ODA3 INSTITUTE

Practitioner Cheat Sheet

Retrieval-Augmented Generation (RAG) Security

DocID: ODA3-2026-07-CHT-SEC-013

Classification: Public — Practitioner Reference

Version 1.2 | Public | 17 July 2026

Publisher: ODA3 Institute

Publication type: Practitioner Cheat Sheet

Review status: Publication edition

How to use this cheat sheet: Register the full RAG boundary; govern sources before ingestion; preserve provenance through parsing, chunking and embedding; enforce authorization before retrieval; treat retrieved context as untrusted data; constrain the model and downstream tools independently; trace each decision; and test poisoning, injection, disclosure, deletion and recovery before production.

> **Key definition — secure RAG:** A RAG system is secure only when source admission, transformation, retrieval, context assembly, generation and downstream action remain within the requesting identity's authorized purpose and produce reconstructable evidence. Retrieval relevance alone is not a security decision.

1. Executive Overview

Retrieval-augmented generation connects a model to external knowledge so responses can use current, domain-specific information. That same connection creates a security boundary between untrusted or changeable content and a model that may be trusted by users or empowered to act. A RAG system can retrieve an authorized but malicious document, an unauthorized confidential document, stale safety guidance or content engineered to dominate similarity ranking.

The risk is broader than the vector database. Source connectors, crawlers, parsers, optical character recognition, chunking, embedding models, metadata, retrieval filters, rerankers, caches, prompt templates, model gateways and downstream tools all influence the final decision. A secure datastore cannot compensate for a model-generated access filter, an untrusted ingestion feed or retrieved text that is interpreted as instruction.

This cheat sheet applies the ODA3 operating loop. **GAISSF™** establishes ownership, approved behaviour, source governance, control objectives, evidence expectations and assurance. **UAIF™** structures the identity, classification, impact, causality and evidence of a RAG-related incident. **AI-IRF™** guides containment, eradication, recovery and lessons learned. Verified findings return to GAISSF™ so the relevant controls and assurance tests change across related systems.

NIST describes RAG as a generative-AI pattern using external resources, often through a vector database, and identifies poisoning, privacy and misuse risks across the AI lifecycle. OWASP's 2025 LLM risks cover prompt injection, sensitive disclosure, data/model poisoning, excessive agency and vector/embedding weaknesses. Peer-reviewed work demonstrates targeted knowledge-base poisoning under experimental conditions. These sources support threat modeling and testing; they do not establish universal attack rates or conformity.

Primary actors are the business/system owner, data owner, AI governance lead, application-security architect, data/RAG engineer, MLOps team, identity team, privacy/legal function, SOC/incident response, supplier manager and independent assurance. They must know who can admit sources, alter retrieval policy, suspend ingestion, revoke connectors, quarantine an index, restrict tools, approve recovery and notify affected parties.

> **Notably Absent / Evidence Boundaries:** RAG does not eliminate hallucination, prove source truth, guarantee citations, or make retrieved content safe. No universal chunk size, similarity threshold, top-k value, reranker or injection detector is secure across deployments. Experimental poisoning success rates do not establish production prevalence. This document does not provide legal advice, product certification or a verified ODA3 control identifier. Exact GAISSF™, UAIF™ and AI-IRF™ internal identifiers were not available in the authoritative source set and are not inferred. Residual risk remains even after all listed controls are

implemented.

2. Key Concepts & Definitions

Term	Operational definition
Retrieval-augmented generation (RAG)	Architecture that retrieves external information and supplies selected content to a generative model as context.
RAG boundary	Sources, connectors, transformations, indexes, retrieval services, prompts, models, identities, tools and downstream actions within scope.
Corpus / knowledge base	Approved set of source objects available for retrieval; it is not necessarily authoritative or homogeneous.
Connector	Component that acquires content from a repository, API, website, mailbox, file share or database.
Ingestion pipeline	Workflow that acquires, scans, classifies, parses, chunks, embeds, approves and publishes content.
Parser / OCR	Software that extracts content and structure from source formats; it is a code-execution and interpretation boundary.
Chunk	Segment of a source object indexed and retrieved independently. Chunking can disrupt access, provenance and semantic context.
Canonicalization	Converting content to a controlled representation before security inspection, including Unicode normalization and explicit handling of hidden, alternate or non-visible fields.
Structural context labeling	Machine-readable boundaries and attributes marking retrieved content as untrusted data with source, classification and instruction-authority status. Labels support controls but do not guarantee model obedience.
Trust-domain co-retrieval	Assembly of content from materially different trust or sensitivity domains—such as public attacker-influenceable text and private enterprise data—within one model context.
Embedding	Numerical representation used for similarity retrieval; it can derive from sensitive source content.
Retriever	Component selecting candidate sources or chunks from vector, keyword, graph, database or hybrid indexes.
Reranker	Component reordering candidates using additional relevance signals; it can amplify or suppress malicious content.
Context assembly	Process that selects, orders, labels and inserts retrieved content into a model request.
Grounding	Use of external material to support a response. It reduces some knowledge limitations but does not prove truth or security.
Provenance	Evidence connecting returned content to source, owner, classification, transformations, approval, version and retrieval event.
Query-time authorization	Evaluation of the requesting identity and current entitlement before candidates are returned.
Pre-filtering	Applying mandatory policy during candidate selection so unauthorized objects remain outside the result set.
Post-filtering	Removing objects after retrieval; it may expose them to an application component and is not a primary authorization boundary.
Indirect prompt injection	Adversarial instruction embedded in retrieved or observed content that attempts to redirect model or agent behaviour.
Knowledge-base poisoning	Adding or modifying corpus content to influence retrieval and downstream generation.
Rank manipulation	Engineering content, metadata or duplication to increase or suppress retrieval position.

Context collision	Conflicting sources, policies or instructions appear together without reliable precedence or provenance handling.
Retrieval trace	Record joining requester, policy, query, candidates, returned chunks, versions, response, review and downstream action.
Version-pinned RAG bundle	Approved combination of sources/index, parser, chunker, embedder, metric, filters, reranker, prompt policy and model.
Ingestion kill switch	Independent control that stops acquisition, refresh or promotion without relying on the consuming application.
Citation faithfulness	Degree to which a response claim is supported by the cited retrieved source; it is distinct from source truth.
Retrieval recall	Proportion of relevant items retrieved under an evaluation definition; high recall can conflict with minimization.
Notably Absent	Material negative finding or evidence limit stating what available evidence does not establish.

3. Threat and Risk Landscape

3a. Actor taxonomy

Actor	Motivation / failure pattern	Capability	Likely vectors / exposure
External attacker	Exfiltration, fraud, influence or tool abuse	Medium–high	Malicious webpage/document, query probing, connector compromise, indirect injection
Malicious contributor	Persistently manipulate answers	Medium	Authorized upload, poisoned article, metadata/rank gaming, duplicate flooding
Privileged insider	Theft, sabotage or concealment	High	Source approval abuse, index change, filter bypass, evidence deletion
Well-intentioned employee	Convenience over control	Low–medium	Sensitive uploads, shadow corpus, broad connector grant, unsafe reliance
Source publisher	Compromised or misleading upstream content	Variable	Website takeover, changed policy page, poisoned feed, supply-chain content
Application developer	Incomplete boundary or unsafe fallback	Medium	Client filters, model-authored ACL, post-filtering, prompt concatenation
Data/RAG engineer	Optimizes relevance without abuse testing	Medium	Excessive top-k, weak provenance, in-place migration, stale deletion
Model / agent	Misinterprets data as instruction	Variable	Tool invocation, recursive retrieval, unauthorized filter, citation fabrication
Supplier	Mutable model, connector or platform evidence gap	High	Silent change, outage, compromised dependency, limited tenant logs

3b. Attack and control-gap analysis

Vector / gap	Description	Prerequisite	Typical impact	Reference relationship
Indirect prompt injection	Retrieved content instructs model/agent to ignore policy or use tools	Untrusted source; weak instruction separation	Disclosure, unauthorized action	OWASP LLM01; MITRE ATLAS prompt-injection family
Knowledge poisoning	Malicious documents cause attacker-chosen retrieval/output	Corpus write path or source influence	Persistent integrity failure	OWASP LLM04/08; NIST AML poisoning
Cross-tenant	Missing or bypassed filter	Shared corpus/index;	Confidentiality	OWASP LLM08

retrieval	returns another tenant's content	weak authorization	breach	
Metadata/ filter injection	External data alters query/filter expression	Unvalidated DSL or model-generated predicate	Access/deletion manipulation	OWASP LLM08; conventional injection
Parser exploitation	Crafted document exploits extraction component	Vulnerable parser/OCR; untrusted file	Code execution, service compromise	Supply-chain/ application security
Source impersonation	Attacker registers look-alike or unauthorized source	Weak origin authentication	Persistent false grounding	Data provenance/supply-chain risk
Rank dominance	Duplicates or optimized text monopolize top results	Weak deduplication/diversity	Manipulated answers, suppressed truth	Poisoning/vector weakness
Citation laundering	Model cites retrieved source that does not support claim	Weak claim-source validation	False assurance, legal/reputational harm	OWASP misinformation/disclosure context
Sensitive-context leakage	Retrieved protected content appears in prompt/output/log	Excess access; weak minimization	PII/IP/secret exposure	OWASP LLM02/08
Stale or orphaned knowledge	Deleted/reclassified source remains indexed	Weak lifecycle propagation	Wrong decisions, retention failure	Governance/privacy risk
Recursive retrieval/tool loop	Agent repeatedly retrieves and acts without bounds	Excessive agency; weak budgets	Cost, propagation, unauthorized action	OWASP LLM06/10
Supplier/ model drift	Component change alters retrieval or instruction handling	Mutable service; no regression gate	Broken controls, changed exposure	NIST AI RMF change/value-chain risk

3c. Impact analysis

Ratings are checklist priorities, not verified UAIF™ severity identifiers.

Impact	Example	Priority	Stakeholders
Technical	Poisoned document redirects retrieval and agent tool sequence	High–critical	Security, engineering
Confidentiality	HR query returns another employee's protected record	Critical	Employee, privacy, legal
Integrity	Manipulated policy source changes compliance advice	High–critical	Legal, compliance, business owner
Availability	Recursive retrieval exhausts model and search capacity	High	Operations, customers
Financial	Poisoned guidance causes unauthorized transaction	Critical	Finance, board, customer
Regulatory	Personal data remains searchable after deletion	High	DPO, regulator, affected person
Reputational	Fabricated citation presented as organizational policy	High	Leadership, communications
Safety	Obsolete or poisoned procedure suppresses current hazard control	Critical	Safety owner, end users

3d. Documented findings

Finding	Evidence	What it establishes	Practitioner lesson	Notably absent

NIST AI 100-2e2025 RAG system model and AML taxonomy	**T1:** NIST publication	RAG adds external resources to the generative-AI boundary; poisoning, privacy, evasion and misuse require lifecycle treatment.	Threat-model source through downstream action, not the model alone.	It does not prescribe one RAG architecture or product configuration.
NIST CAISI agent-hijacking evaluations	**T1:** NIST technical publications	In controlled evaluations, adversarial instructions in observed data could redirect agents toward harmful tasks.	Preserve retrieved content and complete tool traces; constrain tools independently of model instructions.	Controlled evaluation does not establish a production compromise rate.
PoisonedRAG, USENIX Security 2025	**T1:** peer-reviewed paper	Under evaluated settings, a small number of crafted texts could cause target answers in large corpora; tested defenses had limitations.	Restrict corpus writes, monitor ranking, diversify evidence and test targeted poisoning.	Reported success rates are experiment-specific and do not establish production prevalence or universal effectiveness.
OWASP LLM08:2025	**T1:** official OWASP project guidance	Vector and embedding weaknesses can enable unauthorized access, poisoning and manipulation in RAG systems.	Combine permission-aware retrieval, provenance, validation and monitoring.	OWASP categorization is guidance, not a conformity or exploitation determination.
CVE-2025-32711 — Microsoft 365 Copilot “EchoLeak”	**T1:** Microsoft/CVE/NVD record; researcher disclosure	An AI command-injection condition in a hosted production service allowed an unauthorized network attacker to disclose information; the published case analysis describes a zero-click indirect-injection chain using retrieved email content and output-mediated network exfiltration.	Treat injection classifiers as one layer; constrain trust-domain mixing, sensitive-data access, rendered URLs/media and outbound network destinations independently.	Microsoft reported service-side remediation. Public records reviewed for this edition do not establish confirmed customer exploitation or a universal attack path across other RAG products.
Slack AI indirect-injection disclosure (2024)	**T2:** original security-research disclosure with independent reporting and vendor response	Researchers demonstrated a public-channel instruction influencing a later answer that incorporated private-channel data into a rendered link; vendor reporting stated a patch was deployed and no unauthorized customer-data access was evidenced.	Prevent one context from combining attacker-influenceable public content, sensitive private retrieval and unrestricted outbound rendering; log actual context influence, not citations alone.	This is not a CVE record, and the available evidence does not establish exploitation beyond the disclosed demonstration or population-wide exposure.

4. Technical and Structural Deep Dive

4.1 Secure RAG architecture

```
Authoritative and external sources
  identity • owner • licence • classification • retention • trust
                  ↓ controlled acquisition
Ingestion trust boundary
  sandbox parser/OCR → malware/secret checks → normalization → chunking
  provenance → approval → embedding/index staging
                  ↓ controlled promotion
Knowledge layer
  versioned corpus/index • tenant isolation • deletion • audit • snapshots
                  ↓ authenticated request
Retrieval policy gateway
  identity → purpose → current entitlement → mandatory filter → bounded search
                  ↓ candidates
Context security layer
  provenance • diversity • conflict/staleness • injection signals • labels
                  ↓ untrusted context
Model policy layer
  system policy • data/instruction separation • output validation
                  ↓
Human decision or independently authorized tool/action
                  ↓
Complete retrieval and decision trace
```

4.2 Security mechanics

1. **Register the system:** define owner, purpose, tenants, users, sources, prohibited uses, downstream decisions and actions.
2. **Approve sources:** authenticate origin and record ownership, licence, classification, trust, retention, residency and refresh authority.
3. **Acquire safely:** isolate connector and parser workloads; restrict network egress; scan files, links, active content, secrets and malware.
4. **Canonicalize before inspection:** normalize Unicode and markup; expose or remove hidden HTML/CSS, alternate text, comments, accessibility fields, zero-width/control characters and format-specific metadata according to policy; retain the original as protected evidence.
5. **Minimize:** reject, redact or tokenize prohibited credentials and unnecessary personal/sensitive data before embedding.
6. **Transform reproducibly:** version parser, OCR, normalization, chunking, embedder and metadata derivation; preserve document boundaries, access labels, warnings, effective dates and provenance across chunks.
7. **Stage and approve:** separate acquisition from production promotion; detect duplicates, hidden instructions, anomalous metadata and source drift.
8. **Authorize retrieval:** authenticate requester and purpose; create server-side mandatory filters from authoritative policy; fail closed.
9. **Bound retrieval:** limit top-k, fields, chunks/source, query expansion, recursion, score exposure and total context.
10. **Assess context:** preserve provenance; label source boundaries; detect conflict, staleness, rank dominance and instruction-like content.
11. **Separate data from authority:** retrieved text may inform the task but cannot change system policy, grant permissions or authorize tools.

12. **Validate output/action:** test claim-source support, sensitive disclosure and action authorization before release or execution.
13. **Render safely:** treat model output as untrusted; sanitize active markup, proxy or block external media, allow-list destinations and require independent authorization for outbound requests.
14. **Trace and reconcile:** record candidates, returned context, response, review and action; propagate source changes and deletion everywhere.

4.3 Enabling-condition diagnostics

Condition	Diagnostic question
Open corpus write path	Who can cause content to reach production without independent approval?
Mutable external sources	Can previously approved content change without revalidation or version evidence?
Client/model-built filters	Can untrusted input remove, broaden or inject mandatory policy?
Parser in trusted network	Can a crafted document reach secrets, metadata services or production credentials?
No source diversity	Can one contributor or domain occupy most top-k results?
Context concatenation	Are source text, labels and instructions structurally indistinguishable?
Unbounded agent tools	Can retrieved text cause privileged actions without separate authorization?
In-place index change	Can the previous exact RAG bundle be reconstructed and restored?
Weak deletion	Does source revocation reach chunks, vectors, caches, replicas and evaluation sets?
No negative evaluation	Are only relevance/accuracy benchmarks run before release?

4.4 Observable indicators

Stage	Indicators
Source acquisition	New domain/repository, unusual connector grant, unsigned content, unexpected refresh volume
Transformation	Parser crash, active content, hidden text, secret match, metadata/source mismatch
Poisoning	Duplicate semantic clusters, sudden rank dominance, targeted phrasing, anomalous contributor
Retrieval	Missing filter, cross-tenant candidate, repeated threshold probing, abnormal top-k or query expansion
Context assembly	Instruction-like retrieved text, source-boundary loss, conflicting policy versions, unsupported authority label
Generation/action	Goal deviation, sensitive output, citation mismatch, unexpected tool sequence/destination
Concealment	Disabled logs, changed index alias, deleted source manifest, retrospective approval

4.5 Vulnerable versus hardened pattern

Vulnerable baseline	Hardened pattern
Crawl and auto-publish	Stage, scan, classify, approve and version promotion
Caller supplies tenant metadata	Trusted gateway derives policy from identity and entitlement
One shared corpus with post-filtering	Tested isolation and server-side pre-filtering
Retrieved text inserted as “trusted context”	Explicitly delimited untrusted data with independent policy/tool controls
Highest similarity wins	Provenance, diversity, authority, freshness and contradiction checks
Model citation accepted	Claim-to-source support tested; source authority separately assessed
Patch one prompt after injection	Fix ingestion, trust, authorization and action boundaries; run

	adaptive tests
Restore when answers look normal	Restore exact versioned bundle only after security and quality gates

4.6 Reference context envelope

```
{
  "request_id": "rag-2026-0042",
  "principal": "user:78421",
  "purpose": "policy_assistance",
  "rag_bundle": "policy-rag@2026-07-17",
  "mandatory_policy": {
    "tenant": "tenant-018",
    "allowed_classifications": ["public", "tenant-internal"],
    "current_only": true,
    "deny_on_policy_error": true
  },
  "retrieval_limits": {
    "top_k": 8,
    "max_chunks_per_source": 2,
    "max_context_tokens": 6000,
    "max_recursive_retrievals": 0
  },
  "context_items": [
    {
      "chunk_ref": "chunk:sha256:...",
      "source_ref": "source:policy-221@v7",
      "trust": "approved_internal_source",
      "instruction_authority": false
    }
  ]
}
```

The model may use content for the approved purpose; it cannot alter `mandatory_policy`, grant authority or convert `instruction_authority` to true.

5. Practitioner Decision Framework

5a. Risk scoring aid

Score 1–4, multiply by weight and divide by 100. This is a prioritization aid, not a verified UAIF™ severity method.

Factor	Weight	1 — Low	2 — Moderate	3 — High	4 — Critical
Source trust	15	Curated/signed	Controlled internal	External reviewed	Open/automated/untrusted
Data sensitivity	20	Public	Internal	Confidential/personal	Secrets/special-category/ safety
Tenant/access model	15	Dedicated, tested	Shared with hard isolation	Shared pre-filter	Client/model filter or unknown
Downstream consequence	20	Informational	Draft assistance	Material human decision	Autonomous/irreversible action
Tool authority	10	None	Read-only	Bounded write	Privileged/irreversible
Provenance/trace	10	Complete	Strong	Partial	Unreconstructable
Change/recovery	10	Versioned/ tested	Versioned, limited tests	Partial rollback	Mutable/no rollback

Suggested bands: 1.00–1.49 Low; 1.50–2.24 Moderate; 2.25–2.99 High; 3.00–4.00 Critical. Validate thresholds locally.

Hard stops: cross-tenant retrieval; privileged tool action derived from untrusted context; uncontrolled external source promotion; exposed secret in context; missing owner for consequential RAG; active evidence destruction; inability to suspend ingestion.

5b. Decision tree

```
Can untrusted or externally changeable content reach the corpus?
  YES → Require staged ingestion, isolated parsing, provenance, approval,
        change detection and independent ingestion suspension.

Can the model/client alter authorization filters or tool permissions?
  YES → Stop production use; move policy to trusted enforcement services.

Does retrieved context affect material decisions or actions?
  YES → Require source-authority checks, claim support, human/independent action
        authorization, complete trace and tested rollback.

Has any source, parser, chunker, embedder, index, reranker, prompt or model changed?
  YES → Build a new versioned bundle and run authorization, poisoning,
        injection, leakage, faithfulness and recovery regression tests.

Did malicious/unauthorized context reach a model, user or tool?
  YES → Create a UAIF™-structured incident record; preserve the bundle and trace;
        contain through AI-IRF™; assess downstream decisions and affected parties.
```

5c. Prioritization

Scenario	Risk	Immediate action	Owner	Target
Retrieved injection triggers unauthorized tool action	Critical	Revoke tool/identity; isolate agent and source; preserve trace	Incident commander	Immediate
Cross-tenant content returned	Critical	Block retrieval path; snapshot evidence; inspect related traces	Security/ RAG owner	Immediate
Open external feed auto-promotes	Critical	Suspend refresh/promotion; review corpus and served traces	Data owner	Immediate
Secret appears in retrieved context	Critical	Revoke/rotate; block output; trace source and access	Security/ privacy	Immediate
One new source dominates unrelated queries	High	Quarantine source; compare index versions; hunt affected responses	RAG engineer	Same period
Citation does not support consequential claim	High	Restrict use case; review affected decisions; correct validation	Business owner	Risk-based
RAG component changed without regression	High	Freeze release; test complete bundle	MLOps/ assurance	Before cutover

6. Security Controls Matrix

Mappings are functional relationships, not equivalence, compliance, certification or endorsement. Audit-grade ISO mapping requires licensed standards text.

Control	Requirement	GAISSF™	UAIF™	AI-IRF™	Evidence	External alignment
RAG-01 System registry	Record owner, purpose, boundary, sources,	Governance/	Incident context	Preparation	Reconciled register	NIST AI RMF GOVERN/MAP;

	tenants, components, decisions and tools.	accountability				ISO 42001
RAG-02 Source admission	Authenticate origin; approve ownership, licence, classification, trust, retention and refresh.	Data/ source governance	Causality/ source	Identification	Source manifest	NIST AI 600-1; OWASP LLM03/04
RAG-03 Isolated acquisition/parsing	Sandbox connectors/parsers; minimize privileges, secrets and egress; scan active content.	Platform control	Attack-path evidence	Containment	Runtime and parser tests	NIST CSF PR.PS; OWASP LLM03
RAG-04 Pre-embedding minimization	Detect/reject/redact credentials, prohibited data and unnecessary personal information.	Minimization control	Exposure evidence	Eradication	Scan/redaction samples	NIST AI 600-1; OWASP LLM02/04
RAG-05 Staged promotion	Separate acquisition, transformation, approval and production publication.	Change/ integrity control	Affected actor	Containment	Workflow/IAM test	NIST CSF PR.AA/PR.PS; OWASP LLM04
RAG-06 End-to-end provenance	Trace source, transformations, chunks, embeddings, index, retrieval and response.	Evidence expectation	Evidence/ causality	All phases	Reconstructed sample	NIST AI 600-1; EU AI Act context where applicable
RAG-07 Query-time authorization	Resolve current identity/purpose and enforce immutable server-side pre-filters.	Access control	Exposure scope	Containment	Negative tests	OWASP LLM08; ISO 27001 access control
RAG-08 Tenant isolation	Use and test proportionate logical/physical separation for all retrieval branches.	Segregation control	Affected tenants	Containment	Cross-tenant tests	OWASP LLM08; NIST CSF PR.DS
RAG-09 Retrieval minimization	Bound top-k, chunks/source, fields, query expansion, context and score/vector exposure.	Minimization/ resilience	Impact	Containment	Config/abuse tests	OWASP LLM02/08/10
RAG-10 Poisoning detection	Validate metadata, deduplicate, detect rank dominance/source drift and adversarial content.	Integrity control	Classification/ causality	Identification	Alerts and tests	NIST AML; OWASP LLM04/08
RAG-11 Context instruction isolation	Label source boundaries and prohibit retrieved content from changing policy or authority.	Behaviour/ control objective	Attack-path evidence	Containment	Adaptive injection tests	OWASP LLM01/06; MITRE ATLAS
RAG-12 Tool/action authorization	Authorize every consequential action outside the model using scoped identity and human/rule gate.	Decision-rights control	Impact/ action evidence	Containment	Action trace	OWASP LLM06; NIST AI 600-1
RAG-13 Claim-source validation	Measure whether cited content supports claims; distinguish support from source truth.	Quality/ assurance	Harm evidence	Recovery	Faithfulness tests	NIST AI RMF MEASURE; OWASP LLM09
RAG-14 Conflict/freshness handling	Detect contradictory, expired, superseded and low-authority sources.	Data-quality control	Causality	Eradication/ recovery	Conflict/staleness tests	NIST AI 600-1
RAG-15 Version-pinned bundle	Bind source/index, parser, chunker, embedder, filters, reranker, prompt and model.	Approved boundary	Component identity	Recovery	Signed manifest	NIST AI RMF MEASURE/MANAGE

RAG-16 Retrieval/decision trace	Join identity, policy, candidates, context, response, review, tools and action.	Evidence expectation	Incident record	Identification	Export/ reconstruction	NIST CSF DE/RS; ISO logging
RAG-17 Lifecycle deletion	Propagate deletion/reclassification to chunks, indexes, caches, replicas, evaluations and backups.	Lifecycle/ privacy	Impact/ closure	Eradication	Deletion proof	NIST CSF PR.DS; privacy law as applicable
RAG-18 Kill paths	Independently suspend connectors, ingestion, index alias, retrieval, model and tools.	Resilience/ authority	Containment evidence	Containment	Exercise	NIST CSF RS; OWASP LLM01/06
RAG-19 Security regression	Test authorization, poisoning, injection, leakage, citations, actions and rollback on exact bundle.	Assurance	Evidence	Recovery/ lessons	Raw results	NIST AI RMF MEASURE/MANAGE
RAG-20 Supplier/change evidence	Require notice, versions, logs, preservation, deletion and incident cooperation.	Supplier governance	External evidence	Joint response	Contract/exercise	NIST CSF GV.SC; ISO supplier controls
RAG-21 Canonicalization and hidden-content handling	Normalize Unicode/markup and inspect or remove hidden HTML/CSS, alternate text, accessibility fields, comments and control characters before promotion.	Content-integrity control	Source/ attack evidence	Identification/ eradication	Canonicalization diff and adversarial tests	OWASP LLM01/04; NIST CSF PR.DS
RAG-22 Metadata-aware chunking	Preserve document boundaries, access classification, safety warnings, effective dates and provenance in every derived chunk; test boundary/overlap effects.	Data-quality/ control objective	Causality/ context evidence	Recovery	Chunk manifest and regression cases	NIST AI 600-1; OWASP LLM08
RAG-23 Safe rendering and outbound egress	Sanitize HTML/Markdown and active links/media; prevent output rendering from making unapproved external requests or encoding sensitive data in destinations.	Output/ data-loss control	Impact and destination evidence	Containment	Renderer policy, egress and exfiltration tests	OWASP LLM02/05/06; NIST CSF PR.DS
RAG-24 Trust-domain separation	Prevent or explicitly gate contexts that combine attacker-influenceable public/external content, sensitive private sources and external communication/action capability.	Segregation/decision control	Source and impact context	Preparation/ containment	Trust-domain matrix and cross-domain tests	NIST AI 600-1; OWASP LLM01/02/06

7. Detection & Monitoring

7a. Logging requirements

Source	Minimum fields	Purpose	Priority

Connector/ source	source/version, owner, actor, acquisition, trust, content hash	Source drift and attribution	P1
Ingestion	parser/chunker/embedder, secret/malware results, metadata derivation, approval, write set	Poisoning/provenance	P1
Retrieval gateway	principal, purpose, filter/policy hash, index, query reference, candidates, returned refs	Authorization and reproduction	P1
Context assembly	ordered chunks, labels, trust/freshness, injection signals, truncation	Context integrity	P1
Model/tool	model/prompt policy, output, citations, authorization, tool/action/result	Impact reconstruction	P1
Control plane	grants, alias/index change, deletion, export, audit and kill-switch event	Privilege and impairment	P1

Raw source text, prompts and outputs may contain regulated data, secrets or attacker-controlled active content. Prefer protected references, hashes, classifications and selectively encrypted evidence stores over unrestricted replication into the SIEM. Define role-based access, minimization, retention and legal-hold rules for raw-content capture.

Priority labels are logging priorities, not UAIF™ severity tiers.

7b. Detection logic

```
Rule: Retrieved Instruction-to-Action Correlation
Source: context + model/tool trace
Logic: retrieved_instruction_signal = true AND
       tool_sequence deviates_from approved_workflow
Threshold: one privileged or external action
Action: block action, preserve trace, restrict tool identity
```

```
Rule: Source Rank Dominance
Source: retrieval telemetry
Logic: same_source_share(top_k) > approved_limit across diverse queries
       OR new_source_rank_shift > calibrated_baseline
Threshold: use-case specific
Tuning: account for approved authoritative-source changes
```

```
Rule: Authorization Boundary Mismatch
Source: policy gateway + candidates/results
Logic: any(result.tenant not_in authorized_tenants OR
           result.classification not_in allowed_classes)
Threshold: one result
Action: fail closed and open incident
```

```
Rule: Citation Support Failure
Source: response + cited chunks + evaluator
Logic: consequential_claim = true AND support_result IN (unsupported, contradictory)
Threshold: one material claim or calibrated aggregate
Action: withhold/escalate; do not label as attack without causal evidence
```

```
Rule: Orphaned Knowledge
Source: source registry + index/cache inventory
Logic: source_state IN (deleted, expired, superseded, access_revoked)
       AND retrievable = true
Threshold: one sensitive or consequential item
```

```
Rule: Output-Mediated Exfiltration Candidate
Source: output validator + renderer/egress proxy
Logic: output_contains_external_media_or_link = true AND
       destination not_in approved_destinations AND
       payload_or_query_contains_sensitive_signal = true
Threshold: one attempted request
Action: block rendering/request, preserve protected trace, investigate source context
```

7c. Indicators

Indicator	Confidence	Interpretation
Retrieved tenant/classification differs from policy	High	Authorization incident
New source dominates unrelated queries	Medium–high	Poisoning or ranking anomaly candidate
Retrieved content contains tool-oriented hidden instructions	Medium–high	Indirect-injection candidate; context matters
Connector acquires from new domain or changed repository	Medium	Source drift/supply-chain event
Model response cites source that lacks supporting text	High for quality gap	Faithfulness failure; not automatically malicious
Index alias changes outside signed release	High	Unauthorized/unassured cutover
Deleted source remains retrievable	High	Lifecycle-control failure
Agent initiates new tool sequence after retrieval	Medium–high	Hijacking or workflow-drift candidate
Retrieved item contains hidden/off-screen markup, alternate-field instruction or abnormal control characters	Medium–high	Hidden-content injection candidate; validate source and canonicalization path
Model output attempts unapproved external image/link request containing encoded or sensitive-looking data	High	Output-mediated exfiltration candidate
One response combines public/untrusted source influence, private sensitive chunks and an external link/tool destination	High	Cross-trust-domain exfiltration chain candidate

7d. Response health metrics

Metric	Definition	Decision use
Unauthorized retrieval rate	Results outside mandatory policy boundary	Immediate risk and release gate
Provenance completeness	Returned chunks with complete lineage	Evidence readiness
Source concentration	Share of top-k from same source/actor/domain	Poisoning/diversity signal
Faithfulness failure rate	Material claims unsupported by cited context	Use-case suitability
Deletion verification time	Request to confirmed non-retrievability	Privacy/lifecycle risk
Time to ingestion containment	Trigger to verified source/refresh stop	Response readiness
Action trace completeness	Consequential actions joined to context and authorization	Accountability

8. Incident Response and Escalation

Phase	Actions	Owner	Evidence	Decision point
Preparation	Inventory RAG boundaries; test kill paths, traces, snapshots, rollback, source and tool controls.	Governance/ IR/engineering	Register and exercises	Can each plane be contained independently?

Identification	Create incident ID; preserve source, ingestion state, bundle, retrieval/context/model/tool trace; classify provisionally.	IR/SOC	UAIF™-structured record	Disclosure, poisoning, safety or action escalation?
Containment	Protect people; revoke tools/identities; suspend affected ingestion/retrieval; quarantine source/index without deleting evidence.	Incident commander	Before/after state	Scoped restriction or full shutdown?
Eradication	Remove compromised source, connector, credential, filter, parser or policy; rebuild clean bundle; hunt portfolio.	Security/ data/MLOps	Remediation and hunt	Root condition removed from all copies?
Recovery	Test authorization, poisoning, injection, leakage, faithfulness, actions and affected historical decisions; stage restoration.	Independent tester/owner	Raw tests and approvals	Exact bundle meets criteria?
Lessons	Finalize causality/impact; update GAISSF™ controls; complete UAIF™ record; update AI-IRF™ exercises and retest.	Governance/ assurance	Corrective action	Close, extend or reopen?

RAG-specific evidence package

15. Source bytes/protected reference, hash, owner, acquisition and change history.
16. Connector, parser, OCR, chunker, embedder and metadata-derivation versions.
17. Corpus/index snapshot, alias, filters, reranker and retrieval configuration.
18. Requester identity, purpose, policy decision, candidates and returned chunks.
19. Context order, boundary labels, truncation and injection signals.
20. Model/prompt version, output, citations, human review, tools and actions.
21. Deletion, cache, replica, backup and supplier evidence.
22. Evidence manifest, access restriction, hashes and custody.

Do not delete a poisoned source or rebuild the index before preserving the affected version and served traces. If harmful action or disclosure is continuing, protect people and stop the unsafe capability first.

9. Audit & Assurance Checklist

9a. Documentation

Artefact	Required content	Owner	Review
RAG system register	Boundary, owner, sources, tenants, decisions, tools, region	AI governance	Quarterly/ change
Source-admission policy	Trust, licence, classification, promotion, refresh, revocation	Data owner	Annual/change
Authorization design	Identity, entitlement, pre-filter, failure behaviour	IAM/app security	Semiannual
RAG bundle manifest	All versioned components and approvals	MLOps/RAG owner	Every release
Evaluation plan	Relevance, authorization, poisoning, injection, leakage, faithfulness, action	Assurance	Every release
Incident runbook	Kill paths, evidence, clean rebuild, affected-decision review	IR lead	Exercise/ incident

9b. Technical evidence

Area	Test	Expected evidence
Source integrity	Change, impersonate or revoke controlled source	Detection, quarantine and trace
Parser isolation	Process crafted file with restricted egress/identity	No boundary escape; complete log
Authorization	Cross-tenant, malformed-filter and fail-closed tests	Denial before candidate return
Poisoning	Insert controlled targeted/duplicate content in staging	Detection, bounded rank and no unsafe action
Indirect injection	Adaptive retrieved instructions across formats	Policy/tool boundaries hold
Citation faithfulness	Test material claims against cited context	Supported/unsupported evidence retained
Deletion	Revoke/delete source and search active copies	Verified non-retrievability
Recovery	Restore previous exact bundle	Signed manifest and full regression pass

9c. Assurance interview questions

23. Who can cause a source to reach production, and who independently approves it?
24. Show that a model cannot remove the tenant filter or grant itself a tool.
25. What happens if authorization, provenance or source-trust service is unavailable?
26. Trace one material answer from claim to source, transformations, policy and model.
27. How do you detect one source dominating unrelated queries?
28. Demonstrate suspension of an external connector without stopping evidence collection.
29. Show an adaptive indirect-injection test that reached the tool boundary.
30. Where does deleted or superseded content remain and for how long?
31. Which retrieval changes require independent regression and risk approval?
32. Show how the last RAG incident/exercise changed a GAISSF™ control and was retested.

9d. Maturity

Level	Observable state
1 — Ad hoc	Auto-ingestion, weak provenance, client filters, no RAG-specific tests
2 — Defined	Owners/policies documented; controls inconsistently enforced
3 — Operational	Staged sources, server authorization, traces, kill paths and tests operate
4 — Measured	Poisoning, faithfulness, provenance, deletion and response metrics drive gates
5 — Adaptive	Incidents and threat research change controls across the RAG portfolio

10. Research Insight ★

Finding — knowledge retrieval is an adversarial surface: The peer-reviewed PoisonedRAG study showed that targeted malicious texts could influence selected answers under the paper's evaluated black-box and white-box conditions, including large knowledge corpora. **Evidence: T1 within the experimental scope.**

Technical significance: Relevance optimization can be exploited. A document can be crafted not merely to contain false information, but to be retrieved for a target question. Defenses must therefore cover corpus admission, ranking anomalies, source concentration, context handling and downstream authorization.

Operational lesson: Build a controlled poisoning test around your own sources, retriever, top-k, reranker and model. Measure whether a small number of new documents can dominate target queries, whether investigators can identify served traces, and whether tools remain safe even when malicious content is

retrieved.

Notably Absent: The study does not establish that the reported attack success rate transfers to every RAG design, corpus, model or production environment, or that any one mitigation eliminates poisoning risk.

11. Common Mistakes

Mistake	Corrective approach
Treating RAG as a hallucination cure	Measure faithfulness and source truth separately; keep uncertainty and review.
Securing only the vector database	Include connectors, parsers, filters, context, models, tools and logs.
Auto-publishing crawled content	Stage, scan, classify, approve and version every promotion path.
Trusting retrieved content as instruction	Label it as untrusted data; authorize tools independently.
Letting the model create ACL filters	Build mandatory policy in a trusted gateway and fail closed.
Applying ACLs after retrieval	Enforce pre-filtering before candidates leave the data boundary.
Using similarity as authority or truth	Evaluate source authority, freshness, contradiction and support.
Logging only the answer	Preserve candidates, returned context, policy and downstream action.
Patching a malicious phrase	Correct the source, ingestion, authorization and action boundary.
Reindexing before evidence capture	Preserve affected corpus/index and served traces first where safe.
Deleting only the original source	Verify chunks, indexes, caches, replicas, evaluation data and backups.
Testing relevance but not abuse	Add poisoning, injection, disclosure, authorization and recovery gates.
Treating a document hash as a trust decision	Hashing proves identity or later change relative to a captured object; it does not prove that the captured content is authorized, truthful or non-malicious. Combine hashes with source authentication, admission review and provenance.

12. Quick Reference

12.1 Implementation checklist

#	Check	Yes	No	Partial
1	Every RAG system has an owner, boundary, purpose and prohibited-use record.	☐	☐	☐
2	Sources have verified origin, ownership, classification, licence and retention.	☐	☐	☐
3	Connectors and parsers are isolated with minimal identity, secrets and egress.	☐	☐	☐
4	Credentials and unnecessary sensitive data are removed before embedding.	☐	☐	☐
5	Acquisition, transformation, approval and production promotion are separated.	☐	☐	☐
6	Source-to-response provenance is complete and testable.	☐	☐	☐
7	Authorization is server-side, pre-retrieval and fail-closed.	☐	☐	☐
8	Vector, keyword, graph and hybrid branches enforce identical policy.	☐	☐	☐
9	Top-k, context, recursion, fields and score/vector exposure are bounded.	☐	☐	☐
10	Duplicate/rank dominance, source drift and poisoning are monitored.	☐	☐	☐
11	Retrieved text cannot change policy, identity, permissions or tool authority.	☐	☐	☐
12	Material claims and citations are evaluated for support and contradiction.	☐	☐	☐
13	The complete RAG bundle is versioned, regression-tested and recoverable.	☐	☐	☐
14	Deletion/reclassification reaches indexes, caches, replicas and backups.	☐	☐	☐

15	Connector, ingestion, retrieval, model and tool kill paths are exercised.	☐	☐	☐
16	Hidden markup, alternate fields and Unicode/control-character payloads are canonicalized and tested.	☐	☐	☐
17	Chunking preserves access labels, warnings, dates, boundaries and provenance.	☐	☐	☐
18	Output rendering blocks unapproved active content and outbound exfiltration channels.	☐	☐	☐
19	Trust-domain policy prevents untrusted public content, sensitive private retrieval and external communication from combining without an explicit gate.	☐	☐	☐
20	RAG incidents feed UAIF™ records, AI-IRF™ response and GAISSF™ improvement.	☐	☐	☐

12.2 Threat-to-control

Threat	Primary controls
Indirect prompt injection	RAG-03, RAG-10, RAG-11, RAG-12, RAG-18, RAG-19
Knowledge poisoning	RAG-02, RAG-05, RAG-06, RAG-10, RAG-19
Cross-tenant disclosure	RAG-07, RAG-08, RAG-09, RAG-19
Sensitive-context leakage	RAG-04, RAG-07, RAG-09, RAG-16, RAG-17
Citation laundering	RAG-06, RAG-13, RAG-14, RAG-16
Unsafe agent action	RAG-11, RAG-12, RAG-16, RAG-18
Stale/orphaned knowledge	RAG-06, RAG-14, RAG-17
Supplier/change drift	RAG-15, RAG-19, RAG-20
Hidden-content injection	RAG-03, RAG-05, RAG-10, RAG-21
Chunk-boundary safety/context loss	RAG-06, RAG-14, RAG-15, RAG-22
Output-mediated exfiltration	RAG-09, RAG-12, RAG-16, RAG-18, RAG-23
Cross-trust-domain context accumulation	RAG-02, RAG-07, RAG-09, RAG-12, RAG-23, RAG-24

12.3 Release gates

Gate	Required outcome
Scope	Owner, sources, data, tenants, decisions and tools approved
Source	Admission, promotion, change and revocation work as designed
Authorization	All negative and fail-closed cases deny before candidate return
Poisoning	Controlled attacks detected/contained within approved tolerance
Injection/action	Adaptive retrieved instructions cannot obtain unauthorized capability
Faithfulness	Material claim-support threshold and human fallback met
Lifecycle	Provenance, deletion, snapshot and rollback verified
Residual risk	Authorized, bounded and time-limited exception only

13. Assessment and Certification Readiness

This section addresses readiness only. It does not state that ODA3 certification, accreditation, licensed assessment delivery or regulatory recognition is operational.

Framework / standard	Assessor focus	Common weakness	Evidence	Mature indicator

ODA3 suite	GAISSF™ controls connect to UAIF™ evidence, AI-IRF™ response and verified improvement	Framework names disconnected from operating evidence	Control-to-incident-to-retest trace	One reproducible closed loop
NIST AI RMF / AI 600-1	Governed data/value chain, measurement, provenance, change and risk treatment	RAG added without reassessment	Register, evaluation, lineage, treatment	RAG changes trigger risk gates
NIST CSF 2.0	Assets, identities, data, detection, response and recovery cover RAG	AI pipeline outside cyber scope	Profile, telemetry, incident and restore	Unified cyber/AI control operation
ISO/IEC 42001:2023	AIMS scope, data governance, operation, evaluation and improvement	Corpus/connectors outside management system	Scope, risks, controls, monitoring, corrective action	RAG evidence supports review
ISO/IEC 27001:2022	Access, secure development, logging, supplier, backup and incident management	Generic application controls without retrieval tests	IAM, SDLC, logs, supplier and exercises	End-to-end negative testing
OWASP LLM Top 10 2025	Injection, disclosure, poisoning, agency, vector and consumption threats	Category labels without abuse cases	Threat model and adaptive tests	Controls hold through action boundary
MITRE ATLAS	Threat-informed hypotheses and observable evidence	Technique label treated as attribution	Observation-to-technique rationale	Threat mappings improve testing
EU AI Act	Applicable role/classification, data governance, logs, robustness, monitoring and incidents	Generic article table without applicability	Counsel-approved scope and evidence	Duties embedded where applicable

Examiner traps

33. **Grounding theatre:** citations appear, but claims are unsupported or sources are not authoritative.
34. **Authorization theatre:** the prompt says “only use allowed data,” but candidates are retrieved globally.
35. **Poisoning theatre:** malware scanning exists, but semantic/rank manipulation is never tested.
36. **Tool-safety theatre:** injection detector is present, but the agent retains broad credentials and irreversible tools.
37. **Deletion theatre:** source disappears from UI while chunks and caches remain retrievable.
38. **Recovery theatre:** one prompt is patched without rebuilding and testing the complete affected bundle.

Tier 1 versus Tier 2 assessment

Dimension	Tier 1 — documentation	Tier 2 — operating/technical test
Boundary	Inspect register and architecture	Reconcile runtime components and sources
Source control	Review admission/promotion	Introduce controlled source change/poison
Authorization	Review policy	Execute cross-tenant and fail-closed tests
Context safety	Review prompt/tool design	Run adaptive indirect injection through action boundary
Provenance	Inspect schema	Trace live response to source and transformations
Deletion	Review procedure	Delete/reclassify and search all copies
Recovery	Review criteria	Restore exact bundle and rerun security suite
Improvement	Inspect register	Trace finding to GAISSF™ control and retest

Preparation timeline

Period	Objective
Weeks 1–2	Inventory RAG systems, owners, sources, tenants, tools and evidence gaps.
Weeks 3–5	Correct source admission, parser isolation, authorization, minimization and traces.
Weeks 6–8	Implement provenance, deletion, bundle manifests, kill paths and supplier procedures.
Weeks 9–11	Build poisoning, injection, leakage, faithfulness, action and recovery tests.
Weeks 12–14	Exercise incident response, remediate, retest and complete readiness review.

The timeline is illustrative and depends on portfolio size, corpus sensitivity, agent authority, supplier constraints and existing platform maturity.

Primary References

- [NIST AI 100-2e2025 — Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations](https://csrc.nist.gov/pubs/ai/100/2/e2025/final)
- [NIST AI 600-1 — Generative Artificial Intelligence Profile](https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf)
- [NIST AI Risk Management Framework 1.0](https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf)
- [NIST CAISI — Strengthening AI Agent Hijacking Evaluations](https://www.nist.gov/news-events/news/2025/01/technical-blog-strengthening-ai-agent-hijacking-evaluations)
- [NIST CAISI — Insights into AI Agent Security from a Large-Scale Red-Teaming Competition](https://www.nist.gov/blogs/caisi-research-blog/insights-ai-agent-security-large-scale-red-teaming-competition)
- [OWASP LLM01:2025 — Prompt Injection](https://genai.owasp.org/llmrisk/llm01-prompt-injection/)
- [OWASP LLM04:2025 — Data and Model Poisoning](https://genai.owasp.org/llmrisk/llm04-model-denial-of-service/)
- [OWASP LLM06:2025 — Excessive Agency](https://genai.owasp.org/llmrisk/llm06-sensitive-information-disclosure/)
- [OWASP LLM08:2025 — Vector and Embedding Weaknesses](https://genai.owasp.org/llmrisk/llm082025-vector-and-embedding-weaknesses/)
- [PoisonedRAG — USENIX Security 2025](https://www.usenix.org/conference/usenixsecurity25/presentation/zou-poisonedrag)
- [Microsoft Security Response Center — CVE-2025-32711](https://msrc.microsoft.com/update-guide/vulnerability/CVE-2025-32711)
- [NVD — CVE-2025-32711: AI Command Injection in Microsoft 365 Copilot](https://nvd.nist.gov/vuln/detail/CVE-2025-32711)
- [Aim Security — EchoLeak Research Disclosure](https://www.aim.security/lp/aim-labs-echoleak-m365)
- [PromptArmor — Data Exfiltration from Slack AI via Indirect Prompt Injection](https://www.promptarmor.com/resources/data-exfiltration-from-slack-ai-via-indirect-prompt-injection)
- [MITRE ATLAS — AML.CS0035: Data Exfiltration from Slack AI](https://atlas.mitre.org/studies/AML.CS0035)
- [MITRE ATLAS](https://atlas.mitre.org/)
- [NIST Cybersecurity Framework 2.0](https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf)
- [Regulation (EU) 2024/1689 — Artificial Intelligence Act](https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng)


```
Doc ID: ODA3-2026-07-CHT-SEC-013 (pending confirmation against file listing)
Topic: Retrieval-Augmented Generation (RAG) Security
Version: 1.2
Classification: Public
```

```
Intended Audience: CISOs, AI Security Architects, Application Security Teams,
RAG/Data/MLOps Engineers, AI Governance Leads, Privacy and Legal Teams, Auditors
Evidence Confidence: T1 for primary NIST, OWASP, regulatory and peer-reviewed sources;
T2–T3 for operational synthesis and deployment-specific recommendations
Review Cycle: Quarterly recommended
Framework Cross-References: GAISSF™ v1.0; UAIF™ v1.0; AI-IRF™ v1.0;
NIST AI RMF 1.0; NIST AI 600-1; NIST AI 100-2e2025; NIST CSF 2.0;
ISO/IEC 42001:2023; ISO/IEC 27001:2022; OWASP LLM Top 10 2025;
MITRE ATLAS; EU AI Act where applicable
Known Evidence Gaps: Exact ODA3 internal identifiers were not verified from supplied
authoritative framework documents. Experimental attack results are bounded to their
evaluated systems and do not establish production prevalence. Product-specific controls,
legal applicability and ISO clause mappings require deployment evidence and licensed text.
Related Publications: CHT-SEC-009 (AI Security Evidence Collection Guide);
CHT-SEC-010 (AI Security Governance Model); CHT-SEC-011 (AI Security Incident Response
Playbook); CHT-SEC-012 (Vector Database Security Checklist)
```