ODA3 INSTITUTE

Practitioner Cheat Sheet

AI Security Control Validation Guide

DocID: ODA3-2026-07-CHT-SEC-007
Classification: Public — Practitioner Reference

CHT-SEC-007: AI Security Control Validation Guide

ODA3 Institute | Practitioner Cheat Sheet

Field	Detail
Classification	Public — Practitioner Reference
Intended Audience	Security Architects, AI Governance Leads, Compliance Officers
Evidence Confidence	Moderate–High (see inline tiers)
Review Cycle	Semi-annual, or following significant changes to AI systems, controls, or threat landscape
Framework Cross-References	Maps to GAISSF™ v1.0 (Assurance Track); control failures feed UAIF™ incident classification; classified failures route per AI-IRF™ v1.0
Related Publications	CHT-SEC-001 through CHT-SEC-006; CHT-SEC-008; CHT-SEC-009

GAISSF™, UAIF™, and AI-IRF™ are proprietary frameworks of ODA3 Pvt Ltd, referenced here under the GAISSF Ecosystem License (GEL) v1.0. This document maps to those frameworks; it does not certify, endorse, or attest to any organization's compliance.

Why This Matters

Most AI security programs can produce a control catalogue on request. Far fewer can produce evidence that any given control actually operated as claimed on a specific date, for a specific system. A written policy proves intent, not operation — and unlike conventional enterprise applications, AI systems shift underneath their own controls through model updates, prompt engineering changes, retrieval pipeline changes, and agent orchestration updates. A control validated at launch can quietly stop working after the next model version ships, without anyone noticing until it's tested — or until it fails during an incident.

This gap becomes expensive at exactly the wrong moment: during a regulatory inquiry, a customer due-diligence review, or the aftermath of an incident, when “the policy says we do this” is no longer acceptable and someone needs the log, the test result, or the rotation record that proves it.

This cheat sheet gives security architects and governance leads a rapid, evidence-tiered approach to validating that AI security controls are operating as claimed — not merely documented as claimed. It sits at the point where ODA3's three frameworks meet: GAISSF™ supplies the control catalogue and certification scoring this checklist feeds evidence into; UAIF™ provides the taxonomy for classifying what kind of failure occurred when a control check fails; AI-IRF™ v1.0 governs what happens operationally once a failure is classified and requires response. This guide covers validation only — it does not certify, classify, or respond. It produces the evidence those three activities consume.

1. Why Validation ≠ Documentation

A control that exists on paper is not a control that functions. This guide separates “control is described” from “control is demonstrated” at every checkpoint, and routes what happens next when the two don't match.

2. Validation, Testing, Assessment, and Assurance

These related activities serve different operational purposes and are frequently conflated:

Activity	Primary Objective
Implementation	Deploy a security control
Testing	Determine whether the control functions under defined conditions
Validation	Confirm the control achieves its intended security objective using objective evidence
Assessment	Evaluate the effectiveness of multiple controls across a system or process
Assurance	Establish justified confidence that controls continue to operate effectively over time

Validation is the operational bridge between implementation and assurance. A prompt injection filter can be successfully installed (implementation) and confirmed to execute (testing) while still failing validation — because validation demonstrates that it consistently blocks malicious prompts under representative conditions and generates appropriate logs and alerts [T1].

3. Principles of Effective Validation

- Objective — conclusions rest on observable evidence, not assumption
- Repeatable — equivalent validation activities produce comparable outcomes under similar conditions
- Risk-based — effort scales with the business impact of control failure
- Representative — reflects realistic operating conditions, not a clean lab environment
- Evidence-driven — every activity produces documented evidence (logs, configuration snapshots, test outputs, alert records)
- Continuous — integrated into ongoing operations, not a one-time pre-launch exercise [T2]

4. Core Validation Domains

A. Access & Identity Controls

☐ Model/API credentials rotate on a defined schedule (evidence: rotation logs, not policy text) [T1]

☐ Privileged access to training data and model weights is logged and independently reviewable [T1]

☐ Service-account sprawl assessed — orphaned credentials identified in last 90 days [T2]

☐ Break-glass/emergency access paths tested, not just documented [T3]

B. Input/Output Integrity

☐ Prompt injection defenses tested against a documented adversarial test set (not vendor claim alone) [T2]

☐ Output filtering validated against known jailbreak corpora in a controlled simulation [T3]

☐ Rate limiting and anomaly detection produce reviewable alert logs [T1]

☐ Adversarial input handling has a defined escalation path to incident response [T2]

☐ For RAG pipelines specifically: retrieval authorization, document filtering, and retrieval logging validated as a distinct sub-check, not assumed covered by prompt testing alone [T2]

C. Data Governance

☐ Training/fine-tuning data lineage documented and independently spot-checked [T2]

☐ PII/sensitive data handling in prompts and completions has a tested redaction path [T2]

- ☐ Data retention policy has a corresponding deletion verification mechanism [T1]
- ☐ Data poisoning detection tested against academic proxy datasets [T3]

D. Model Lifecycle Controls

- ☐ Model version changes trigger mandatory re-validation of prior control tests [T1]
- ☐ Rollback capability tested (not just documented) within the last cycle [T3]
- ☐ Shadow/champion-challenger deployment has monitored divergence thresholds [T2]
- ☐ Model deprecation/retirement has a verified decommissioning checklist [T2]
- ☐ Model hashes/signatures verified against deployment records, and unauthorized replacement attempts have been tested [T1]

E. Agentic & Autonomous Controls

- ☐ Tool-calling permissions tested against least-privilege expectations, not just configured [T2]
- ☐ Human-approval workflows tested with a controlled unauthorized-action attempt [T2]
- ☐ Agent execution logs confirmed complete across delegation chains [T1]
- ☐ Failure/escalation handling tested — including forced termination of an unsafe operation [T3]

F. Third-Party & Supply Chain

- ☐ Vendor/model-provider security attestations are current and independently verifiable [T2]
- ☐ Open-source model or library provenance checked against known-vulnerability registries [T1]
- ☐ Fourth-party dependencies (sub-processors of your model vendor) mapped at least annually [T3]

G. Monitoring & Detection

- ☐ Drift detection thresholds are defined, tested, and tied to an owner [T2]
- ☐ Alert fatigue assessed — false-positive rate measured, not assumed [T2]
- ☐ Detection-to-notification time measured against an internal SLA [T1]
- ☐ Monitoring pipeline itself has been checked for whether it's actually running, not just configured [T2]

5. Common Failure Modes by Domain

Domain	Recurring Failure Pattern
Access & Identity	Excessive privileges, shared admin accounts, inactive accounts left enabled, missing MFA
Input/Output Integrity	Filter bypass via obfuscated or multi-step prompts, inconsistent enforcement, missing prompt logging
Data Governance	Untracked lineage, redaction paths untested against edge cases
Model Lifecycle	Unknown/unsigned model versions, untracked configuration changes, unauthorized replacement
Agentic & Autonomous	Excessive tool permissions, missing approvals, incomplete audit trails, failure to terminate unsafe operations
Third-Party & Supply Chain	Stale vendor attestations, unmapped fourth-party dependencies
Monitoring & Detection	Silent failures, disabled alerts, excessive false positives/negatives masking real signal

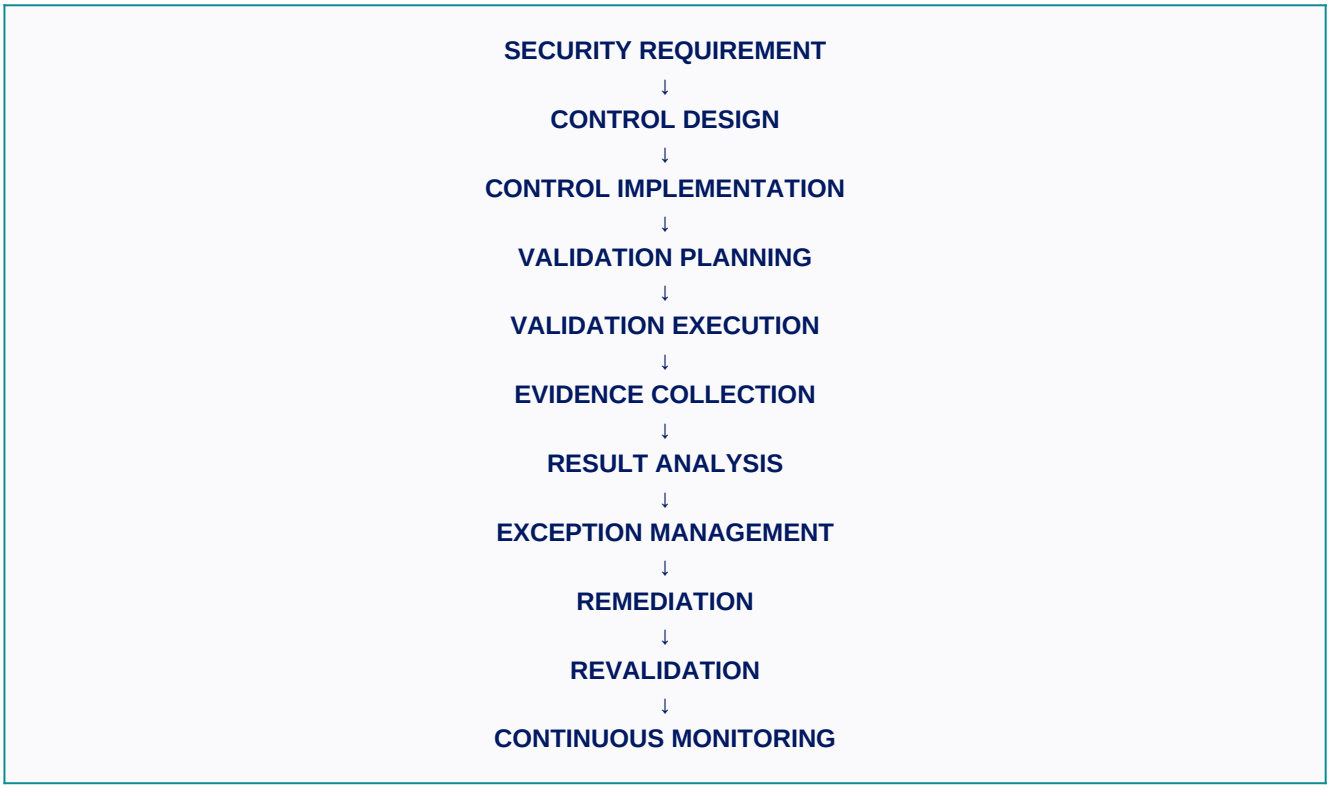
6. Validation Method by Tier

Tier	Method	What It Proves
T1	Primary log/artifact review	Control operated as claimed
T2	Secondary/independent corroboration	Control likely operated as claimed
T3	Controlled simulation or academic proxy	Control would perform under modeled conditions
T4	Anecdotal/unverified report	Signal only — never sufficient for a certification decision

Where relevant, T2–T3 evidence may draw on external verification schemes — for example OWASP AISVS assurance levels (Level 1 baseline, Level 2 production, Level 3 high-assurance) — as a reference point for depth of testing, not as a substitute for ODA3's own tier definitions above. Practitioners citing external framework version numbers or control counts (e.g., specific CSA AICM or MITRE ATLAS statistics) should verify the current figure at time of use — these frameworks update frequently, and a hardcoded figure risks going stale within months [T4].

7. The Validation Lifecycle

Validation is an ongoing operational process, not an isolated pre-launch event:



Each stage produces evidence that feeds forward into assessment and assurance activities [T1].

8. Validation Frequency: A Quick Decision Path

Not every control needs the same validation cadence. Use this to triage:

- Is the control mandated by a binding regulatory requirement (e.g., a high-risk-system obligation under an applicable AI law)?
- YES → Continuous validation. Automate evidence capture. Frequency: real-time / hourly.
- NO → Is the control highly susceptible to drift (e.g., semantic filters, model-dependent guardrails)?
- YES → Frequent validation. Automated test harness. Frequency: daily / per model update.
- NO → Is the control static and infrastructure-based (e.g., network segmentation, IAM roles)?
- YES → Periodic validation. Automated config scanning. Frequency: weekly / monthly.
- NO → Point-in-time validation. Manual review. Frequency: quarterly / annual.

This sits upstream of Section 4 — it tells you how often to re-run the checklist above for a given control, not what to check.

9. Quick Reference: Suggested Cadence by Domain

Domain	Typical Driver	Suggested Default Cadence
A. Access & Identity	Static infrastructure, some drift via credential sprawl	Periodic (weekly/monthly); continuous for break-glass paths
B. Input/Output Integrity	High drift — model updates shift guardrail behavior	Frequent (daily / per model update)
C. Data Governance	Regulatory exposure plus moderate drift	Frequent to continuous depending on data sensitivity
D. Model Lifecycle	Event-driven — tied to release cadence, not calendar time	Per model version change
E. Agentic & Autonomous	High drift — orchestration and tool integrations change frequently	Frequent (per orchestration/tool change)
F. Third-Party & Supply Chain	Low drift, external dependency	Periodic (quarterly); continuous for critical vendors
G. Monitoring & Detection	Validates the validators — needs its own check	Periodic (monthly); continuous for production alerting

Treat these as defaults to adjust for your own regulatory exposure and drift profile, not fixed rules. Domain D is deliberately event-driven rather than calendar-driven — validation should trigger on the model change itself, not wait for the next scheduled cycle.

10. Common Validation Pitfalls

- Treating documentation as evidence — a written policy proves intent, not operation.
- Testing only the easy case — validating a guardrail against one obvious bad input while leaving obfuscated or multi-step variants unchecked.
- One-time testing with no revalidation — a control validated at launch and never re-checked after a model or configuration change.
- Mismatched test environment — validating in a clean staging environment that doesn't reflect production noise, latency, or data volume.
- Measuring the wrong thing — tracking only successful blocks while ignoring override rates, bypass attempts, or exception volume.
- No validation of the validator — the test harness or monitoring pipeline itself is never checked for whether it's actually running.

- Static thresholds — a fixed pass/fail bar that doesn't account for normal variance in AI system behavior, producing alert fatigue or false confidence.
- Excessive reliance on automation — manual weaknesses (e.g., governance gaps, ownership ambiguity) remain unidentified when validation is treated as a purely technical exercise.

11. Practitioner Interview Prompts

Useful opening questions during a control walkthrough or vendor assurance review — each maps back to a domain in Section 4:

- (Access & Identity) “Walk me through what happens when a service credential is compromised — what's the evidence it gets rotated, not just that a policy says it should?”
- (Input/Output Integrity) “How was this guardrail last tested against something more sophisticated than a simple known-bad string?”
- (Data Governance) “If I asked you to prove this model's training data lineage right now, what would you show me?”
- (Model Lifecycle) “When was rollback last actually executed and verified, not just documented as a capability?”
- (Agentic & Autonomous) “Show me an example where an agent's action required approval and didn't get it automatically — what stopped it?”
- (Monitoring & Detection) “How would you know if your detection pipeline itself stopped working?”

12. Routing: What Happens When a Control Fails

This is the piece most control checklists skip.

- Failure detected → classify using UAIF™ taxonomy (failure type, sector context, harm category) — distinguishing, for example, control failure, control bypass, control absence, control misconfiguration, and operational misuse. Classification only — UAIF™ does not prescribe remediation. External threat taxonomies (e.g., MITRE ATLAS tactics/techniques) may inform how a technical failure is described, but do not replace UAIF™ classification.
- Classification complete → determine severity and routing per AI-IRF™ v1.0. Article 73 governs routing logic for AI-related incidents; low-severity findings may close at team level, higher-severity findings escalate per AI-IRF™ procedure.
- Response executed and closed → closure evidence (what was done, when, verified how) becomes a new control-evidence input, feeding back into the domain checklists above for the next validation cycle.
- Aggregated evidence across cycles → feeds GAISSF™ Assurance Track scoring.

13. Mapping to ODA3 Frameworks

Findings from this checklist are evidence inputs only. They do not themselves constitute a certification decision, an incident classification, or a response determination. Each framework retains its own controlled methodology:

- GAISSF™ — Findings feed Assurance Track scoring, but certification scoring, thresholds, and methodology remain controlled assets under the GAISSF™ Ecosystem License, assessed separately by qualified auditors.
- UAIF™ — Failed controls should be classified using UAIF™ taxonomy, but official mappings and classification methodology remain controlled; this checklist does not perform classification itself.

- AI-IRF™ v1.0 — Classified failures route per AI-IRF™ procedure (see Article 73 routing), but the full response methodology is the controlled operational layer and is not reproduced or substituted by this checklist.

13a. External Framework Landscape — Reference, Not Endorsement

Cited for practitioner orientation only. ODA3 Institute does not assert affiliation, certification, or equivalence with any of the following.

Framework	Relevant Validation-Related Provision
OWASP AISVS	Defines assurance levels (Level 1–3) usable as a reference point for depth of control testing [T4]
MITRE ATLAS	Documents adversarial tactics/techniques that can inform how a technical control failure is described, not how it's classified
CSA AICM	Publishes control objectives and auditing guidelines relevant to AI supply-chain control validation
NIST AI Risk Management Framework	Encourages ongoing measurement and testing of controls as part of the Measure/Manage functions [T4]
NIST SP 800-53	Provides enterprise security control families that may inform validation activity design
ISO/IEC 27001	Defines requirements for control evaluation and continual improvement within an information security management system
ISO/IEC 42001	Introduces AI management system requirements emphasizing documented controls and operational effectiveness
EU AI Act	Introduces documentation and monitoring obligations for certain AI systems that may influence validation activity design

External control counts and version numbers (e.g., specific CSA AICM control totals) change frequently — verify the current figure at time of use rather than citing a fixed number.

14. Real-World Incident / Pattern Reference

Illustrative only — public, disclosed patterns, not accusations or technical findings about named parties.

Pattern	Public Reference	Validation Lesson
Employees bypassed a documented data-handling policy by pasting sensitive code into a public LLM	Samsung employee data-exposure reports, widely covered 2023 [T2]	A written data-handling policy is not evidence that a technical control (e.g., DLP, egress filtering) actually enforces it
A dealership chatbot was manipulated via prompt injection into agreeing to an unauthorized commitment	Chevrolet dealership chatbot incident, widely reported December 2023 [T2]	Guardrails tested only against obvious bad inputs miss adversarial prompt variants
A chatbot's stated policy diverged from the operator's actual terms, later legally contested	Air Canada chatbot tribunal case, 2024 [T2]	Output filtering/guardrail validation needs testing against realistic multi-turn scenarios, not launch-time checks alone

15. Assessment & Assurance Readiness Checklist

Governance

- ☐ Control inventory maintained, with owners assigned
- ☐ Validation responsibilities and schedule documented and approved

Planning & technical coverage

- ☐ Validation objectives and success criteria defined before testing begins
- ☐ Every control in Sections 4A–4G has documented evidence, not just policy text, on file
- ☐ Validation cadence for each control matches the decision path in Section 8
- ☐ At least one control per domain has been tested against a realistic adversarial or edge-case scenario, not just the obvious case
- ☐ Test/monitoring harnesses themselves have been checked for whether they're actually running

Evidence & continuous improvement

- ☐ Validation evidence retained and independently verifiable
- ☐ Failed-control routing (Section 12) is documented and understood by the team, not improvised during an incident
- ☐ Controls revalidated after significant model, configuration, or architectural changes
- ☐ Evidence from the last validation cycle has been fed forward into GAISSF™ Assurance Track scoring inputs

16. Five Priority Actions

1. Pick one control per domain (Section 4) and confirm you can produce log/artifact evidence for it today — not just policy text
2. Test at least one guardrail against a realistic adversarial variant, not only the obvious bad input
3. Confirm your test harness or monitoring pipeline itself is being checked for whether it's running
4. Match validation cadence to the decision path in Section 8 rather than a single fixed schedule for all controls
5. Document the control-failure routing path (Section 12) before a failure happens, not during one

Notably Absent

- Physical security of model training infrastructure (separate facility-control guidance)
- Legal/contractual liability allocation for control failures
- Real-time production telemetry analysis — this is a point-in-time validation checklist, not continuous monitoring
- Sector-specific control thresholds (see UAIF™ SCPs for sector calibration, e.g. UAIF-CAL-BANKING-v1.0)
- Full AI-IRF™ v1.0 response procedure detail (see AI-IRF™ v1.0 primary documentation)
- GAISSF™ certification scoring methodology (controlled asset, not reproduced here)
- Detailed technical mapping to third-party frameworks (OWASP AISVS, MITRE ATLAS, CSA AICM, NIST AI RMF, NIST SP 800-53, ISO/IEC 27001, ISO/IEC 42001) — referenced above only as external context, not adopted as ODA3 methodology
- Penetration testing methodologies, digital forensic procedures, or malware analysis guidance

• Organization-specific risk management, governance, or assurance programs, which this checklist supports but does not replace

This cheat sheet maps to UAIF™ v1.0, AI-IRF™ v1.0, and GAISSF™ v1.0 under GEL v1.0. It does not constitute certification, compliance confirmation, or endorsement of any organization, product, or vendor.

ODA3 Institute — Where AI Governance meets Operational Reality.