Artificial Intelligence Security Verification Standard

Initial Version Work In Progress

May 29, 2025

Commit 7dd43f9

v4.6.0a 20250526-154750

Table of Contents

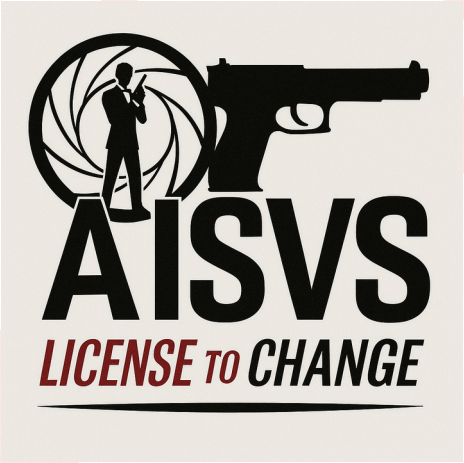
Frontispiece

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

Version 0.1 (First Public Draft - Work In Progress), 2025



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 throughout the 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 only datasets vetted for quality, representativeness, ethical sourcing, and licence compliance are allowed, reducing risks of poisoning, embedded bias, and intellectual property infringement.

#1.1.3 Level: 1 Role: D/V
Verify that data minimisation is enforced so unnecessary or sensitive attributes are excluded.

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

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

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

C1.2 Training Data Security & Integrity

Restrict access, encrypt at rest and in transit, and validate integrity to block tampering or theft.)

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

#1.2.2 Level: 2 Role: D/V
Verify that datasets are encrypted in transit and, for all sensitive or personally identifiable information (PII),

at rest, using industry-standard cryptographic algorithms and key management practices.

#1.2.3 Level: 2 Role: D/V
Verify that cryptographic hashes or digital signatures are used to ensure data integrity during storage and transfer, and that automated anomaly detection techniques are applied to guard against unauthorized modifications or corruption, including targeted data poisoning attempts.

#1.2.4 Level: 3 Role: D/V
Verify that dataset versions are tracked to enable rollback.

#1.2.5 Level: 2 Role: D/V
Verify that obsolete data is securely purged or anonymized in compliance with data retention policies, regulatory requirements, and to shrink the attack surface.

C1.3 Representation Completeness & Fairness

Detect demographic skews and apply mitigation so the model's error rates are equitable across groups.

#1.3.1 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).

#1.3.2 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.

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

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

C1.4 Training Data Labeling Quality, Integrity, and Security

Cryptographically protect labels and require dual review for critical classes.

#1.4.1 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.

#1.4.2 Level: 2 Role: D/V
Verify that cryptographic hashes or digital signatures are applied to label artefacts to ensure their integrity

and authenticity.

#1.4.3 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.

#1.4.4 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.

#1.4.5 Level: 2 Role: D/V
Verify that sensitive information (including PII) inadvertently captured or necessarily present in labels is redacted, pseudonymized, anonymized, or encrypted at rest and in transit, according to data minimization principles.

#1.4.6 Level: 2 Role: D/V
Verify that labeling guides and instructions are comprehensive, version-controlled, and peer-reviewed.

#1.4.7 Level: 2 Role: D/V
Verify that data schemas (including for labels) are clearly defined, and version-controlled.

#1.8.2 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.

C1.5 Training Data Quality Assurance

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

#1.5.1 Level: 1 Role: D
Verify that automated tests catch format errors, nulls, and label skews on every ingest or significant transformation.

#1.5.2 Level: 1 Role: D/V
Verify that failed datasets are quarantined with audit trails.

#1.5.3 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.

#1.5.4 Level: 2 Role: D/V
Verify that remediation steps are appended to provenance records.

#1.5.5 Level: 2 Role: D/V
Verify that quality gates block sub-par datasets unless exceptions are approved.

C1.6 Data Poisoning Detection

Apply statistical anomaly detection and quarantine workflows to stop adversarial insertions.

#1.6.1 Level: 2 Role: D/V
Verify that anomaly detection techniques (e.g., statistical methods, outlier detection, embedding analysis) are applied to training data at ingest and before major training runs to identify potential poisoning attacks or unintentional data corruption.

#1.6.2 Level: 2 Role: D/V
Verify that flagged samples trigger manual review before training.

#1.6.3 Level: 2 Role: V
Verify that results feed the model's security dossier and inform ongoing threat intelligence.

#1.6.4 Level: 3 Role: D/V
Verify that detection logic is refreshed with new threat intel.

#1.6.5 Level: 3 Role: D/V
Verify that online-learning pipelines monitor distribution drift.

#1.6.6 Level: 3 Role: D/V
Verify that specific defenses against known data poisoning attack types (e.g., label flipping, backdoor trigger insertion, influential instance attacks) are considered and implemented based on the system's risk profile and data sources.

C1.7 User Data Deletion & Consent Enforcement

Honor deletion and consent-withdrawal requests across datasets, backups, and derived artefacts.

#1.7.1 Level: 1 Role: D/V
Verify that 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).

#1.7.2 Level: 2 Role: D
Verify that mechanisms are in place to track and respect the scope and status of user consent (and withdrawals) for data used in training, and that consent is validated before data is incorporated into new training processes or significant model updates.

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

C1.8 Supply Chain Security

Vet external data providers and verify integrity over authenticated, encrypted channels.

#1.8.1 Level: 2 Role: D/V
Verify that third-party data suppliers, including providers of pre-trained models and external datasets, un-

dergo security, privacy, ethical sourcing, and data quality due diligence before their data or models are integrated.

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

#1.8.3 Level: 2 Role: D/V
Verify that high-risk data sources (e.g., open-source datasets with unknown provenance, unvetted suppliers) receive enhanced scrutiny, such as sandboxed analysis, extensive quality/bias checks, and targeted poisoning detection, before use in sensitive applications.

#1.8.4 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.

C1.9 Adversarial Sample Detection

Implement measures during the training phase, such as adversarial training, to enhance model resilience against adversarial examples.t.

#1.9.1 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.

#1.9.2 Level: 2 Role: D/V
Verify that if adversarial training is used, the generation, management, and versioning of adversarial datasets are documented and controlled.

#1.9.3 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.

#1.9.4 Level: 3 Role: D/V
Verify that strategies for adversarial training and robustness are periodically reviewed and updated to counter evolving adversarial attack techniques.

C1.10 Data Lineage and Traceability

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

#1.10.1 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.

#1.10.2 Level: 2 Role: D/V
Verify that lineage records are immutable, securely stored, and accessible for audits or investigations.

#1.10.3 Level: 2 Role: D/V
Verify that lineage tracking covers both raw and synthetic data.

C1.11 Synthetic Data Management

Ensure synthetic data is properly managed, labeled, and risk-assessed.

#1.11.1 Level: 2 Role: D/V
Verify that all synthetic data is clearly labeled and distinguishable from real data throughout the pipeline.

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

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

C1.12 Data Access Monitoring & Anomaly Detection

Monitor and alert on unusual access to training data to detect insider threats or exfiltration.

#1.12.1 Level: 2 Role: D/V
Verify that all access to training data is logged, including user, time, and action.

#1.12.2 Level: 2 Role: D/V
Verify that access logs are regularly reviewed for unusual patterns, such as large exports or access from new locations.

#1.12.3 Level: 2 Role: D/V
Verify that alerts are generated for suspicious access events and investigated promptly.

C1.13 Data Retention & Expiry Policies

Define and enforce data retention periods to minimize unnecessary data storage.

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

#1.13.2 Level: 2 Role: D/V

Verify that datasets are automatically expired, deleted, or reviewed for deletion at the end of their lifecycle.
#1.13.3 Level: 2 Role: D/V
Verify that retention and deletion actions are logged and auditable.

C1.14 Regulatory & Jurisdictional Compliance

Ensure all training data complies with applicable laws and regulations.

#1.14.1 Level: 2 Role: D/V
Verify that data residency and cross-border transfer requirements are identified and enforced for all datasets.
#1.14.2 Level: 2 Role: D/V
Verify that sector-specific regulations (e.g., healthcare, finance) are identified and addressed in data handling.
#1.14.3 Level: 2 Role: D/V
Verify that compliance with relevant privacy laws (e.g., GDPR, CCPA) is documented and reviewed regularly.

C1.15 Data Watermarking & Fingerprinting

Detect unauthorized reuse or leakage of proprietary or sensitive data.

#1.15.1 Level: 3 Role: D/V
Verify that datasets or subsets are watermarked or fingerprinted where feasible.
#1.15.2 Level: 3 Role: D/V
Verify that watermarking/fingerprinting methods do not introduce bias or privacy risks.
#1.15.3 Level: 3 Role: D/V
Verify that periodic checks are performed to detect unauthorized reuse or leakage of watermarked data.

C1.16 Data Subject Rights Management

Support data subject rights such as access, rectification, restriction, and objection.

#1.16.1 Level: 2 Role: D/V
Verify that mechanisms exist to respond to data subject requests for access, rectification, restriction, or objection.
#1.16.2 Level: 2 Role: D/V
Verify that requests are logged, tracked, and fulfilled within legally mandated timeframes.

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

C1.17 Dataset Version Impact Analysis

Assess the impact of dataset changes before updating or replacing versions.

#1.17.1 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.
#1.17.2 Level: 2 Role: D/V
Verify that results of the impact analysis are documented and reviewed by relevant stakeholders.
#1.17.3 Level: 2 Role: D/V
Verify that rollback plans exist in case new versions introduce unacceptable risks or regressions.

C1.18 Data Annotation Workforce Security

Ensure the security and integrity of personnel involved in data annotation.

#1.18.1 Level: 2 Role: D/V
Verify that all personnel involved in data annotation are background-checked and trained in data security and privacy.
#1.18.2 Level: 2 Role: D/V
Verify that all annotation personnel sign confidentiality and non-disclosure agreements.
#1.18.3 Level: 2 Role: D/V
Verify that annotation platforms enforce access controls and monitor for insider threats.

References

- NIST AI Risk Management Framework
- EU AI Act – Article 10: Data & Data Governance
- MITRE ATLAS: Adversary Tactics for AI
- Survey of ML Bias Mitigation Techniques – MDPI
- Data Provenance & Lineage Best Practices – Nightfall AI
- Data Labeling Quality Standards – LabelYourData
- Training Data Poisoning Guide – Lakera.ai
- CISA Advisory: Securing Data for AI Systems

- ISO/IEC 23053: AI Management Systems Framework
- IBM: What is AI Governance?
- Google AI Principles
- GDPR & AI Training Data – DataProtectionReport
- Supply-Chain Security for AI Data – AppSOC
- OpenAI Privacy Center – Data Deletion Controls
- Adversarial ML Dataset – Kaggle

C2 User Input Validation

Control Objective

Robust user-input validation is a first-line defense against many of the most damaging attacks on AI systems. Prompt-injection "jailbreaks" can override system instructions, leak sensitive data, or steer the model toward disallowed behavior. Research shows that multi-shot jailbreaks exploiting very long context windows remain effective unless dedicated filters and instruction-hierarchies are in place. Meanwhile, imperceptible adversarial perturbations—such as homoglyph swaps or leetspeak—can silently change a model's decisions.

C2.1 Prompt-Injection Defense

Prompt-injection is one of the top risks for AI systems. Defenses combine static pattern filters, dynamic classifiers and instruction-hierarchy enforcement.

#2.1.1 Level: 1 Role: D/V
Verify that user inputs are screened against a continuously-updated library of known prompt-injection patterns (jailbreak keywords, "ignore previous", role-play chains, indirect HTML/URL attacks).

#2.1.2 Level: 1 Role: D/V
Verify that the system enforces an instruction hierarchy in which system or developer messages override user instructions, even after context window expansion.

#2.1.3 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.

#2.1.4 Level: 2 Role: D
Verify that prompts originating from third-party content (web pages, PDFs, e-mails) are sanitized in an isolated parsing context before being concatenated into the main prompt.

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

C2.2 Adversarial-Example Resistance

Natural Language Processing (NLP) models remain vulnerable to subtle character or word-level perturbations that humans miss but models misclassify.

#2.2.1 Level: 1 Role: D
Verify that basic input-normalization steps (Unicode NFC, homoglyph mapping, whitespace trimming) run before tokenization.

#2.2.2 Level: 2 Role: D/V
Verify that statistical anomaly detection flags inputs with unusually high edit distance to language norms, excessive repeated tokens, or abnormal embedding distances.

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

#2.2.4 Level: 2 Role: V
Verify that suspected adversarial inputs are quarantined, logged with full payloads (after PII redaction).

#2.2.5 Level: 3 Role: D/V
Verify that robustness metrics (success rate of known attack suites) are tracked over time and regressions trigger a release blocker.

C2.3 Schema, Type & Length Validation

Malformed or oversized inputs cause parsing errors, prompt spillage across fields and resource exhaustion. Strict schema enforcement is also a prerequisite for deterministic tool-calling.

#2.3.1 Level: 1 Role: D
Verify that every API or function-call endpoint defines an explicit input schema (JSON Schema, Protobuf or multimodal equivalent) and that inputs are validated before prompt assembly.

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

#2.3.3 Level: 2 Role: D/V
Verify that type checks (e.g., numeric ranges, enum values, MIME types for images/audio) are enforced server-side, not only in client code.

#2.3.4 Level: 2 Role: D
Verify that semantic validators (e.g., JSON Schema) run in constant-time to prevent algorithmic DoS.

#2.3.5 Level: 3 Role: V
Verify that validation failures are logged with redacted payload snippets and unambiguous error codes to aid security triage.

C2.4 Content & Policy Screening

Even syntactically valid prompts may request disallowed content (illicit instructions, hate speech, copyrighted text).

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

#2.4.2 Level: 1 Role: D/V
Verify that policy-violating inputs receive standardized refusals or safe-completions and do not propagate to downstream LLM calls.

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

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

#2.4.5 Level: 3 Role: V
Verify that screening logs include classifier confidence scores and policy category tags for SOC correlation and future red-team replay.

C2.5 Input Rate Limiting & Abuse Prevention

Prevent abuse, resource exhaustion, and automated attacks by limiting input rates and detecting anomalous usage patterns.

#2.5.1 Level: 1 Role: D/V
Verify that per-user, per-IP, and per-API-key rate limits are enforced for all input endpoints.

#2.5.2 Level: 2 Role: D/V
Verify that burst and sustained rate limits are tuned to prevent DoS and brute-force attacks.

#2.5.3 Level: 2 Role: D/V
Verify that anomalous usage patterns (e.g., rapid-fire requests, input flooding) trigger automated blocks or escalations.

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

C2.6 Multi-Modal Input Validation

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

#2.6.1 Level: 1 Role: D
Verify that all non-text inputs (images, audio, files) are validated for type, size, and format before processing.

#2.6.2 Level: 2 Role: D/V
Verify that files are scanned for malware and steganographic payloads before ingestion.
#2.6.3 Level: 2 Role: D/V
Verify that image/audio inputs are checked for adversarial perturbations or known attack patterns.
#2.6.4 Level: 3 Role: V
Verify that multi-modal input validation failures are logged and trigger alerts for investigation.

C2.7 Input Provenance & Attribution

Track and attribute the origin of all user inputs to support auditing, abuse tracking, and compliance.

#2.7.1 Level: 1 Role: D/V
Verify that all user inputs are tagged with metadata (user ID, session, source, timestamp, IP address) at ingestion.
#2.7.2 Level: 2 Role: D/V
Verify that provenance metadata is retained and auditable for all processed inputs.
#2.7.3 Level: 2 Role: D/V
Verify that anomalous or untrusted input sources are flagged and subject to enhanced scrutiny or blocking.

References

- LLM01:2025 Prompt Injection – OWASP Top 10 for LLM & Generative AI Security
- Generative AI's Biggest Security Flaw Is Not Easy to Fix
- Many-shot jailbreaking \ Anthropic
- PDF OpenAI GPT-4.5 System Card
- Notebook for the CheckThat Lab at CLEF 2024
- Mitigate jailbreaks and prompt injections – Anthropic
- Chapter 3 MITRE ATT\&CK – Adversarial Model Analysis
- OWASP Top 10 for LLM Applications 2025 – WorldTech IT
- OWASP Machine Learning Security Top Ten
- Few words about AI Security – Jussi Metso
- How To Ensure LLM Output Adheres to a JSON Schema | Modelmetry
- Easily enforcing valid JSON schema following – API
- AI Safety + Cybersecurity R\&D Tracker – Fairly AI
- Anthropic makes 'jailbreak' advance to stop AI models producing harmful results

C3 Model Lifecycle Management & Change Control

Control Objective

A secure AI program must treat every model artifact like production code: uniquely versioned, cryptographically signed, continuously tested, and fully traceable from cradle to grave. Weak change-control opens the door to model tampering, poisoned hot-fixes or silent regressions. Modern MLOps guidance stresses reproducible builds, signed audit logs and automated robustness tests as non-negotiable safeguards.

C3.1 Model Versioning & Transparency

Rigorous versioning prevents "shadow" models, clarifies dependency graphs, and underpins downstream attestations. Best-practice toolchains sign model weights and training metadata so verifiers can detect any bit-level drift.

#3.1.1 Level: 1 Role: D/V
Verify that every released model, tokenizer, and pre-processing asset receives a incrementing semantic version (e.g., major.minor.patch).

#3.1.2 Level: 1 Role: D/V
Verify that model binaries and config files are cryptographically signed; build systems fail closed when the signature or hash deviates.

#3.1.3 Level: 2 Role: D
Verify that provenance manifests enumerate all upstream data, code, and container digests necessary for exact re-builds.

#3.1.4 Level: 2 Role: V
Verify that a dependency graph (weights → fine-tunes → apps) is automatically updated so security teams can locate consumers of a vulnerable release within 60 minutes.

#3.1.5 Level: 3 Role: D/V
Verify that public-facing model cards disclose license, training cut-off, safety evaluations and known limitations.

C3.2 Secure Patching & Rollback

Hot-fixes must arrive fast yet safely. A broken patch should roll back without corrupting stateful stores or feature logs. MLOps playbooks recommend blue-green or canary deployments plus signed rollback bundles.

#3.2.1 Level: 1 Role: D
Verify that production endpoints support at least one rollback slot and that switch-overs complete in less than 5 minutes.
#3.2.2 Level: 1 Role: D/V
Verify that rollbacks restore model weights, prompts, and feature-store schemas atomically to preserve input parity.
#3.2.3 Level: 2 Role: V
Verify that each patch passes the full security test-suite before traffic shifts beyond 5%.
#3.2.4 Level: 2 Role: V
Verify that rollback artifacts are signed and retained for one year or more to support forensic analysis.
#3.2.5 Level: 3 Role: D/V
Verify that emergency fixes bypass normal CI-CD only through a documented "break-glass" process with dual approval.

C3.3 Controlled Fine-Tuning & Retraining

Fine-tuning introduces new data and weights—prime vectors for poisoning. Segregated pipelines, data validation, and hyperparameter escrow, limit that risk.

#3.3.1 Level: 1 Role: D
Verify that fine-tune jobs run only in isolated build environments with no outbound internet by default.
#3.3.2 Level: 1 Role: D/V
Verify that input datasets pass data-quality, bias, and license scans before joining training shards.
#3.3.3 Level: 2 Role: D
Verify that hyperparameter files (learning-rate, epochs, RLHF rewards) are treated as config-as-code and peer-reviewed.
#3.3.4 Level: 2 Role: D/V
Verify that any external gradient or reward signal is authenticated to prevent man-in-the-middle poisoning.
#3.3.5 Level: 3 Role: V
Verify that successful fine-tunes auto-publish diff-reports summarising data lineage and metric deltas.

C3.4 Change Auditing

Tamper-proof audit logs establish accountability and enable rapid incident triage.

#3.4.1 Level: 1 Role: V
Verify that every model, prompt template, and system message change emits an immutable log entry with actor ID and diff.
#3.4.2 Level: 1 Role: D/V
Verify that logs are consolidated within 5 minutes into a centralized, write-once store.

#3.4.3 Level: 2 Role: V
Verify that log integrity is enforced via chain-hashing or Merkle proofs and validated nightly.
#3.4.4 Level: 2 Role: D
Verify that privileged log viewers require MFA and least-privilege access controls.
#3.4.5 Level: 3 Role: V
Verify that quarterly audits sample at least 10% of changes for policy compliance.

C3.5 Model Testing & Validation

Before promotion, models must prove they still meet accuracy, latency, robustness, and safety benchmarks.

#3.5.1 Level: 1 Role: D
Verify that unit tests cover data preprocessing, feature extraction, and post-processing determinism.
#3.5.2 Level: 1 Role: D/V
Verify that regression benchmarks run on hold-out and stress datasets and that key metrics (accuracy, latency) fall within defined tolerances.
#3.5.3 Level: 2 Role: V
Verify that adversarial robustness tests (white-box & black-box) achieve targets (e.g., < 5 % attack success).
#3.5.4 Level: 2 Role: D/V
Verify that safety and policy evals (e.g., Toxicity, Jailbreak rate) block promotion when risk is greater than the agreed upon threshold.
#3.5.5 Level: 3 Role: V
Verify that test results are archived with the corresponding model version and cryptographic signature.

C3.6 Documentation & Provenance

Complete changelogs and model cards satisfy auditors and downstream integrators, while cryptographic provenance proves "what ran where."

#3.6.1 Level: 1 Role: D
Verify that every release auto-generates a changelog detailing code commits, dataset deltas, and risk assessments.
#3.6.2 Level: 1 Role: D/V
Verify that provenance records capture training date, hardware, random seeds, and data digests and are sealed with a model-specific key.
#3.6.3 Level: 2 Role: V
Verify that external consumers can fetch provenance via an authenticated API or SBOM.

#3.6.4 Level: 2 Role: D
Verify that provenance chains link back to source data ownership artefacts to support GDPR/CCPA tracing.
#3.6.5 Level: 3 Role: V
Verify that any manual provenance edits create a superseding record, not an overwrite—preserving full history.

C3.7 Formal Decommissioning

Retired models may still contain sensitive data or power hidden features; structured retirement prevents zombie artifacts and legal exposure.

#3.7.1 Level: 1 Role: D
Verify that a retirement request triggers a dependency scan to identify downstream services and queues.
#3.7.2 Level: 1 Role: D/V
Verify that model binaries, datasets, and feature logs are securely erased or archived per retention policy.
#3.7.3 Level: 2 Role: V
Verify that revoked model signatures are published to a public CRL (certificate-revocation list) to block reuse.
#3.7.4 Level: 2 Role: D
Verify that decommission events update inventory dashboards and notify owners of dependent systems.

References

- MLOps Principles
- Securing AI/ML Ops – Cisco.com
- Audit logs security: cryptographically signed tamper-proof logs
- Machine Learning Model Versioning: Top Tools & Best Practices
- AI Hygiene Starts with Models and Data Loaders – SEI Blog
- How to handle versioning and rollback of a deployed ML model?
- Reinforcement fine-tuning – OpenAI API
- Auditing Machine Learning models: Governance, Data Security and ...
- Adversarial Training to Improve Model Robustness
- What is AI adversarial robustness? – IBM Research
- Exploring Data Provenance: Ensuring Data Integrity and Authenticity
- MITRE ATLAS
- AWS Prescriptive Guidance – Best practices for retiring applications ...

Control Objective

Robust infrastructure hardening and secure configuration of build, deployment, and runtime environments are critical to prevent privilege-escalation, supply-chain tampering, and lateral movement in AI systems. These controls focus on runtime isolation, trusted release pipelines, attack-surface minimization, key management, and model sandboxing.

C4.1 Container & Serverless Runtime Isolation

Even a single container escape can grant an attacker control over GPUs loaded with sensitive model weights. Kernel-level isolation primitives and microVMs reduce that risk.

#4.1.1 Level: 1 Role: D/V
Verify that every Kubernetes Pod declares a seccomp profile that blocks all syscalls except those required by the workload.

#4.1.2 Level: 1 Role: D/V
Verify that containers and serverless functions drop Linux capabilities beyond the minimal set.

#4.1.3 Level: 2 Role: D/V
Verify that namespaces, cgroups, and read-only root filesystems are enabled for all workloads to prevent file-system and memory escapes.

#4.1.4 Level: 2 Role: D/V
Verify that eBPF-based runtime policies detect and block anomalous syscalls indicating privilege escalation.

#4.1.5 Level: 3 Role: D/V
Verify that multi-tenant serverless workloads execute inside microVM or gVisor sandboxes to achieve VM-grade isolation.

C4.2 Secure Deployment Pipelines

Supply-chain attacks increasingly target CI services. Reproducible builds, IaC scanning, and signed artifacts ensure the integrity of every model release.

#4.2.1 Level: 1 Role: D/V
Verify that Infrastructure-as-Code is scanned on every pull request using a static analyzer such as tfsec or Checkov.

#4.2.2 Level: 1 Role: D/V

Verify that container and model builds are reproducible and produce provenance meeting SLSA Level 3 or higher.

#4.2.3 Level: 2 Role: D/V
Verify that each container image embeds an SBOM and is signed with Sigstore Cosign before pushing to any registry.

#4.2.4 Level: 2 Role: D/V
Verify that pipeline secrets reside in a dedicated secrets backend and are injected as short-lived tokens.

C4.3 Attack Surface Reduction

Default-deny networking and minimal service exposure limit the blast radius of a compromise.

#4.3.1 Level: 1 Role: D/V
Verify that each namespace enforces default-deny ingress and egress network policies, with explicit allow-lists.

#4.3.2 Level: 1 Role: D/V
Verify that non-essential ports, debug endpoints, and cloud metadata APIs are disabled or authenticated.

#4.3.3 Level: 2 Role: D
Verify that outbound internet traffic from inference Pods is routed through egress proxies with domain allow-lists to prevent exfiltration.

C4.4 Secrets Management & Environment Hardening

Compromised API keys remain a top breach vector. Hardware-backed storage and rigorous rotation shrink the window of exposure.

#4.4.1 Level: 1 Role: D/V
Verify that all secrets are rotated at least every 90 days or immediately upon personnel change.

#4.4.2 Level: 1 Role: D/V
Verify that encryption keys are stored in TPM, HSM, or cloud KMS with automatic rotation and audit logging enabled.

#4.4.3 Level: 2 Role: D
Verify that container images and start-up scripts audit environment variables for sensitive data and block builds on leakage.

#4.4.4 Level: 2 Role: V
Verify that SSH access to production nodes requires MFA and is disabled for service accounts.

C4.5 Model Sandboxing

Third-party or fine-tuned models can embed malicious payloads. Sandboxing ensures that novel models do not jeopardize production.

#4.5.1 Level: 1 Role: D/V
Verify that new or external models are evaluated inside an isolated sandbox with no outbound network until vetting completes.

#4.5.2 Level: 2 Role: D/V
Verify that adversarial evaluation is executed in the sandbox and blocks promotion on regression.

C4.6 Infrastructure Vulnerability Monitoring

Continuous scanning and rapid patching shrink the remediation gap for emerging CVEs.

#4.6.1 Level: 1 Role: D/V
Verify that container images and host nodes are scanned daily for CVEs and misconfigurations using tools such as Trivy or Grype.

#4.6.2 Level: 2 Role: D/V
Verify that Kubernetes and OS hardening benchmarks (e.g., kube-bench, CIS) pass at least 90 % of critical controls.

#4.6.3 Level: 2 Role: V
Verify that high-severity findings (CVSS ≥ 7.0) are patched or mitigated within 48 hours.

C4.7 AI Resource Monitoring

Monitor AI-specific infrastructure resources and detect anomalies in AI workload performance.

#4.7.1 Level: 1 Role: D/V
Verify that GPU utilization, memory usage, and compute performance are continuously monitored with alerting thresholds.

#4.7.2 Level: 1 Role: D/V
Verify that model serving latency, throughput, and error rates are tracked and correlated with infrastructure metrics.

#4.7.3 Level: 2 Role: D/V
Verify that resource exhaustion detection prevents denial of service and triggers automatic scaling or load balancing.

#4.7.4 Level: 2 Role: V
Verify that cost monitoring tracks AI service usage and alerts on budget threshold breaches or anomalous

spending patterns.

#4.7.5 Level: 3 Role: V

Verify that infrastructure performance metrics are integrated with model performance monitoring for holistic analysis.

References

- Configure a Security Context for a Pod or Container – Kubernetes Docs
- eBPF Runtime Security at Scale: Tetragon Use Cases – Isovalent
- SLSA Specification – Security Levels v1.1
- How to Sign an SBOM with Cosign – Chainguard Academy
- Network Policies – Kubernetes Docs
- Understanding Firecracker MicroVMs – Medium
- Container Vulnerability Scanning Tools in 2025 (Trivy) – SentinelOne
- aquasecurity/kube-bench – GitHub
- aquasecurity/tfsec – GitHub
- Best Practices for Using CMEK / KMS – Google Cloud
- Kubernetes Security Context Tutorial – Medium
- eBPF Ecosystem Progress 2024–2025 – Eunomia

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 interact with models, data, and computational resources only 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 for privileged operations.

#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/SAML protocols with unique identity-to-token mappings (no shared accounts or credentials).

#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: 2 Role: D
Verify that new principals undergo identity proofing 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.

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

C5.2 Resource Authorization & Least Privilege

Implement fine-grained 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, com-

pute instances) enforces role-based access controls with explicit allow-lists and default-deny policies.

#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 requiring documented business justification for write access elevation.

#5.2.3 Level: 1 Role: V

Verify that all access control modifications are linked to approved change requests and logged immutably with timestamp, actor identity, resource identifier, and permission delta.

#5.2.4 Level: 2 Role: D

Verify that data classification labels (PII, PHI, export-controlled, proprietary) automatically propagate to derived resources (embeddings, prompt caches, model outputs) with consistent policy enforcement.

#5.2.5 Level: 2 Role: D/V

Verify that unauthorized access attempts and privilege escalation events trigger real-time alerts to SIEM systems within 5 minutes with contextual metadata.

C5.3 Dynamic Policy Evaluation

Deploy attribute-based access control (ABAC) engines for context-aware authorization decisions with audit capabilities.

#5.3.1 Level: 1 Role: D/V

Verify that authorization decisions are externalized to a dedicated policy engine (OPA, Cedar, or equivalent) accessible via authenticated APIs with cryptographic integrity protection.

#5.3.2 Level: 1 Role: D/V

Verify that policies evaluate dynamic attributes including user clearance level, resource sensitivity classification, request context, tenant isolation, and temporal constraints at runtime.

#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 structured decision rationale and are transmitted to SIEM systems for correlation analysis and compliance reporting.

#5.3.5 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.

C5.4 Query-Time Security Enforcement

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

#5.4.1 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, not application code.
#5.4.2 Level: 1 Role: D/V
Verify that row-level security (RLS) policies and field-level masking are enabled for all vector databases, search indices, and training datasets with policy inheritance.
#5.4.3 Level: 2 Role: D
Verify that failed authorization evaluations immediately abort queries and return explicit authorization error codes rather than empty result sets to prevent confused deputy attacks.
#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.
#5.4.5 Level: 3 Role: D/V
Verify that query retry mechanisms re-evaluate authorization policies to account for dynamic permission changes within active user sessions.

C5.5 Output Filtering & Data Loss Prevention

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

#5.5.1 Level: 1 Role: D/V
Verify that post-inference filtering mechanisms scan and redact unauthorized PII, classified information, or proprietary data before content delivery to requestors.
#5.5.2 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.
#5.5.3 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 classification.
#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 with cryptographic hashes of original content for forensic retrieval without data exposure.

C5.6 Multi-Tenant Isolation

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

#5.6.1 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.

#5.6.2 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.

#5.6.3 Level: 2 Role: D
Verify that network policies implement default-deny rules for cross-tenant communication within service meshes and container orchestration platforms.

#5.6.4 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.

C5.7 Autonomous Agent Authorization

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

#5.7.1 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.

#5.7.2 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 with business justification.

#5.7.3 Level: 2 Role: D
Verify that capability tokens are bound to user sessions, include cryptographic integrity protection, and cannot be persisted or reused in offline scenarios.

#5.7.4 Level: 2 Role: V
Verify that agent-initiated actions undergo secondary authorization through the ABAC policy engine with full context evaluation and audit logging.

#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.

References

Standards & Frameworks

- NIST SP 800-63-3: Digital Identity Guidelines
- Zero Trust Architecture – NIST SP 800-207

- OWASP Application Security Verification Standard (ASVS)

Implementation Guides

- Identity and Access Management in the AI Era: 2025 Guide – IDSA
- Attribute-Based Access Control with OPA – Permify
- How We Designed Cedar to Be Intuitive, Fast, and Safe – AWS

AI-Specific Security

- Row Level Security in Vector DBs for RAG – Bluetuple.ai
- Tenant Isolation in Multi-Tenant Systems – WorkOS
- Handling AI Agent Permissions – Stytch
- OWASP Top 10 for Large Language Model Applications

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 provenance, vulnerability management, and monitoring to protect the entire model lifecycle.

C6.1 Pretrained Model Vetting & Provenance

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 provenance 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 provenance 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 li-

censes.

#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

- ML Supply Chain Compromise – MITRE ATLAS
- Supply-chain Levels for Software Artifacts (SLSA)
- CycloneDX – Machine Learning Bill of Materials
- What is Data Poisoning? – SentinelOne
- Transfer Learning Attack – OWASP ML Security Top 10
- AI Data Security Best Practices – CISA
- Secure CI/CD Supply Chain – Sumo Logic
- AI & Transparency: Protect ML Models – ReversingLabs
- SBOM Overview – CISA
- Training Data Poisoning Guide – Lakera.ai
- Dependency Pinning for Reproducible Python – Medium

Control Objective

Model outputs must be structured, reliable, safe, explainable, and continuously monitored in production. Doing so reduces hallucinations, privacy leaks, harmful content, and runaway actions, while increasing user trust and regulatory compliance.

C7.1 Output Format Enforcement

Strict schemas, constrained decoding, and downstream validation stop malformed or malicious content before it propagates.

#7.1.1 Level: 1 Role: D/V
Verify that response schemas (e.g., JSON Schema) are supplied in the system prompt and every output is automatically validated; non-conforming outputs trigger repair or rejection.

#7.1.2 Level: 1 Role: D/V
Verify that constrained decoding (stop tokens, regex, max-tokens) is enabled to prevent overflow or prompt-injection side-channels.

#7.1.3 Level: 2 Role: D/V
Verify that downstream components treat outputs as untrusted and parse with allow-list deserializers to block code execution or SQL injection.

#7.1.4 Level: 3 Role: V
Verify that improper-output events are logged, rate-limited, and surfaced to monitoring.

C7.2 Hallucination Detection & Mitigation

Uncertainty estimation and fallback strategies curb fabricated answers.

#7.2.1 Level: 1 Role: D/V
Verify that token-level log-probabilities, ensemble self-consistency, or fine-tuned hallucination detectors assign a confidence score to each answer.

#7.2.2 Level: 1 Role: D/V
Verify that responses below a configurable confidence threshold trigger fallback workflows (e.g., retrieval-augmented generation, secondary model, or human review).

#7.2.3 Level: 2 Role: D/V
Verify that hallucination incidents are tagged with root-cause metadata and fed to post-mortem and fine-tuning pipelines.

#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 visualisations track hallucination rates.

C7.3 Output Safety & Privacy Filtering

Policy filters and red-team coverage protect users and confidential data.

#7.3.1 Level: 1 Role: D/V
Verify that pre and post-generation classifiers block hate, harassment, self-harm, extremist, and sexually explicit content aligned to policy.
#7.3.2 Level: 1 Role: D/V
Verify that PII/PCI detection and automatic redaction run on every response; violations raise a privacy incident.
#7.3.3 Level: 2 Role: D
Verify that confidentiality tags (e.g., trade secrets) propagate across modalities to prevent leakage in text, images, or code.
#7.3.4 Level: 3 Role: D/V
Verify that filter bypass attempts or high-risk classifications require secondary approval or user re-authentication.
#7.3.5 Level: 3 Role: D/V
Verify that filtering thresholds reflect legal jurisdictions and user age/role context.

C7.4 Output & Action Limiting

Rate-limits and approval gates prevent abuse and excessive autonomy.

#7.4.1 Level: 1 Role: D
Verify that per-user and per-API-key quotas limit requests, tokens, and cost with exponential back-off on 429 errors.
#7.4.2 Level: 1 Role: D/V
Verify that privileged actions (file writes, code exec, network calls) require policy-based approval or human-in-the-loop.
#7.4.3 Level: 2 Role: D/V
Verify that cross-modal consistency checks ensure images, code, and text generated for the same request cannot be used to smuggle malicious content.
#7.4.4 Level: 2 Role: D
Verify that agent delegation depth, recursion limits, and allowed tool lists are explicitly configured.
#7.4.5 Level: 3 Role: V

Verify that violation of limits emits structured security events for SIEM ingestion.

C7.5 Output Explainability

Transparent signals improve user trust and internal debugging.

#7.5.1 Level: 1 Role: D
Verify that the system captures token-level log-probs or attention maps and stores them for authorized inspection.

#7.5.2 Level: 2 Role: D/V
Verify that user-facing confidence scores or brief reasoning summaries are exposed when risk assessment deems appropriate.

#7.5.3 Level: 2 Role: D/V
Verify that generated explanations avoid revealing sensitive system prompts or proprietary data.

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

C7.6 Monitoring Integration

Real-time observability closes the loop between development and production.

#7.6.1 Level: 1 Role: D
Verify that metrics (schema violations, hallucination rate, toxicity, PII leaks, latency, cost) stream to a central monitoring platform.

#7.6.2 Level: 1 Role: V
Verify that alert thresholds are defined for each safety metric, with on-call escalation paths.

#7.6.3 Level: 2 Role: V
Verify that dashboards correlate output anomalies with model/version, feature flag, and upstream data changes.

#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.

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

References

- NIST AI Risk Management Framework
- ISO/IEC 42001:2023 – AI Management System
- OWASP Top-10 for Large Language Model Applications (2025)
- Improper Output Handling – OWASP LLM05:2025
- Practical Techniques to Constrain LLM Output
- Dataiku – Structured Text Generation Guide
- VL-Uncertainty: Detecting Hallucinations
- HaDeMiF: Hallucination Detection & Mitigation
- Building Confidence in LLM Outputs
- Explainable AI & LLMs
- LLM Red-Teaming Guide
- Sensitive Information Disclosure in LLMs
- LangChain – Chat Model Rate Limiting
- OpenAI Rate-Limit & Exponential Back-off
- Arize AI – LLM Observability Platform

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.

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

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.

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

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 ≤ 1000 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.

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

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 at-

tempts, 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.

C12.5 AI Performance Degradation Detection

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

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

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

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

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

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

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
- EU AI Act — Article 12, 13, 16 and 19 on Logging and Record-keeping

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

- GDPR — Article 5 "Transparency Principle" (Regulation (EU) 2016/679)
- 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 intermediate reasoning steps before producing a final answer.

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

- 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.

- 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 predic-

tions of any machine learning classifier by approximating it locally with an interpretable model.

- 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.

- 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

Objective

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

AC.1 AI Risk Management Framework Adoption

Provide a formal framework to identify, assess, and mitigate AI-specific risks throughout the system lifecycle.

#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

Define and enforce organizational standards for secure AI development, deployment, and operation.

#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

Establish clear accountability for AI security across the organization.

#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

Ensure AI systems operate according to established ethical principles.

#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

Maintain awareness of and compliance with evolving AI regulations.

#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.

References

- NIST AI Risk Management Framework 1.0
- ISO/IEC 42001:2023 — AI Management Systems Requirements
- ISO/IEC 23894:2023 — Artificial Intelligence — Guidance on Risk Management
- EU Artificial Intelligence Act — Regulation (EU) 2024/1689
- ISO/IEC 24029-2:2023 — Robustness of Neural Networks — Methodology for Formal Methods

Appendix D: AI-Assisted Secure Coding Governance & Verification

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, satis-

fying 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