Level: 3 Role: D/V
Verify that sanitization configs are version-controlled and changes undergo peer review.

C8.3 Memory Expiry, Revocation & Deletion

GDPR "right to be forgotten" and similar laws require timely erasure; vector stores must therefore support TTLs, hard deletes, and tomb-stoning so that revoked vectors cannot be recovered or re-indexed.

#8.3.1 Level: 1 Role: D/V
Verify that every vector and metadata record carries a TTL or explicit retention label honoured by automated cleanup jobs.

#8.3.2 Level: 1 Role: D/V
Verify that user-initiated deletion requests purge vectors, metadata, cache copies, and derivative indices within 30 days.

#8.3.3 Level: 2 Role: D
Verify that logical deletes are followed by cryptographic shredding of storage blocks if hardware supports it, or by key-vault key destruction.

#8.3.4 Level: 3 Role: D/V
Verify that expired vectors are excluded from nearest-neighbour search results in < 500 ms after expiration.

C8.4 Prevent Embedding Inversion & Leakage

Recent defences—noise superposition, projection networks, privacy-neuron perturbation, and application-layer encryption—can cut token-level inversion rates below 5%.

#8.4.1 Level: 1 Role: V
Verify that a formal threat model covering inversion, membership and attribute-inference attacks exists and is reviewed yearly.

#8.4.2 Level: 2 Role: D/V
Verify that application-layer encryption or searchable encryption shields vectors from direct reads by infrastructure admins or cloud staff.

#8.4.3 Level: 3 Role: V
Verify that defence parameters (ε for DP, noise σ, projection rank k) balance privacy ≥ 99 % token protection and utility ≤ 3 % accuracy loss.

#8.4.4 Level: 3 Role: D/V
Verify that inversion-resilience metrics are part of release gates for model updates, with regression budgets defined.

C8.5 Scope Enforcement for User-Specific Memory

Cross-tenant leakage remains a top RAG risk: improperly filtered similarity queries can surface another customer's private docs.

#8.5.1 Level: 1 Role: D/V
Verify that every retrieval query is post-filtered by tenant/user ID before being passed to the LLM prompt.

#8.5.2 Level: 1 Role: D
Verify that collection names or namespaced IDs are salted per user or tenant so vectors cannot collide across scopes.

#8.5.3 Level: 2 Role: D/V
Verify that similarity results above a configurable distance threshold but outside the caller's scope are discarded and trigger security alerts.

#8.5.4 Level: 2 Role: V
Verify that multi-tenant stress tests simulate adversarial queries attempting to retrieve out-of-scope documents and demonstrate zero leakage.

#8.5.5 Level: 3 Role: D/V
Verify that encryption keys are segregated per tenant, ensuring cryptographic isolation even if physical storage is shared.

C8.6 Advanced Memory System Security

Security controls for sophisticated memory architectures including episodic, semantic, and working memory with specific isolation and validation requirements.

#8.6.1 Level: 1 Role: D/V
Verify that different memory types (episodic, semantic, working) have isolated security contexts with role-based access controls, separate encryption keys, and documented access patterns for each memory type.

#8.6.2 Level: 2 Role: D/V
Verify that memory consolidation processes include security validation to prevent injection of malicious memories through content sanitization, source verification, and integrity checks before storage.

#8.6.3 Level: 2 Role: D/V
Verify that memory retrieval queries are validated and sanitized to prevent extraction of unauthorized information through query pattern analysis, access control enforcement, and result filtering.

#8.6.4 Level: 3 Role: D/V
Verify that memory forgetting mechanisms securely delete sensitive information with cryptographic erasure guarantees using key deletion, multi-pass overwriting, or hardware-based secure deletion with verification certificates.

#8.6.5 Level: 3 Role: D/V
Verify that memory system integrity is continuously monitored for unauthorized modifications or corruption through checksums, audit logs, and automated alerts when memory content changes outside normal operations.

References

- Vector database security: Pinecone – IronCore Labs
- Securing the Backbone of AI: Safeguarding Vector Databases and Embeddings – Privacera
- Enhancing Data Security with RBAC of Qdrant Vector Database – AI Advances
- Mitigating Privacy Risks in LLM Embeddings from Embedding Inversion – arXiv
- DPPN: Detecting and Perturbing Privacy-Sensitive Neurons – OpenReview
- Art. 17 GDPR – Right to Erasure
- Sensitive Data in Text Embeddings Is Recoverable – Tonic.ai
- PII Identification and Removal – NVIDIA NeMo Docs
- De-identifying Sensitive Data – Google Cloud DLP
- Remove PII from Conversations Using Sensitive Information Filters – AWS Bedrock Guardrails
- Think Your RAG Is Secure? Think Again – Medium
- Design a Secure Multitenant RAG Inferencing Solution – Microsoft Learn
- Best Practices for Multi-Tenancy RAG with Milvus – Milvus Blog

9 Autonomous Orchestration & Agentic Action Security

Control Objective

Ensure that autonomous or multi-agent AI systems can only execute actions that are explicitly intended, authenticated, auditable, and within bounded cost and risk thresholds. This protects against threats such as Autonomous-System Compromise, Tool Misuse, Agent Loop Detection, Communication Hijacking, Identity Spoofing, Swarm Manipulation, and Intent Manipulation.

9.1 Agent Task-Planning & Recursion Budgets

Throttle recursive plans and force human checkpoints for privileged actions.

#9.1.1 Level: 1 Role: D/V
Verify that maximum recursion depth, breadth, wall-clock time, tokens, and monetary cost per agent execution are centrally configured and version-controlled.

#9.1.2 Level: 1 Role: D/V
Verify that privileged or irreversible actions (e.g., code commits, financial transfers) require explicit human approval via an auditable channel before execution.

#9.1.3 Level: 2 Role: D
Verify that real-time resource monitors trigger circuit-breaker interruption when any budget threshold is exceeded, halting further task expansion.

#9.1.4 Level: 2 Role: D/V
Verify that circuit-breaker events are logged with agent ID, triggering condition, and captured plan state for forensic review.

#9.1.5 Level: 3 Role: V
Verify that security tests cover budget-exhaustion and runaway-plan scenarios, confirming safe halting without data loss.

#9.1.6 Level: 3 Role: D
Verify that budget policies are expressed as policy-as-code and enforced in CI/CD to block configuration drift.

9.2 Tool Plugin Sandboxing

Isolate tool interactions to prevent unauthorized system access or code execution.

#9.2.1 Level: 1 Role: D/V
Verify that every tool/plugin executes inside an OS, container, or WASM-level sandbox with least-privilege file-system, network, and system-call policies.

#9.2.2 Level: 1 Role: D/V
Verify that sandbox resource quotas (CPU, memory, disk, network egress) and execution timeouts are enforced and logged.

#9.2.3 Level: 2 Role: D/V
Verify that tool binaries or descriptors are digitally signed; signatures are validated before loading.

#9.2.4 Level: 2 Role: V
Verify that sandbox telemetry streams to a SIEM; anomalies (e.g., attempted outbound connections) raise alerts.

#9.2.5 Level: 3 Role: V
Verify that high-risk plugins undergo security review and penetration testing before production deployment.

#9.2.6 Level: 3 Role: D/V
Verify that sandbox escape attempts are automatically blocked and the offending plugin is quarantined pending investigation.

9.3 Autonomous Loop & Cost Bounding

Detect and stop uncontrolled agent-to-agent recursion and cost explosions.

#9.3.1 Level: 1 Role: D/V
Verify that inter-agent calls include a hop-limit or TTL that the runtime decrements and enforces.

#9.3.2 Level: 2 Role: D
Verify that agents maintain a unique invocation-graph ID to spot self-invocation or cyclical patterns.

#9.3.3 Level: 2 Role: D/V
Verify that cumulative compute-unit and spend counters are tracked per request chain; breaching the limit aborts the chain.

#9.3.4 Level: 3 Role: V
Verify that formal analysis or model checking demonstrates absence of unbounded recursion in agent protocols.

#9.3.5 Level: 3 Role: D
Verify that loop-abort events generate alerts and feed continuous-improvement metrics.

9.4 Protocol-Level Misuse Protection

Secure communication channels between agents and external systems to prevent hijacking or manipulation.

#9.4.1 Level: 1 Role: D/V
Verify that all agent-to-tool and agent-to-agent messages are authenticated (e.g., mutual TLS or JWT) and end-to-end encrypted.

#9.4.2 Level: 1 Role: D
Verify that schemas are strictly validated; unknown fields or malformed messages are rejected.

#9.4.3 Level: 2 Role: D/V
Verify that integrity checks (MACs or digital signatures) cover the entire message payload including tool parameters.

#9.4.4 Level: 2 Role: D
Verify that replay-protection (nonces or timestamp windows) is enforced at the protocol layer.

#9.4.5 Level: 3 Role: V
Verify that protocol implementations undergo fuzzing and static analysis for injection or deserialization flaws.

9.5 Agent Identity & Tamper-Evidence

Ensure actions are attributable and modifications detectable.

#9.5.1 Level: 1 Role: D/V
Verify that each agent instance possesses a unique cryptographic identity (key-pair or hardware-rooted credential).

#9.5.2 Level: 2 Role: D/V
Verify that all agent actions are signed and timestamped; logs include the signature for non-repudiation.

#9.5.3 Level: 2 Role: V
Verify that tamper-evident logs are stored in an append-only or write-once medium.

#9.5.4 Level: 3 Role: D
Verify that identity keys rotate on a defined schedule and on compromise indicators.

#9.5.5 Level: 3 Role: D/V
Verify that spoofing or key-conflict attempts trigger immediate quarantine of the affected agent.

9.6 Multi-Agent Swarm Risk Reduction

Mitigate collective-behavior hazards through isolation and formal safety modeling.

#9.6.1 Level: 1 Role: D/V
Verify that agents operating in different security domains execute in isolated runtime sandboxes or network segments.

#9.6.2 Level: 3 Role: V
Verify that swarm behaviors are modeled and formally verified for liveness and safety before deployment.
#9.6.3 Level: 3 Role: D
Verify that runtime monitors detect emergent unsafe patterns (e.g., oscillations, deadlocks) and initiate corrective action.

9.7 User & Tool Authentication / Authorization

Implement robust access controls for every agent-triggered action.

#9.7.1 Level: 1 Role: D/V
Verify that agents authenticate as first-class principals to downstream systems, never reusing end-user credentials.
#9.7.2 Level: 2 Role: D
Verify that fine-grained authorization policies restrict which tools an agent may invoke and which parameters it may supply.
#9.7.3 Level: 2 Role: V
Verify that privilege checks are re-evaluated on every call (continuous authorization), not only at session start.
#9.7.4 Level: 3 Role: D
Verify that delegated privileges expire automatically and require re-consent after timeout or scope change.

9.8 Agent-to-Agent Communication Security

Encrypt and integrity-protect all inter-agent messages to thwart eavesdropping and tampering.

#9.8.1 Level: 1 Role: D/V
Verify that mutual authentication and perfect-forward-secret encryption (e.g. TLS 1.3) are mandatory for agent channels.
#9.8.2 Level: 1 Role: D
Verify that message integrity and origin are validated before processing; failures raise alerts and drop the message.
#9.8.3 Level: 2 Role: D/V
Verify that communication metadata (timestamps, sequence numbers) is logged to support forensic reconstruction.
#9.8.4 Level: 3 Role: V
Verify that formal verification or model checking confirms that protocol state machines cannot be driven into unsafe states.

9.9 Intent Verification & Constraint Enforcement

Validate that agent actions align with the user's stated intent and system constraints.

#9.9.1 Level: 1 Role: D
Verify that pre-execution constraint solvers check proposed actions against hard-coded safety and policy rules.

#9.9.2 Level: 2 Role: D/V
Verify that high-impact actions (financial, destructive, privacy-sensitive) require explicit intent confirmation from the initiating user.

#9.9.3 Level: 2 Role: V
Verify that post-condition checks validate that completed actions achieved intended effects without side effects; discrepancies trigger rollback.

#9.9.4 Level: 3 Role: V
Verify that formal methods (e.g., model checking, theorem proving) or property-based tests demonstrate that agent plans satisfy all declared constraints.

#9.9.5 Level: 3 Role: D
Verify that intent-mismatch or constraint-violation incidents feed continuous-improvement cycles and threat-intel sharing.

9.10 Agent Reasoning Strategy Security

Secure selection and execution of different reasoning strategies including ReAct, Chain-of-Thought, and Tree-of-Thoughts approaches.

#9.10.1 Level: 1 Role: D/V
Verify that reasoning strategy selection uses deterministic criteria (input complexity, task type, security context) and identical inputs produce identical strategy selections within the same security context.

#9.10.2 Level: 1 Role: D/V
Verify that each reasoning strategy (ReAct, Chain-of-Thought, Tree-of-Thoughts) has dedicated input validation, output sanitization, and execution time limits specific to its cognitive approach.

#9.10.3 Level: 2 Role: D/V
Verify that reasoning strategy transitions are logged with complete context including input characteristics, selection criteria values, and execution metadata for audit trail reconstruction.

#9.10.4 Level: 2 Role: D/V
Verify that Tree-of-Thoughts reasoning includes branch pruning mechanisms that terminate exploration when policy violations, resource limits, or safety boundaries are detected.

#9.10.5 Level: 2 Role: D/V
Verify that ReAct (Reason-Act-Observe) cycles include validation checkpoints at each phase: reasoning step verification, action authorization, and observation sanitization before proceeding.

#9.10.6 Level: 3 Role: D/V
Verify that reasoning strategy performance metrics (execution time, resource usage, output quality) are

monitored with automated alerts when metrics deviate beyond configured thresholds.

#9.10.7 Level: 3 Role: D/V

Verify that hybrid reasoning approaches that combine multiple strategies maintain input validation and output constraints of all constituent strategies without bypassing any security controls.

#9.10.8 Level: 3 Role: D/V

Verify that reasoning strategy security testing includes fuzzing with malformed inputs, adversarial prompts designed to force strategy switching, and boundary condition testing for each cognitive approach.

9.11 Agent Lifecycle State Management & Security

Secure agent initialization, state transitions, and termination with cryptographic audit trails and defined recovery procedures.

#9.11.1 Level: 1 Role: D/V

Verify that agent initialization includes cryptographic identity establishment with hardware-backed credentials and immutable startup audit logs containing agent ID, timestamp, configuration hash, and initialization parameters.

#9.11.2 Level: 2 Role: D/V

Verify that agent state transitions are cryptographically signed, timestamped, and logged with complete context including triggering events, previous state hash, new state hash, and security validations performed.

#9.11.3 Level: 2 Role: D/V

Verify that agent shutdown procedures include secure memory wiping using cryptographic erasure or multi-pass overwriting, credential revocation with certificate authority notification, and generation of tamper-evident termination certificates.

#9.11.4 Level: 3 Role: D/V

Verify that agent recovery mechanisms validate state integrity using cryptographic checksums (SHA-256 minimum) and rollback to known-good states when corruption is detected with automated alerts and manual approval requirements.

#9.11.5 Level: 3 Role: D/V

Verify that agent persistence mechanisms encrypt sensitive state data with per-agent AES-256 keys and implement secure key rotation on configurable schedules (maximum 90 days) with zero-downtime deployment.

9.12 Tool Integration Security Framework

Security controls for dynamic tool loading, execution, and result validation with defined risk assessment and approval processes.

#9.12.1 Level: 1 Role: D/V

Verify that tool descriptors include security metadata specifying required privileges (read/write/execute), risk levels (low/medium/high), resource limits (CPU, memory, network), and validation requirements documented in tool manifests.

#9.12.2 Level: 1 Role: D/V

Verify that tool execution results are validated against expected schemas (JSON Schema, XML Schema) and security policies (output sanitization, data classification) before integration with timeout limits and error handling procedures.

#9.12.3 Level: 2 Role: D/V

Verify that tool interaction logs include detailed security context including privilege usage, data access patterns, execution time, resource consumption, and return codes with structured logging for SIEM integration.

#9.12.4 Level: 2 Role: D/V

Verify that dynamic tool loading mechanisms validate digital signatures using PKI infrastructure and implement secure loading protocols with sandbox isolation and permission verification before execution.

#9.12.5 Level: 3 Role: D/V

Verify that tool security assessments are automatically triggered for new versions with mandatory approval gates including static analysis, dynamic testing, and security team review with documented approval criteria and SLA requirements.

C9.13 Model Context Protocol (MCP) Security

Ensure secure discovery, authentication, authorization, transport, and use of MCP-based tool and resource integrations to prevent context confusion, unauthorized tool invocation, or cross-tenant data exposure.

Component Integrity & Supply Chain Hygiene

#9.13.1 Level: 1 Role: D/V

Verify that MCP server, client, and tool implementations are manually reviewed or automatically analyzed to identify insecure function exposure, unsafe defaults, missing authentication, or missing input validation.

#9.13.2 Level: 1 Role: D/V

Verify that external or open-source MCP servers or packages undergo automated vulnerability and supply-chain scanning (e.g., SCA) before integration, and that components with known critical vulnerabilities are not used.

#9.13.3 Level: 1 Role: D/V

Verify that MCP server and client components are obtained only from trusted sources and verified using signatures, checksums, or secure package metadata, rejecting tampered or unsigned builds.

Authentication & Authorization

#9.13.4 Level: 2 Role: D/V

Verify that MCP clients and servers mutually authenticate using strong, non-user credentials (e.g., mTLS,

signed tokens, or platform-issued identities), and that unauthenticated MCP endpoints are rejected.

#9.13.5 Level: 2 Role: D/V

Verify that MCP servers are registered through a controlled technical onboarding mechanism requiring explicit owner, environment, and resource definitions; unregistered or undiscoverable servers must not be callable in production.

#9.13.6 Level: 2 Role: D/V

Verify that each MCP tool or resource defines explicit authorization scopes (e.g., read-only, restricted queries, side-effect levels), and that agents cannot invoke MCP functions outside their assigned scope.

Secure Transport & Network Boundary Protection

#9.13.7 Level: 2 Role: D/V

Verify that authenticated, encrypted streamable-HTTP is used as the primary MCP transport in production environments; alternate transports (stdio, SSE) are restricted to local or tightly controlled environments with explicit justification.

#9.13.8 Level: 2 Role: D/V

Verify that streamable-HTTP MCP transports use authenticated, encrypted channels (TLS 1.3 or later) with certificate validation and forward secrecy to ensure confidentiality and integrity of streamed MCP messages.

#9.13.9 Level: 2 Role: D/V

Verify that SSE-based MCP transports are used only within private, authenticated internal channels and enforce TLS, authentication, schema validation, payload size limits, and rate limiting; SSE endpoints must not be exposed to the public internet.

#9.13.10 Level: 2 Role: D/V

Verify that MCP servers validate the `Origin` and `Host` headers on all HTTP-based transports (including SSE and streamable-HTTP) to prevent DNS rebinding attacks, and reject requests from untrusted, mismatched, or missing origins.

Schema, Message, and Input Validation

#9.13.11 Level: 2 Role: D/V

Verify that MCP tool and resource schemas (e.g., JSON schemas or capability descriptors) are validated for authenticity and integrity using signatures, checksums, or server attestation to prevent schema tampering or malicious parameter modification.

#9.13.12 Level: 2 Role: D/V

Verify that all MCP transports enforce message-framing integrity, strict schema validation, maximum payload sizes, and rejection of malformed, truncated, or interleaved frames to prevent desynchronization or injection attacks.

#9.13.13 Level: 2 Role: D/V

Verify that MCP servers perform strict input validation for all function calls, including type checking, boundary checking, enumeration enforcement, and rejection of unrecognized or oversized parameters.

Outbound Access & Agent Execution Safety

#9.13.14 Level: 2 Role: D/V

Verify that MCP servers may only initiate outbound requests to approved internal or external destinations following least-privilege egress policies, and cannot access arbitrary network targets or internal cloud metadata services.

#9.13.15 Level: 2 Role: D/V

Verify that outbound MCP actions implement execution limits (timeouts, recursion limits, concurrency caps, circuit breakers) to prevent unbounded agent-driven tool invocation or chained side effects.

#9.13.16 Level: 2 Role: D/V

Verify that MCP request and response metadata (server ID, resource name, tool name, session identifier, tenant, environment) is logged with integrity protection and correlated to agent activity for forensic analysis.

Transport Restrictions & High-Risk Boundary Controls

#9.13.17 Level: 3 Role: D/V

Verify that stdio-based MCP transports are limited to co-located, single-process development scenarios, isolated from shell execution, terminal injection, and process-spawning capabilities; stdio must never cross network or multi-tenant boundaries.

#9.13.18 Level: 3 Role: D/V

Verify that MCP servers expose only allow-listed functions and resources, and prohibit dynamic dispatch, reflective invocation, or execution of function names influenced by user or model-provided input.

#9.13.19 Level: 3 Role: D/V

Verify that tenant boundaries, environment boundaries (dev/test/prod), and data domain boundaries are enforced at the MCP layer, preventing cross-tenant or cross-environment server or resource discovery.

References

- MITRE ATLAS tactics ML09
- Circuit-breaker research for AI agents — Zou et al., 2024
- Trend Micro analysis of sandbox escapes in AI agents — Park, 2025
- Auth0 guidance on human-in-the-loop authorization for agents — Martinez, 2025
- Medium deep-dive on MCP & A2A protocol hijacking — ForAISec, 2025
- Rapid7 fundamentals on spoofing attack prevention — 2024
- Imperial College verification of swarm systems — Lomuscio et al.
- NIST AI Risk Management Framework 1.0, 2023
- WIRED security briefing on encryption best practices, 2024
- OWASP Top 10 for LLM Applications, 2025
- Comprehensive Vulnerability Analysis is Necessary for Trustworthy LLM-MAS
- [How Is LLM Reasoning Distracted by Irrelevant Context?

An Analysis Using a Controlled Benchmark](https://www.arxiv.org/pdf/2505.18761)

- Large Language Model Sentinel: LLM Agent for Adversarial Purification
- Securing Agentic AI: A Comprehensive Threat Model and Mitigation Framework for Generative AI Agents
- Model Context Protocol Specification

- Model Context Protocol Tools & Resources Specification
- Model Context Protocol Transport Documentation
- OWASP GenAI Security Project — "A Practical Guide for Securely Using Third-Party MCP Servers 1.0"
- Cloud Security Alliance – Model Context Protocol Security Working Group
- CSA MCP Security: Top 10 Risks
- CSA MCP Security: TTPs & Hardening Guidance

10 Adversarial Robustness & Privacy Defense

Control Objective

Ensure that AI models remain reliable, privacy-preserving, and abuse-resistant when facing evasion, inference, extraction, or poisoning attacks.

10.1 Model Alignment & Safety

Guard against harmful or policy-breaking outputs.

#10.1.1 Level: 1 Role: D/V
Verify that an alignment test-suite (red-team prompts, jailbreak probes, disallowed content) is version-controlled and run on every model release.

#10.1.2 Level: 1 Role: D
Verify that refusal and safe-completion guard-rails are enforced.

#10.1.3 Level: 2 Role: D/V
Verify that an automated evaluator measures harmful-content rate and flags regressions beyond a set threshold.

#10.1.4 Level: 2 Role: D
Verify that counter-jailbreak training is documented and reproducible.

#10.1.5 Level: 3 Role: V
Verify that formal policy-compliance proofs or certified monitoring cover critical domains.

10.2 Adversarial-Example Hardening

Increase resilience to manipulated inputs. Robust adversarial-training and benchmark scoring are the current best practice.

#10.2.1 Level: 1 Role: D
Verify that project repositories include adversarial-training configurations with reproducible seeds.

#10.2.2 Level: 2 Role: D/V

Verify that adversarial-example detection raises blocking alerts in production pipelines.

#10.2.4 Level: 3 Role: V

Verify that certified-robustness proofs or interval-bound certificates cover at least the top critical classes.

#10.2.5 Level: 3 Role: V

Verify that regression tests use adaptive attacks to confirm no measurable robustness loss.

10.3 Membership-Inference Mitigation

Limit the ability to decide whether a record was in training data. Differential privacy and confidence-score masking remain the most effective known defenses.

#10.3.1 Level: 1 Role: D

Verify that per-query entropy regularisation or temperature-scaling reduces overconfident predictions.

#10.3.2 Level: 2 Role: D

Verify that training employs ε-bounded differentially-private optimization for sensitive datasets.

#10.3.3 Level: 2 Role: V

Verify that attack simulations (shadow-model or black-box) show attack AUC ≤ 0.60 on held-out data.

10.4 Model-Inversion Resistance

Prevent reconstruction of private attributes. Recent surveys emphasize output truncation and DP guarantees as practical defenses.

#10.4.1 Level: 1 Role: D

Verify that sensitive attributes are never directly output; where needed, use buckets or one-way transforms.

#10.4.2 Level: 1 Role: D/V

Verify that query-rate limits throttle repeated adaptive queries from the same principal.

#10.4.3 Level: 2 Role: D

Verify that the model is trained with privacy-preserving noise.

10.5 Model-Extraction Defense

Detect and deter unauthorized cloning. Watermarking and query-pattern analysis are recommended.

#10.5.1 Level: 1 Role: D

Verify that inference gateways enforce global and per-API-key rate limits tuned to the model's memorization threshold.

#10.5.2 Level: 2 Role: D/V
Verify that query-entropy and input-plurality statistics feed an automated extraction detector.

#10.5.3 Level: 2 Role: V
Verify that fragile or probabilistic watermarks can be proved with $p < 0.01$ in $\leq 1\,000$ queries against a suspected clone.

#10.5.4 Level: 3 Role: D
Verify that watermark keys and trigger sets are stored in a hardware-security-module and rotated yearly.

#10.5.5 Level: 3 Role: V
Verify that extraction-alert events include offending queries and are integrated with incident-response playbooks.

10.6 Inference-Time Poisoned-Data Detection

Identify and neutralize backdoored or poisoned inputs.

#10.6.1 Level: 1 Role: D
Verify that inputs pass through an anomaly detector (e.g., STRIP, consistency-scoring) before model inference.

#10.6.2 Level: 1 Role: V
Verify that detector thresholds are tuned on clean/poisoned validation sets to achieve less that 5% false positives.

#10.6.3 Level: 2 Role: D
Verify that inputs flagged as poisoned trigger soft-blocking and human review workflows.

#10.6.4 Level: 2 Role: V
Verify that detectors are stress-tested with adaptive, triggerless backdoor attacks.

#10.6.5 Level: 3 Role: D
Verify that detection efficacy metrics are logged and periodically re-evaluated with fresh threat intel.

10.7 Dynamic Security Policy Adaptation

Real-time security policy updates based on threat intelligence and behavioral analysis.

#10.7.1 Level: 1 Role: D/V
Verify that security policies can be updated dynamically without agent restart while maintaining policy version integrity.

#10.7.2 Level: 2 Role: D/V
Verify that policy updates are cryptographically signed by authorized security personnel and validated before application.

#10.7.3 Level: 2 Role: D/V
Verify that dynamic policy changes are logged with full audit trails including justification, approval chains, and rollback procedures.

#10.7.4 Level: 3 Role: D/V
Verify that adaptive security mechanisms adjust threat detection sensitivity based on risk context and behavioral patterns.

#10.7.5 Level: 3 Role: D/V
Verify that policy adaptation decisions are explainable and include evidence trails for security team review.

10.8 Reflection-Based Security Analysis

Security validation through agent self-reflection and meta-cognitive analysis.

#10.8.1 Level: 1 Role: D/V
Verify that agent reflection mechanisms include security-focused self-assessment of decisions and actions.

#10.8.2 Level: 2 Role: D/V
Verify that reflection outputs are validated to prevent manipulation of self-assessment mechanisms by adversarial inputs.

#10.8.3 Level: 2 Role: D/V
Verify that meta-cognitive security analysis identifies potential bias, manipulation, or compromise in agent reasoning processes.

#10.8.4 Level: 3 Role: D/V
Verify that reflection-based security warnings trigger enhanced monitoring and potential human intervention workflows.

#10.8.5 Level: 3 Role: D/V
Verify that continuous learning from security reflections improves threat detection without degrading legitimate functionality.

10.9 Evolution & Self-Improvement Security

Security controls for agent systems capable of self-modification and evolution.

#10.9.1 Level: 1 Role: D/V
Verify that self-modification capabilities are restricted to designated safe areas with formal verification boundaries.

#10.9.2 Level: 2 Role: D/V
Verify that evolution proposals undergo security impact assessment before implementation.

#10.9.3 Level: 2 Role: D/V
Verify that self-improvement mechanisms include rollback capabilities with integrity verification.

#10.9.4 Level: 3 Role: D/V
Verify that meta-learning security prevents adversarial manipulation of improvement algorithms.
#10.9.5 Level: 3 Role: D/V
Verify that recursive self-improvement is bounded by formal safety constraints with mathematical proofs of convergence.

References

- MITRE ATLAS adversary tactics for ML
- NIST AI Risk Management Framework 1.0, 2023
- OWASP Top 10 for LLM Applications, 2025
- Adversarial Training: A Survey — Zhao et al., 2024
- RobustBench adversarial-robustness benchmark
- Membership-Inference & Model-Inversion Risk Survey, 2025
- PURIFIER: Confidence-Score Defense against MI Attacks — AAAI 2023
- Model-Inversion Attacks & Defenses Survey — AI Review, 2025
- Comprehensive Defense Framework Against Model Extraction — IEEE TDSC 2024
- Fragile Model Watermarking Survey — 2025
- Data Poisoning in Deep Learning: A Survey — Zhao et al., 2025
- BDetCLIP: Multimodal Prompting Backdoor Detection — Niu et al., 2024

11 Privacy Protection & Personal Data Management

Control Objective

Maintain rigorous privacy assurances across the entire AI lifecycle—collection, training, inference, and incident response—so that personal data is only processed with clear consent, minimum necessary scope, provable erasure, and formal privacy guarantees.

11.1 Anonymization & Data Minimization

#11.1.1 Level: 1 Role: D/V
Verify that direct and quasi-identifiers are removed, hashed.

#11.1.2 Level: 2 Role: D/V
Verify that automated audits measure k-anonymity/l-diversity and alert when thresholds drop below policy.

#11.1.3 Level: 2 Role: V
Verify that model feature-importance reports prove no identifier leakage beyond $\varepsilon = 0.01$ mutual information.

#11.1.4 Level: 3 Role: V
Verify that formal proofs or synthetic-data certification show re-identification risk ≤ 0.05 even under linkage attacks.

11.2 Right-to-be-Forgotten & Deletion Enforcement

#11.2.1 Level: 1 Role: D/V
Verify that data-subject deletion requests propagate to raw datasets, checkpoints, embeddings, logs, and backups within service level agreements of less than 30 days.

#11.2.2 Level: 2 Role: D
Verify that "machine-unlearning" routines physically re-train or approximate removal using certified unlearning algorithms.

#11.2.3 Level: 2 Role: V
Verify that shadow-model evaluation proves forgotten records influence less than 1% of outputs after unlearning.

#11.2.4 Level: 3 Role: V
Verify that deletion events are immutably logged and auditable for regulators.

11.3 Differential-Privacy Safeguards

#11.3.1 Level: 2 Role: D/V
Verify that privacy-loss accounting dashboards alert when cumulative ε exceeds policy thresholds.
#11.3.2 Level: 2 Role: V
Verify that black-box privacy audits estimate ε̂ within 10% of declared value.
#11.3.3 Level: 3 Role: V
Verify that formal proofs cover all post-training fine-tunes and embeddings.

11.4 Purpose-Limitation & Scope-Creep Protection

#11.4.1 Level: 1 Role: D
Verify that every dataset and model checkpoint carries a machine-readable purpose tag aligned to the original consent.
#11.4.2 Level: 1 Role: D/V
Verify that runtime monitors detect queries inconsistent with declared purpose and trigger soft refusal.
#11.4.3 Level: 3 Role: D
Verify that policy-as-code gates block redeployment of models to new domains without DPIA review.
#11.4.4 Level: 3 Role: V
Verify that formal traceability proofs show every personal data lifecycle remains within consented scope.

11.5 Consent Management & Lawful-Basis Tracking

#11.5.1 Level: 1 Role: D/V
Verify that a Consent-Management Platform (CMP) records opt-in status, purpose, and retention period per data-subject.
#11.5.2 Level: 2 Role: D
Verify that APIs expose consent tokens; models must validate token scope before inference.
#11.5.3 Level: 2 Role: D/V
Verify that denied or withdrawn consent halts processing pipelines within 24 hours.

11.6 Federated Learning with Privacy Controls

#11.6.1 Level: 1 Role: D
Verify that client updates employ local differential privacy noise addition before aggregation.

#11.6.2 Level: 2 Role: D/V
Verify that training metrics are differentially private and never reveal single-client loss.

#11.6.3 Level: 2 Role: V
Verify that poisoning-resistant aggregation (e.g., Krum/Trimmed-Mean) is enabled.

#11.6.4 Level: 3 Role: V
Verify that formal proofs demonstrate overall ε budget with less than 5 utility loss.

References

- GDPR & AI Compliance Best Practices
- EU Parliament Study on GDPR & AI, 2020
- ISO 31700-1:2023 — Privacy by Design for Consumer Products
- NIST Privacy Framework 1.1 (2025 Draft)
- Machine Unlearning: Right-to-Be-Forgotten Techniques
- A Survey of Machine Unlearning, 2024
- Auditing DP-SGD — ArXiv 2024
- DP-SGD Explained — PyTorch Blog
- Purpose-Limitation for AI — IJLIT 2025
- Data-Protection Considerations for AI — URM Consulting
- Top Consent-Management Platforms, 2025
- Secure Aggregation in DP Federated Learning — ArXiv 2024

C12 Monitoring, Logging & Anomaly Detection

Control Objective

This section provides requirements for delivering real-time and forensic visibility into what the model and other AI components see, do, and return, so threats can be detected, triaged, and learned from.

C12.1 Request & Response Logging

#12.1.1 Level: 1 Role: D/V
Verify that all user prompts and model responses are logged with appropriate metadata (e.g. timestamp, user ID, session ID, model version).

#12.1.2 Level: 1 Role: D/V
Verify that logs are stored in secure, access-controlled repositories with appropriate retention policies and backup procedures.

#12.1.3 Level: 1 Role: D/V
Verify that log storage systems implement encryption at rest and in transit to protect sensitive information contained in logs.

#12.1.4 Level: 1 Role: D/V
Verify that sensitive data in prompts and outputs is automatically redacted or masked before logging, with configurable redaction rules for PII, credentials, and proprietary information.

#12.1.5 Level: 2 Role: D/V
Verify that policy decisions and safety filtering actions are logged with sufficient detail to enable audit and debugging of content moderation systems.

#12.1.6 Level: 2 Role: D/V
Verify that log integrity is protected through e.g. cryptographic signatures or write-only storage.

C12.2 Abuse Detection and Alerting

#12.2.1 Level: 1 Role: D/V
Verify that the system detects and alerts on known jailbreak patterns, prompt injection attempts, and adversarial inputs using signature-based detection.

#12.2.2 Level: 1 Role: D/V

Verify that the system integrates with existing Security Information and Event Management (SIEM) platforms using standard log formats and protocols.

#12.2.3 Level: 2 Role: D/V
Verify that enriched security events include AI-specific context such as model identifiers, confidence scores, and safety filter decisions.

#12.2.4 Level: 2 Role: D/V
Verify that behavioral anomaly detection identifies unusual conversation patterns, excessive retry attempts, or systematic probing behaviors.

#12.2.5 Level: 2 Role: D/V
Verify that real-time alerting mechanisms notify security teams when potential policy violations or attack attempts are detected.

#12.2.6 Level: 2 Role: D/V
Verify that custom rules are included to detect AI-specific threat patterns including coordinated jailbreak attempts, prompt injection campaigns, and model extraction attacks.

#12.2.7 Level: 3 Role: D/V
Verify that automated incident response workflows can isolate compromised models, block malicious users, and escalate critical security events.

C12.3 Model Drift Detection

#12.3.1 Level: 1 Role: D/V
Verify that the system tracks basic performance metrics such as accuracy, confidence scores, latency, and error rates across model versions and time periods.

#12.3.2 Level: 2 Role: D/V
Verify that automated alerting triggers when performance metrics exceed predefined degradation thresholds or deviate significantly from baselines.

#12.3.3 Level: 2 Role: D/V
Verify that hallucination detection monitors identify and flag instances when model outputs contain factually incorrect, inconsistent, or fabricated information.

C12.4 Performance & Behavior Telemetry

#12.4.1 Level: 1 Role: D/V
Verify that operational metrics including request latency, token consumption, memory usage, and throughput are continuously collected and monitored.

#12.4.2 Level: 1 Role: D/V
Verify that success and failure rates are tracked with categorization of error types and their root causes.

#12.4.3 Level: 2 Role: D/V
Verify that resource utilization monitoring includes GPU/CPU usage, memory consumption, and storage requirements with alerting on threshold breaches.

C12.5 AI Incident Response Planning & Execution

#12.5.1 Level: 1 Role: D/V
Verify that incident response plans specifically address AI-related security events including model compromise, data poisoning, and adversarial attacks.

#12.5.2 Level: 2 Role: D/V
Verify that incident response teams have access to AI-specific forensic tools and expertise to investigate model behavior and attack vectors.

#12.5.3 Level: 3 Role: D/V
Verify that post-incident analysis includes model retraining considerations, safety filter updates, and lessons learned integration into security controls.

C12.6 AI Performance Degradation Detection

Monitor and detect degradation in AI model performance and quality over time.

#12.6.1 Level: 1 Role: D/V
Verify that model accuracy, precision, recall, and F1 scores are continuously monitored and compared against baseline thresholds.

#12.6.2 Level: 1 Role: D/V
Verify that data drift detection monitors input distribution changes that may impact model performance.

#12.6.3 Level: 2 Role: D/V
Verify that concept drift detection identifies changes in the relationship between inputs and expected outputs.

#12.6.4 Level: 2 Role: D/V
Verify that performance degradation triggers automated alerts and initiates model retraining or replacement workflows.

#12.6.5 Level: 3 Role: V
Verify that degradation root cause analysis correlates performance drops with data changes, infrastructure issues, or external factors.

C12.7 DAG Visualization & Workflow Security

Protect workflow visualization systems from information leakage and manipulation attacks.

#12.7.1 Level: 1 Role: D/V
Verify that DAG visualization data is sanitized to remove sensitive information before storage or transmis-

sion.

#12.7.2 Level: 1 Role: D/V
Verify that workflow visualization access controls ensure only authorized users can view agent decision paths and reasoning traces.

#12.7.3 Level: 2 Role: D/V
Verify that DAG data integrity is protected through cryptographic signatures and tamper-evident storage mechanisms.

#12.7.4 Level: 2 Role: D/V
Verify that workflow visualization systems implement input validation to prevent injection attacks through crafted node or edge data.

#12.7.5 Level: 3 Role: D/V
Verify that real-time DAG updates are rate-limited and validated to prevent denial-of-service attacks on visualization systems.

C12.8 Proactive Security Behavior Monitoring

Detection and prevention of security threats through proactive agent behavior analysis.

#12.8.1 Level: 1 Role: D/V
Verify that proactive agent behaviors are security-validated before execution with risk assessment integration.

#12.8.2 Level: 2 Role: D/V
Verify that autonomous initiative triggers include security context evaluation and threat landscape assessment.

#12.8.3 Level: 2 Role: D/V
Verify that proactive behavior patterns are analyzed for potential security implications and unintended consequences.

#12.8.4 Level: 3 Role: D/V
Verify that security-critical proactive actions require explicit approval chains with audit trails.

#12.8.5 Level: 3 Role: D/V
Verify that behavioral anomaly detection identifies deviations in proactive agent patterns that may indicate compromise.

References

- NIST AI Risk Management Framework 1.0 - Manage 4.1 and 4.3
- ISO/IEC 42001:2023 — AI Management Systems Requirements - Annex B 6.2.6

C13 Human Oversight, Accountability & Governance

Control Objective

This chapter provides requirements for maintaining human oversight and clear accountability chains in AI systems, ensuring explainability, transparency, and ethical stewardship throughout the AI lifecycle.

C13.1 Kill-Switch & Override Mechanisms

Provide shutdown or rollback paths when unsafe behavior of the AI system is observed.

#13.1.1 Level: 1 Role: D/V
Verify that a manual kill-switch mechanism exists to immediately halt AI model inference and outputs.

#13.1.2 Level: 1 Role: D
Verify that override controls are accessible to only to authorized personnel.

#13.1.3 Level: 3 Role: D/V
Verify that rollback procedures can revert to previous model versions or safe-mode operations.

#13.1.4 Level: 3 Role: V
Verify that override mechanisms are tested regularly.

C13.2 Human-in-the-Loop Decision Checkpoints

Require human approvals when stakes surpass predefined risk thresholds.

#13.2.1 Level: 1 Role: D/V
Verify that high-risk AI decisions require explicit human approval before execution.

#13.2.2 Level: 1 Role: D
Verify that risk thresholds are clearly defined and automatically trigger human review workflows.

#13.2.3 Level: 2 Role: D
Verify that time-sensitive decisions have fallback procedures when human approval cannot be obtained within required timeframes.

#13.2.4 Level: 3 Role: D/V
Verify that escalation procedures define clear authority levels for different decision types or risk categories, if applicable.

C13.3 Chain of Responsibility & Auditability

Log operator actions and model decisions.

#13.3.1 Level: 1 Role: D/V
Verify that all AI system decisions and human interventions are logged with timestamps, user identities, and decision rationale.
#13.3.2 Level: 2 Role: D
Verify that audit logs cannot be tampered with and include integrity verification mechanisms.

C13.4 Explainable-AI Techniques

Surface feature importance, counter-factuals, and local explanations.

#13.4.1 Level: 1 Role: D/V
Verify that AI systems provide basic explanations for their decisions in human-readable format.
#13.4.2 Level: 2 Role: V
Verify that explanation quality is validated through human evaluation studies and metrics.
#13.4.3 Level: 3 Role: D/V
Verify that feature importance scores or attribution methods (SHAP, LIME, etc.) are available for critical decisions.
#13.4.4 Level: 3 Role: V
Verify that counterfactual explanations show how inputs could be modified to change outcomes, if applicable to the use case and domain.

C13.5 Model Cards & Usage Disclosures

Maintain model cards for intended use, performance metrics, and ethical considerations.

#13.5.1 Level: 1 Role: D
Verify that model cards document intended use cases, limitations, and known failure modes.
#13.5.2 Level: 1 Role: D/V
Verify that performance metrics across different applicable use cases are disclosed.

#13.5.3 Level: 2 Role: D
Verify that ethical considerations, bias assessments, fairness evaluations, training data characteristics, and known training data limitations are documented and updated regularly.
#13.5.4 Level: 2 Role: D/V
Verify that model cards are version-controlled and maintained throughout the model lifecycle with change tracking.

C13.6 Uncertainty Quantification

Propagate confidence scores or entropy measures in responses.

#13.6.1 Level: 1 Role: D
Verify that AI systems provide confidence scores or uncertainty measures with their outputs.
#13.6.2 Level: 2 Role: D/V
Verify that uncertainty thresholds trigger additional human review or alternative decision pathways.
#13.6.3 Level: 2 Role: V
Verify that uncertainty quantification methods are calibrated and validated against ground truth data.
#13.6.4 Level: 3 Role: D/V
Verify that uncertainty propagation is maintained through multi-step AI workflows.

C13.7 User-Facing Transparency Reports

Provide periodic disclosures on incidents, drift, and data usage.

#13.7.1 Level: 1 Role: D/V
Verify that data usage policies and user consent management practices are clearly communicated to stakeholders.
#13.7.2 Level: 2 Role: D/V
Verify that AI impact assessments are conducted and results are included in reporting.
#13.7.3 Level: 2 Role: D/V
Verify that transparency reports published regularly disclose AI incidents and operational metrics in reasonable detail.

References

- EU Artificial Intelligence Act — Regulation (EU) 2024/1689 (Official Journal, 12 July 2024)
- ISO/IEC 23894:2023 — Artificial Intelligence — Guidance on Risk Management
- ISO/IEC 42001:2023 — AI Management Systems Requirements
- NIST AI Risk Management Framework 1.0
- NIST SP 800-53 Revision 5 — Security and Privacy Controls

- A Unified Approach to Interpreting Model Predictions (SHAP, ICML 2017)
- Model Cards for Model Reporting (Mitchell et al., 2018)
- Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning (Gal & Ghahramani, 2016)
- ISO/IEC 24029-2:2023 — Robustness of Neural Networks — Methodology for Formal Methods
- IEEE 7001-2021 — Transparency of Autonomous Systems
- Human Oversight under Article 14 of the EU AI Act (Fink, 2025)

Appendix A: Glossary

This comprehensive glossary provides definitions of key AI, ML, and security terms used throughout the AISVS to ensure clarity and common understanding.

- Adversarial Example: An input deliberately crafted to cause an AI model to make a mistake, often by adding subtle perturbations imperceptible to humans.
- Adversarial Robustness – Adversarial robustness in AI refers to a model's ability to maintain its performance and resist being fooled or manipulated by intentionally crafted, malicious inputs designed to cause errors.
- Agent – AI agents are software systems that use AI to pursue goals and complete tasks on behalf of users. They show reasoning, planning, and memory and have a level of autonomy to make decisions, learn, and adapt.
- Agentic AI: AI systems that can operate with some degree of autonomy to achieve goals, often making decisions and taking actions without direct human intervention.
- Attribute-Based Access Control (ABAC): An access control paradigm where authorization decisions are based on attributes of the user, resource, action, and environment, evaluated at query time.
- Backdoor Attack: A type of data poisoning attack where the model is trained to respond in a specific way to certain triggers while behaving normally otherwise.
- Bias: Systematic errors in AI model outputs that can lead to unfair or discriminatory outcomes for certain groups or in specific contexts.
- Bias Exploitation: An attack technique that takes advantage of known biases in AI models to manipulate outputs or outcomes.
- Cedar: Amazon's policy language and engine for fine-grained permissions used in implementing ABAC for AI systems.
- Chain of Thought: A technique for improving reasoning in language models by generating in-

termediate reasoning steps before producing a final answer.

- Circuit Breakers: Mechanisms that automatically halt AI system operations when specific risk thresholds are exceeded.

- Confidential Inference Service: An inference service that runs AI models inside a trusted execution environment (TEE) or equivalent confidential computing mechanism, ensuring model weights and inference data remain encrypted, sealed, and protected from unauthorized access or tampering.

- Confidential Workload: An AI workload (e.g., training, inference, preprocessing) executed inside a trusted execution environment (TEE) with hardware-enforced isolation, memory encryption, and remote attestation to protect code, data, and models from host or co-tenant access.

- Data Leakage: Unintended exposure of sensitive information through AI model outputs or behavior.

- Data Poisoning: The deliberate corruption of training data to compromise model integrity, often to install backdoors or degrade performance.

- Differential Privacy – Differential privacy is a mathematically rigorous framework for releasing statistical information about datasets while protecting the privacy of individual data subjects. It enables a data holder to share aggregate patterns of the group while limiting information that is leaked about specific individuals.

- Embeddings: Dense vector representations of data (text, images, etc.) that capture semantic meaning in a high-dimensional space.

- Explainability – Explainability in AI is the ability of an AI system to provide human-understandable reasons for its decisions and predictions, offering insights into its internal workings.

- Explainable AI (XAI): AI systems designed to provide human-understandable explanations for their decisions and behaviors through various techniques and frameworks.

- Federated Learning: A machine learning approach where models are trained across multiple decentralized devices holding local data samples, without exchanging the data itself.

- Formulation: The recipe or method used to produce an artifact or dataset, such as hyperparameters, training configuration, preprocessing steps, or build scripts.

- Guardrails: Constraints implemented to prevent AI systems from producing harmful, biased,

or otherwise undesirable outputs.

- Hallucination – An AI hallucination refers to a phenomenon where an AI model generates incorrect or misleading information that is not based on its training data or factual reality.

- Human-in-the-Loop (HITL): Systems designed to require human oversight, verification, or intervention at crucial decision points.

- Infrastructure as Code (IaC): Managing and provisioning infrastructure through code instead of manual processes, enabling security scanning and consistent deployments.

- Jailbreak: Techniques used to circumvent safety guardrails in AI systems, particularly in large language models, to produce prohibited content.

- Least Privilege: The security principle of granting only the minimum necessary access rights for users and processes.

- LIME (Local Interpretable Model-agnostic Explanations): A technique to explain the predictions of any machine learning classifier by approximating it locally with an interpretable model.

- MCP (Model Context Protocol): A protocol that enables AI models and agents to access external tools, data sources, and resources by exchanging structured, typed requests and responses over a defined transport.

- Membership Inference Attack: An attack that aims to determine whether a specific data point was used to train a machine learning model.

- MITRE ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems; a knowledge base of adversarial tactics and techniques against AI systems.

- Model Card – A model card is a document that provides standardized information about an AI model's performance, limitations, intended uses, and ethical considerations to promote transparency and responsible AI development.

- Model Extraction: An attack where an adversary repeatedly queries a target model to create a functionally similar copy without authorization.

- Model Inversion: An attack that attempts to reconstruct training data by analyzing model outputs.

- Model Lifecycle Management – AI Model Lifecycle Management is the process of overseeing

all stages of an AI model's existence, including its design, development, deployment, monitoring, maintenance, and eventual retirement, to ensure it remains effective and aligned with objectives.

- Model Poisoning: Introducing vulnerabilities or backdoors directly into a model during the training process.

- Model Stealing/Theft: Extracting a copy or approximation of a proprietary model through repeated queries.

- Multi-agent System: A system composed of multiple interacting AI agents, each with potentially different capabilities and goals.

- OPA (Open Policy Agent): An open-source policy engine that enables unified policy enforcement across the stack.

- Privacy-Preserving Machine Learning (PPML): Techniques and methods to train and deploy ML models while protecting the privacy of the training data.

- Prompt Injection: An attack where malicious instructions are embedded in inputs to override a model's intended behavior.

- RAG (Retrieval-Augmented Generation): A technique that enhances large language models by retrieving relevant information from external knowledge sources before generating a response.

- Red-Teaming: The practice of actively testing AI systems by simulating adversarial attacks to identify vulnerabilities.

- SBOM (Software Bill of Materials): A formal record containing the details and supply chain relationships of various components used in building software or AI models.

- SHAP (SHapley Additive exPlanations): A game theoretic approach to explain the output of any machine learning model by computing the contribution of each feature to the prediction.

- Strong Authentication: Authentication that resists credential theft and replay by requiring at least two factors (knowledge, possession, inherence) and phishing-resistant mechanisms such as FIDO2/WebAuthn, certificate-based service auth, or short-lived tokens.

- Supply Chain Attack: Compromising a system by targeting less-secure elements in its supply chain, such as third-party libraries, datasets, or pre-trained models.

- Transfer Learning: A technique where a model developed for one task is reused as the starting point for a model on a second task.

- Vector Database: A specialized database designed to store high-dimensional vectors (embeddings) and perform efficient similarity searches.

- Vulnerability Scanning: Automated tools that identify known security vulnerabilities in software components, including AI frameworks and dependencies.

- Watermarking: Techniques to embed imperceptible markers in AI-generated content to track its origin or detect AI generation.

- Zero-Day Vulnerability: A previously unknown vulnerability that attackers can exploit before developers create and deploy a patch.

Appendix B: References

TODO

Appendix C: AI Security Governance & Documentation (Reorg

Objective

This appendix provides foundational requirements for establishing organizational structures, policies, documentation, and processes to govern AI security throughout the system lifecycle.

AC.1 AI Risk Management Framework Adoption

#AC.1.1 Level: 1 Role: D/V
Verify that an AI-specific risk assessment methodology is documented and implemented.

#AC.1.2 Level: 2 Role: D
Verify that risk assessments are conducted at key points in the AI lifecycle and prior to significant changes.

#AC.1.3 Level: 3 Role: D/V
Verify that the risk management framework aligns with established standards (e.g., NIST AI RMF).

AC.2 AI Security Policy & Procedures

#AC.2.1 Level: 1 Role: D/V
Verify that documented AI security policies exist.

#AC.2.2 Level: 2 Role: D
Verify that policies are reviewed and updated at least annually and after significant threat-landscape changes.

#AC.2.3 Level: 3 Role: D/V
Verify that policies address all AISVS categories and applicable regulatory requirements.

AC.3 Roles & Responsibilities for AI Security

#AC.3.1 Level: 1 Role: D/V
Verify that AI security roles and responsibilities are documented.

#AC.3.2 Level: 2 Role: D
Verify that responsible individuals possess appropriate security expertise.

#AC.3.3 Level: 3 Role: D/V
Verify that an AI ethics committee or governance board is established for high-risk AI systems.

AC.4 Ethical AI Guidelines Enforcement

#AC.4.1 Level: 1 Role: D/V
Verify that ethical guidelines for AI development and deployment exist.

#AC.4.2 Level: 2 Role: D
Verify that mechanisms are in place to detect and report ethical violations.

#AC.4.3 Level: 3 Role: D/V
Verify that regular ethical reviews of deployed AI systems are performed.

AC.5 AI Regulatory Compliance Monitoring

#AC.5.1 Level: 1 Role: D/V
Verify that processes exist to identify applicable AI regulations.

#AC.5.2 Level: 2 Role: D
Verify that compliance with all regulatory requirements is assessed.

#AC.5.3 Level: 3 Role: D/V
Verify that regulatory changes trigger timely reviews and updates to AI systems.

AC.6 Training Data Governance, Documentation & Process

AC.6.1 Data Sourcing & Due Diligence

#AC.6.1.1 Level: 1 Role: D/V
Verify that only datasets vetted for quality, representativeness, ethical sourcing, and license compliance are allowed, reducing risks of poisoning, embedded bias, and intellectual property infringement.

#AC.6.1.2 Level: 2 Role: D/V
Verify that third-party data suppliers, including providers of pre-trained models and external datasets, undergo security, privacy, ethical sourcing, and data quality due diligence before their data or models are integrated.

#AC.6.1.3 Level: 1 Role: D
Verify that external transfers use TLS/auth and integrity checks.

#AC.6.1.4 Level: 2 Role: D/V
Verify that high-risk data sources (e.g., open-source datasets with unknown provenance, unvetted suppli-

ers) receive enhanced scrutiny, such as sandboxed analysis, extensive quality/bias checks, and targeted poisoning detection, before use in sensitive applications.

#AC.6.1.5 Level: 3 Role: D/V
Verify that Verify that pre-trained models obtained from third parties are evaluated for embedded biases, potential backdoors, integrity of their architecture, and the provenance of their original training data before fine-tuning or deployment.

AC.6.2 Bias & Fairness Management

#AC.6.2.1 Level: 1 Role: D/V
Verify that datasets are profiled for representational imbalance and potential biases across legally protected attributes (e.g., race, gender, age) and other ethically sensitive characteristics relevant to the model's application domain (e.g., socio-economic status, location).

#AC.6.2.2 Level: 2 Role: D/V
Verify that that identified biases are mitigated via documented strategies such as re-balancing, targeted data augmentation, algorithmic adjustments (e.g., pre-processing, in-processing, post-processing techniques), or re-weighting, and the impact of mitigation on both fairness and overall model performance is assessed.

#AC.6.2.3 Level: 2 Role: D/V
Verify that post-training fairness metrics are evaluated and documented.

#AC.6.2.4 Level: 3 Role: D/V
Verify that a lifecycle bias-management policy assigns owners and review cadence.

AC.6.3 Labeling & Annotation Governance

#AC.6.3.1 Level: 2 Role: D/V
Verify that labelling/annotation quality is ensured via reviewer cross-checks or consensus.

#AC.6.3.2 Level: 2 Role: D/V
Verify that data cards are maintained for significant training datasets, detailing characteristics, motivations, composition, collection processes, preprocessing, licenses, and recommended/discouraged uses.

#AC.6.3.3 Level: 2 Role: D/V
Verify that data cards document bias risks, demographic skews, and ethical considerations relevant to the dataset.

#AC.6.3.4 Level: 2 Role: D/V
Verify that data cards are versioned alongside datasets and updated whenever the dataset is modified.

#AC.6.3.5 Level: 2 Role: D/V
Verify that data cards are reviewed and approved by both technical and non-technical stakeholders (e.g., compliance, ethics, domain experts).

#AC.6.3.6 Level: 2 Role: D/V
Verify that labeling/annotation quality is ensured via clear guidelines, reviewer cross-checks, consensus mechanisms (e.g., monitoring inter-annotator agreement), and defined processes for resolving discrepancies.

#AC.6.3.7 Level: 3 Role: D/V
Verify that labels critical to safety, security, or fairness (e.g., identifying toxic content, critical medical findings) receive mandatory independent dual review or equivalent robust verification.

#AC.6.3.8 Level: 2 Role: D/V
Verify that labeling guides and instructions are comprehensive, version-controlled, and peer-reviewed.
#AC.6.3.9 Level: 2 Role: D/V
Verify that data schemas for labels are clearly defined, and version-controlled.
#AC.6.3.10 Level: 2 Role: D/V
Verify that outsourced or crowdsourced labeling workflows include technical/procedural safeguards to ensure data confidentiality, integrity, label quality, and prevent data leakage.
#AC.6.3.11 Level: 2 Role: D/V
Verify that all personnel involved in data annotation are background-checked and trained in data security and privacy.
#AC.6.3.12 Level: 2 Role: D/V
Verify that all annotation personnel sign confidentiality and non-disclosure agreements.
#AC.6.3.13 Level: 2 Role: D/V
Verify that annotation platforms enforce access controls and monitor for insider threats.

AC.6.4 Dataset Quality Gates & Quarantine

#AC.6.4.1 Level: 2 Role: D/V
Verify that failed datasets are quarantined with audit trails.
#AC.6.4.2 Level: 2 Role: D/V
Verify that quality gates block sub-par datasets unless exceptions are approved.
#AC.6.4.3 Level: 2 Role: V
Verify that manual spot-checks by domain experts cover a statistically significant sample (e.g., ≥1% or 1,000 samples, whichever is greater, or as determined by risk assessment) to identify subtle quality issues not caught by automation.

AC.6.5 Threat/Poisoning Detection & Drift

#AC.6.5.1 Level: 2 Role: D/V
Verify that flagged samples trigger manual review before training.
#AC.6.5.2 Level: 2 Role: V
Verify that results feed the model's security dossier and inform ongoing threat intelligence.
#AC.6.5.3 Level: 3 Role: D/V
Verify that detection logic is refreshed with new threat intel.
#AC.6.5.4 Level: 3 Role: D/V
Verify that online-learning pipelines monitor distribution drift.

AC.6.6 Deletion, Consent, Rights, Retention & Compliance

#AC.6.6.1 Level: 1 Role: D/V
Verify that training data deletion workflows purge primary and derived data and assess model impact, and that the impact on affected models is assessed and, if necessary, addressed (e.g., through retraining or recalibration).
#AC.6.6.2 Level: 2 Role: D
Verify that mechanisms are in place to track and respect the scope and status of user consent (and with-

drawals) for data used in training, and that consent is validated before data is incorporated into new training processes or significant model updates.

#AC.6.6.3 Level: 2 Role: V
Verify that workflows are tested annually and logged.

#AC.6.6.4 Level: 1 Role: D/V
Verify that explicit retention periods are defined for all training datasets.

#AC.6.6.5 Level: 2 Role: D/V
Verify that datasets are automatically expired, deleted, or reviewed for deletion at the end of their lifecycle.

#AC.6.6.6 Level: 2 Role: D/V
Verify that retention and deletion actions are logged and auditable.

#AC.6.6.7 Level: 2 Role: D/V
Verify that data residency and cross-border transfer requirements are identified and enforced for all datasets.

#AC.6.6.8 Level: 2 Role: D/V
Verify that sector-specific regulations (e.g., healthcare, finance) are identified and addressed in data handling.

#AC.6.6.9 Level: 2 Role: D/V
Verify that compliance with relevant privacy laws (e.g., GDPR, CCPA) is documented and reviewed regularly.

#AC.6.6.10 Level: 2 Role: D/V
Verify that mechanisms exist to respond to data subject requests for access, rectification, restriction, or objection.

#AC.6.6.11 Level: 2 Role: D/V
Verify that requests are logged, tracked, and fulfilled within legally mandated timeframes.

#AC.6.6.12 Level: 2 Role: D/V
Verify that data subject rights processes are tested and reviewed regularly for effectiveness.

AC.6.7 Versioning & Change Management

#AC.6.7.1 Level: 2 Role: D/V
Verify that an impact analysis is performed before updating or replacing a dataset version, covering model performance, fairness, and compliance.

#AC.6.7.2 Level: 2 Role: D/V
Verify that results of the impact analysis are documented and reviewed by relevant stakeholders.

#AC.6.7.3 Level: 2 Role: D/V
Verify that rollback plans exist in case new versions introduce unacceptable risks or regressions.

AC.6.8 Synthetic Data Governance

#AC.6.8.1 Level: 2 Role: D/V
Verify that the generation process, parameters, and intended use of synthetic data are documented.

#AC.6.8.2 Level: 2 Role: D/V
Verify that synthetic data is risk-assessed for bias, privacy leakage, and representational issues before use in training.

AC.6.9 Access Monitoring

#AC.6.9.1 Level: 2 Role: D/V
Verify that access logs are regularly reviewed for unusual patterns, such as large exports or access from new locations.
#AC.6.9.2 Level: 2 Role: D/V
Verify that alerts are generated for suspicious access events and investigated promptly.

AC.6.10 Adversarial Training Governance

#AC.6.10.1 Level: 2 Role: D/V
Verify that if adversarial training is used, the generation, management, and versioning of adversarial datasets are documented and controlled.
#AC.6.10.2 Level: 3 Role: D/V
Verify that the impact of adversarial robustness training on model performance (against both clean and adversarial inputs) and fairness metrics is evaluated, documented, and monitored.
#AC.6.10.3 Level: 3 Role: D/V
Verify that strategies for adversarial training and robustness are periodically reviewed and updated to counter evolving adversarial attack techniques.

AC.7 Model Lifecycle Governance & Documentation

#AC.7.1 Level: 2 Role: D/V
Verify that all model artifacts use semantic versioning (MAJOR.MINOR.PATCH) with documented criteria specifying when each version component increments.
#AC.7.2 Level: 2 Role: D/V
Verify that emergency deployments require documented security risk assessment and approval from a pre-designated security authority within pre-agreed timeframes.
#AC.7.3 Level: 2 Role: V
Verify that rollback artifacts (previous model versions, configurations, dependencies) are retained according to organizational policies.
#AC.7.4 Level: 2 Role: D/V
Verify that audit log access requires appropriate authorization and all access attempts are logged with user identity and a timestamp.
#AC.7.5 Level: 1 Role: D/V
Verify that retired model artifacts are retained according to data retention policies.

AC.8 Prompt, Input, and Output Safety Governance

AC.8.1 Prompt Injection Defense

#AC.8.1.1 Level: 2 Role: D/V

Verify that adversarial evaluation tests (e.g., Red Team "many-shot" prompts) are run before every model or prompt-template release, with success-rate thresholds and automated blockers for regressions.

#AC.8.1.2 Level: 3 Role: D/V
Verify that all prompt-filter rule updates, classifier model versions and block-list changes are version-controlled and auditable.

AC.8.2 Adversarial-Example Resistance

#AC.8.2.1 Level: 3 Role: D/V
Verify that robustness metrics (success rate of known attack suites) are tracked over time via automation and regressions trigger an alert.

AC.8.3 Content & Policy Screening

#AC.8.3.1 Level: 2 Role: D
Verify that the screening model or rule set is retrained/updated at least quarterly, incorporating newly observed jailbreak or policy bypass patterns.

AC.8.4 Input Rate Limiting & Abuse Prevention

#AC.8.4.1 Level: 3 Role: V
Verify that abuse prevention logs are retained and reviewed for emerging attack patterns.

AC.8.5 Input Provenance & Attribution

#AC.8.5.1 Level: 1 Role: D/V
Verify that all user inputs are tagged with metadata (user ID, session, source, timestamp, IP address) at ingestion.

#AC.8.5.2 Level: 2 Role: D/V
Verify that provenance metadata is retained and auditable for all processed inputs.

#AC.8.5.3 Level: 2 Role: D/V
Verify that anomalous or untrusted input sources are flagged and subject to enhanced scrutiny or blocking.

AC.9 Multimodal Validation, MLOps & Infrastructure Governance

AC.9.1 Multimodal Security Validation Pipeline

#AC.9.1.1 Level: 3 Role: D/V
Verify that modality-specific content classifiers are updated according to documented schedules (minimum quarterly) with new threat patterns, adversarial examples, and performance benchmarks maintained above baseline thresholds.

AC.9.2 CI/CD & Build Security

#AC.9.2.1 Level: 1 Role: D/V
Verify that infrastructure-as-code is scanned on every commit, and merges are blocked on critical or high-severity findings.

#AC.9.2.2 Level: 2 Role: D/V
Verify that CI/CD pipelines use short-lived, scoped identities for access to secrets and infrastructure.

#AC.9.2.3 Level: 2 Role: D/V
Verify that build environments are isolated from production networks and data.

AC.9.3 Container & Image Security

#AC.9.3.1 Level: 2 Role: D/V
Verify that container images are scanned to block hardcoded secrets (e.g., API keys, credentials, certificates).

#AC.9.3.2 Level: 1 Role: D/V
Verify that container images are scanned according to organizational schedules with CRITICAL vulnerabilities blocking deployment based on organizational risk thresholds.

AC.9.4 Monitoring, Alerting & SIEM

#AC.9.4.1 Level: 2 Role: V
Verify that security alerts integrate with SIEM platforms (Splunk, Elastic, or Sentinel) using CEF or STIX/-TAXII formats with automated enrichment.

AC.9.5 Vulnerability Management

#AC.9.5.1 Level: 2 Role: D/V
Verify that HIGH severity vulnerabilities are patched according to organizational risk management timelines with emergency procedures for actively exploited CVEs.

AC.9.6 Configuration & Drift Control

#AC.9.6.1 Level: 2 Role: D/V
Verify that configuration drift is detected using tools (Chef InSpec, AWS Config) according to organizational monitoring requirements with automatic rollback for unauthorized changes.

AC.9.7 Production Environment Hardening

#AC.9.7.1 Level: 2 Role: D/V
Verify that production environments block SSH access, disable debug endpoints, and require change requests with organizational advance notice requirements except emergencies.

AC.9.8 Release Promotion Gates

#AC.9.8.1 Level: 2 Role: D/V
Verify that promotion gates include automated security tests (SAST, DAST, container scanning) with zero CRITICAL findings required for approval.

AC.9.9 Workload, Capacity & Cost Monitoring

#AC.9.9.1 Level: 1 Role: D/V
Verify that GPU/TPU utilization is monitored with alerts triggered at organizationally defined thresholds and automatic scaling or load balancing activated based on capacity management policies.

#AC.9.9.2 Level: 1 Role: D/V
Verify that AI workload metrics (inference latency, throughput, error rates) are collected according to organizational monitoring requirements and correlated with infrastructure utilization.

#AC.9.9.3 Level: 2 Role: V
Verify that cost monitoring tracks spending per workload/tenant with alerts based on organizational budget thresholds and automated controls for budget overruns.

#AC.9.9.4 Level: 3 Role: V
Verify that capacity planning uses historical data with organizationally defined forecasting periods and automated resource provisioning based on demand patterns.

AC.9.10 Approvals & Audit Trails

#AC.9.10.1 Level: 1 Role: D/V
Verify that environment promotion requires approval from organizationally defined authorized personnel with cryptographic signatures and immutable audit trails.

AC.9.11 IaC Governance

#AC.9.11.1 Level: 2 Role: D/V
Verify that infrastructure-as-code changes require peer review with automated testing and security scanning before merge to main branch.

AC.9.12 Data Handling in Non-Production

#AC.9.12.1 Level: 2 Role: D/V
Verify that non-production data is anonymized according to organizational privacy requirements, synthetic data generation, or complete data masking with PII removal verified.

AC.9.13 Backup & Disaster Recovery

#AC.9.13.1 Level: 1 Role: D/V
Verify that infrastructure configurations are backed up according to organizational backup schedules to geographically separate regions with 3-2-1 backup strategy implementation.

#AC.9.13.2 Level: 2 Role: V
Verify that recovery procedures are tested and validated through automated testing according to organizational schedules with RTO and RPO targets meeting organizational requirements.

#AC.9.13.3 Level: 3 Role: V
Verify that disaster recovery includes AI-specific runbooks with model weight restoration, GPU cluster rebuilding, and service dependency mapping.

AC.9.14 Compliance & Documentation

#AC.9.14.1 Level: 2 Role: D/V
Verify that infrastructure compliance is assessed according to organizational schedules against SOC 2, ISO 27001, or FedRAMP controls with automated evidence collection.

#AC.9.14.2 Level: 2 Role: V
Verify that infrastructure documentation includes network diagrams, data flow maps, and threat models updated according to organizational change management requirements.

#AC.9.14.3 Level: 3 Role: D/V
Verify that infrastructure changes undergo automated compliance impact assessment with regulatory approval workflows for high-risk modifications.

AC.9.15 Hardware & Supply Chain

#AC.9.15.1 Level: 2 Role: D/V
Verify that AI accelerator firmware (GPU BIOS, TPU firmware) is verified with cryptographic signatures and updated according to organizational patch management timelines.

#AC.9.15.2 Level: 3 Role: V
Verify that the AI hardware supply chain includes provenance verification with manufacturer certificates and tamper-evident packaging validation.

AC.9.16 Cloud Strategy & Portability

#AC.9.16.1 Level: 3 Role: V
Verify that cloud vendor lock-in prevention includes portable infrastructure-as-code, standardized APIs, and data export capabilities with format conversion tools.

#AC.9.16.2 Level: 3 Role: V
Verify that multi-cloud cost optimization includes security controls preventing resource sprawl as well as unauthorized cross-cloud data transfer charges.

AC.9.17 GitOps & Self-Healing

#AC.9.17.1 Level: 2 Role: D/V
Verify that GitOps repositories require signed commits with GPG keys and branch protection rules preventing direct pushes to main branches.

#AC.9.17.2 Level: 3 Role: V
Verify that self-healing infrastructure includes security event correlation with automated incident response and stakeholder notification workflows.

AC.9.18 Zero-Trust, Agents, Provisioning & Residency Attestation

#AC.9.18.1 Level: 2 Role: D/V
Verify that cloud resource access includes zero-trust verification with continuous authentication.
#AC.9.18.2 Level: 2 Role: D/V
Verify that automated infrastructure provisioning includes security policy validation with deployment blocking for non-compliant configurations.
#AC.9.18.3 Level: 2 Role: D/V
Verify that automated infrastructure provisioning validates security policies during CI/CD, with non-compliant configurations blocked from deployment.
#AC.9.18.4 Level: 3 Role: D/V
Verify that data residency requirements are enforced by cryptographic attestation of storage locations.
#AC.9.18.5 Level: 3 Role: D/V
Verify that cloud provider security assessments include agent-specific threat modeling and risk evaluation.

AC.9.19 Access Control & Identity

#5.1.3 Level: 2 Role: D
Verify that new principals undergo identity-proofing that is aligned with NIST 800-63-3 IAL-2 or equivalent standards before receiving production system access.
#5.1.4 Level: 2 Role: V
Verify that access reviews are conducted quarterly with automated detection of dormant accounts, credential rotation enforcement, and de-provisioning workflows.
#5.2.2 Level: 1 Role: D/V
Verify that least-privilege principles are enforced by default with service accounts starting at read-only permissions and documented business justification required for write access.
#5.3.3 Level: 2 Role: D
Verify that policy definitions are version-controlled, peer-reviewed, and validated through automated testing in CI/CD pipelines before production deployment.
#5.3.4 Level: 2 Role: V
Verify that policy evaluation results include decision rationales and are transmitted to SIEM systems for correlation analysis and compliance reporting.
#5.4.4 Level: 2 Role: V
Verify that policy evaluation latency is continuously monitored with automated alerts for timeout conditions that could enable authorization bypass.
#5.5.4 Level: 2 Role: V
Verify that redaction algorithms are deterministic, version-controlled, and maintain audit logs to support compliance investigations and forensic analysis.
#5.5.5 Level: 3 Role: V
Verify that high-risk redaction events generate adaptive logs that include cryptographic hashes of original content for forensic retrieval without data exposure.
#5.7.5 Level: 3 Role: V
Verify that agent error conditions and exception handling include capability scope information to support incident analysis and forensic investigation.
#5.4.2 Level: 1 Role: D/V
Verify that citations, references, and source attributions in model outputs are validated against caller entitlements and removed if unauthorized access is detected.

New Items to be Integrated Above

#2.3.3 Level: 2 Role: D/V
Verify that the allowed character set is regularly reviewed and updated to ensure it remains aligned with business requirements.

#7.2.4 Level: 3 Role: D/V
Verify that thresholds and detectors are re-calibrated after major model or knowledge-base updates.

#7.2.5 Level: 3 Role: V
Verify that dashboard visualizations track hallucination rates.

#7.5.4 Level: 3 Role: V
Verify that explainability artifacts are version-controlled alongside model releases for auditability.

#7.6.5 Level: 3 Role: V
Verify that monitoring pipelines are penetration-tested and access-controlled to avoid leakage of sensitive logs.

#7.6.4 Level: 2 Role: D/V
Verify that monitoring data feeds back into retraining, fine-tuning, or rule updates within a documented MLOps workflow.

Appendix D: AI-Assisted Secure Coding Governance & Verifi

Objective

This chapter defines baseline organizational controls for the safe and effective use of AI-assisted coding tools during software development, ensuring security and traceability across the SDLC.

AD.1 AI-Assisted Secure-Coding Workflow

Integrate AI tooling into the organization's secure-software-development lifecycle (SSDLC) without weakening existing security gates.

#AD.1.1 Level: 1 Role: D/V
Verify that a documented workflow describes when and how AI tools may generate, refactor, or review code.

#AD.1.2 Level: 2 Role: D
Verify that the workflow maps to each SSDLC phase (design, implementation, code review, testing, deployment).

#AD.1.3 Level: 3 Role: D/V
Verify that metrics (e.g., vulnerability density, mean-time-to-detect) are collected on AI-produced code and compared to human-only baselines.

AD.2 AI Tool Qualification & Threat Modeling

Ensure AI coding tools are evaluated for security capabilities, risk, and supply-chain impact before adoption.

#AD.2.1 Level: 1 Role: D/V
Verify that a threat model for each AI tool identifies misuse, model-inversion, data leakage, and dependency-chain risks.

#AD.2.2 Level: 2 Role: D
Verify that tool evaluations include static/dynamic analysis of any local components and assessment of

SaaS endpoints (TLS, authentication/authorization, logging).

#AD.2.3 Level: 3 Role: D/V
Verify that evaluations follow a recognized framework and are re-performed after major version changes.

AD.3 Secure Prompt & Context Management

Prevent leakage of secrets, proprietary code, and personal data when constructing prompts or contexts for AI models.

#AD.3.1 Level: 1 Role: D/V
Verify that written guidance prohibits sending secrets, credentials, or classified data in prompts.

#AD.3.2 Level: 2 Role: D
Verify that technical controls (client-side redaction, approved context filters) automatically strip sensitive artifacts.

#AD.3.3 Level: 3 Role: D/V
Verify that prompts and responses are tokenized, encrypted in transit and at rest, and retention periods comply with data-classification policy.

AD.4 Validation of AI-Generated Code

Detect and remediate vulnerabilities introduced by AI output before the code is merged or deployed.

#AD.4.1 Level: 1 Role: D/V
Verify that AI-generated code is always subjected to human code review.

#AD.4.2 Level: 2 Role: D
Verify that automated scanners (SAST/IAST/DAST) run on every pull request containing AI-generated code and block merges on critical findings.

#AD.4.3 Level: 3 Role: D/V
Verify that differential fuzz testing or property-based tests prove security-critical behaviors (e.g., input validation, authorization logic).

AD.5 Explainability & Traceability of Code Suggestions

Provide auditors and developers with insight into why a suggestion was made and how it evolved.

#AD.5.1 Level: 1 Role: D/V

Verify that prompt/response pairs are logged with commit IDs.

#AD.5.2 Level: 2 Role: D
Verify that developers can surface model citations (training snippets, documentation) supporting a suggestion.

#AD.5.3 Level: 3 Role: D/V
Verify that explainability reports are stored with design artifacts and referenced in security reviews, satisfying ISO/IEC 42001 traceability principles.

AD.6 Continuous Feedback & Model Fine-Tuning

Improve model security performance over time while preventing negative drift.

#AD.6.1 Level: 1 Role: D/V
Verify that developers can flag insecure or non-compliant suggestions, and that flags are tracked.

#AD.6.2 Level: 2 Role: D
Verify that aggregated feedback informs periodic fine-tuning or retrieval-augmented generation with vetted secure-coding corpora (e.g., OWASP Cheat Sheets).

#AD.6.3 Level: 3 Role: D/V
Verify that a closed-loop evaluation harness runs regression tests after every fine-tune; security metrics must meet or exceed prior baselines before deployment.

References

- NIST AI Risk Management Framework 1.0
- ISO/IEC 42001:2023 — AI Management Systems Requirements
- OWASP Secure Coding Practices — Quick Reference Guide

Appendix E: Example Tools and Frameworks

Objective

This chapter provides examples for tooling and frameworks which can support the implementation or fulfillment of a given AISVS requirement. These are not to be viewed as recommendations or endorsements by the AISVS team or the OWASP GenAI Security Project.

AE.1 Training Data Governance & Bias Management

Tooling used for data analytics, governance, and bias management.

#AE.1.1 Section: 1.1
Data Inventory Tooling: Data inventory management tooling like...
#AE.1.2 Section: 1.2
Encryption-In-Transit Use TLS for HTTPS-based applications, with tools like openSSL and python's `ssl` library.

AE.2 User Input Validation

Tooling to handle and validated user inputs.

#AE.2.1 Section: 2.1
Prompt Injection Defense Tooling: Use guardrail tooling like NVIDIA's NeMo or Guardrails AI.

Appendix B: Strategic Controls

C4.15 Quantum-Resistant Infrastructure Security

Prepare AI infrastructure for quantum computing threats through post-quantum cryptography and quantum-safe protocols.

#4.15.1 Level: 3 Role: D/V
Verify that AI infrastructure implements NIST-approved post-quantum cryptographic algorithms (CRYSTALS-Kyber, CRYSTALS-Dilithium, SPHINCS+) for key exchange and digital signatures.
#4.15.2 Level: 3 Role: D/V
Verify that quantum key distribution (QKD) systems are implemented for high-security AI communications with quantum-safe key management protocols.
#4.15.3 Level: 3 Role: D/V
Verify that cryptographic agility frameworks enable rapid migration to new post-quantum algorithms with automated certificate and key rotation.
#4.15.4 Level: 3 Role: V
Verify that quantum threat modeling assesses AI infrastructure vulnerability to quantum attacks with documented migration timelines and risk assessments.
#4.15.5 Level: 3 Role: D/V
Verify that hybrid classical-quantum cryptographic systems provide defense-in-depth during the quantum transition period with performance monitoring.

C4.17 Zero-Knowledge Infrastructure

Implement zero-knowledge proof systems for privacy-preserving AI verification and authentication without revealing sensitive information.

#4.17.1 Level: 3 Role: D/V
Verify that zero-knowledge proofs (ZK-SNARKs) verify AI model integrity and training origin without exposing model weights or training data.
#4.17.2 Level: 3 Role: D/V
Verify that ZK-based authentication systems enable privacy-preserving user verification for AI services without revealing identity-related information.
#4.17.3 Level: 3 Role: D/V

Verify that private set intersection (PSI) protocols enable secure data matching for federated AI without exposing individual datasets.

#4.17.4 Level: 3 Role: D/V

Verify that zero-knowledge machine learning (ZKML) systems enable verifiable AI inferences with cryptographic proof of correct computation.

#4.17.5 Level: 3 Role: D/V

Verify that ZK-rollups provide scalable, privacy-preserving AI transaction processing with batch verification and reduced computational overhead.

C4.18 Side-Channel Attack Prevention

Protect AI infrastructure from timing, power, electromagnetic, and cache-based side-channel attacks that could leak sensitive information.

#4.18.1 Level: 3 Role: D/V

Verify that AI inference timing is normalized using constant-time algorithms and padding to prevent timing-based model extraction attacks.

#4.18.2 Level: 3 Role: D/V

Verify that power analysis protection includes noise injection, power line filtering, and randomized execution patterns for AI hardware.

#4.18.3 Level: 3 Role: D/V

Verify that cache-based side-channel mitigation uses cache partitioning, randomization, and flush instructions to prevent information leakage.

#4.18.4 Level: 3 Role: D/V

Verify that electromagnetic emanation protection includes shielding, signal filtering, and randomized processing to prevent TEMPEST-style attacks.

#4.18.5 Level: 3 Role: D/V

Verify that microarchitectural side-channel defenses include speculative execution controls and memory access pattern obfuscation.

C4.19 Neuromorphic & Specialized AI Hardware Security

Secure emerging AI hardware architectures including neuromorphic chips, FPGAs, custom ASICs, and optical computing systems.

#4.19.1 Level: 3 Role: D/V

Verify that neuromorphic chip security includes spike pattern encryption, synaptic weight protection, and hardware-based learning rule validation.

#4.19.2 Level: 3 Role: D/V

Verify that FPGA-based AI accelerators implement bitstream encryption, anti-tamper mechanisms, and se-

cure configuration loading with authenticated updates.

#4.19.3 Level: 3 Role: D/V

Verify that custom ASIC security includes on-chip security processors, hardware root of trust, and secure key storage with tamper detection.

#4.19.4 Level: 3 Role: D/V

Verify that optical computing systems implement quantum-safe optical encryption, secure photonic switching, and protected optical signal processing.

#4.19.5 Level: 3 Role: D/V

Verify that hybrid analog-digital AI chips include secure analog computation, protected weight storage, and authenticated analog-to-digital conversion.

C4.20 Privacy-Preserving Compute Infrastructure

Implement infrastructure controls for privacy-preserving computation to protect sensitive data during AI processing and analysis.

#4.20.1 Level: 3 Role: D/V

Verify that homomorphic encryption infrastructure enables encrypted computation on sensitive AI workloads with cryptographic integrity verification and performance monitoring.

#4.20.2 Level: 3 Role: D/V

Verify that private information retrieval systems enable database queries without revealing query patterns with cryptographic protection of access patterns.

#4.20.3 Level: 3 Role: D/V

Verify that secure multi-party computation protocols enable privacy-preserving AI inference without exposing individual inputs or intermediate computations.

#4.20.4 Level: 3 Role: D/V

Verify that privacy-preserving key management includes distributed key generation, threshold cryptography, and secure key rotation with hardware-backed protection.

#4.20.5 Level: 3 Role: D/V

Verify that privacy-preserving compute performance is optimized through batching, caching, and hardware acceleration while maintaining cryptographic security guarantees.

4.9.14.9.2 1: 2 D/V: D/V

Verify that multi-cloud deployments use federated identity standards (e.g., OIDC, SAML) with centralized policy enforcement across providers.

4.9.14.9.3 1: 2 D/V: D/V

Verify that cross-cloud and hybrid data transfers use end-to-end encryption with customer-managed keys and enforce jurisdictional data residency requirements.

4.9.14.9.1 1: 1 D/V: D/V

Verify that cloud storage integration uses end-to-end encryption with agent-controlled key management.

4.9.14.9.2 1: 2 D/V: D/V

Verify that hybrid deployment security boundaries are clearly defined with encrypted communication channels.