PENAXTRA SECURITY RESEARCH

Operationalizing the NIST AI RMF for LLM Security Assessments

A practical implementation model for applying GOVERN, MAP, MEASURE, and MANAGE outcomes to LLM applications, RAG systems, MCP integrations, and AI agents.

VERSION	**1.0**
PUBLICATION TYPE	**Implementation Model**
FRAMEWORK	**NIST AI Risk Management Framework**
STATUS	**Published**
PUBLICATION DATE	**2026-07-14**
RESOURCE	**https://penaxtra.com/research/operationalizing-ai-rmf-for-llm-security**

Abstract

This resource converts the NIST AI Risk Management Framework into a repeatable security assessment process for LLM systems. It defines an implementation model, the LLM Security Assessment Operating Model, that moves from system context to assessment evidence, security measurements, and risk treatment decisions using the AI RMF GOVERN, MAP, MEASURE, and MANAGE functions. The model treats an LLM feature as a set of components: model endpoints, prompt and model gateways, RAG pipelines and vector stores, MCP servers and tools, agent runtimes, external providers, service identities, and the downstream systems reached through tools. It includes a governance decision model, a component and trust boundary mapping method with a relationship model for documenting attack paths, twelve evidence classes, a catalog of measurements across inventory, exposure, control, runtime, and change classes, six risk treatment decisions, an event-driven reassessment model, and a worked example of an indirect prompt injection reaching a repository write through a RAG-to-agent path. It is a community resource, not an official NIST profile or a NIST-validated use case.

Keywords: NIST AI RMF; AI risk management; LLM security assessment; AI agent security; MCP security; RAG security; assessment evidence; AI system mapping; LLM risk measurement; security assessment

Contents

1 Overview

The NIST AI Risk Management Framework describes what to govern, map, measure, and manage, but it does not prescribe how a security team assesses an LLM feature that is assembled from a model endpoint, a retrieval pipeline, one or more MCP servers, and an agent that can act on downstream systems. This resource fills that gap with an implementation model. It converts the AI RMF functions into a repeatable security assessment process for LLM systems.

The model is called the LLM Security Assessment Operating Model. It moves from system context to assessment evidence, security measurements, and risk treatment decisions. Each stage produces records that the next stage consumes, and each measurement carries an interpretation boundary so that a count is read as a count and a test result is read as behavior under a defined method.

An LLM feature is not a single assessment object. A single internal assistant may involve a user identity, a web application, an application API, a prompt gateway, a model provider, a retrieval service, a vector database, an agent runtime, an MCP server, and a service account that reaches a source control system. Each of those components exposes different evidence and a different control boundary, and the security impact of a weakness often depends on the complete path rather than a single component.

> This resource is one defensible implementation model. It is not the only way to operationalize the AI RMF, and it is not an official NIST profile or a NIST-validated use case.

2 Purpose

The problem this work addresses is not a shortage of AI risk terminology. It is the conversion of a broad risk management framework into a security assessment process that a team can run repeatably against systems composed of model endpoints, prompt and model gateways, RAG pipelines, vector stores, MCP servers and tools, agent runtimes, external providers, service identities, and the downstream systems reached through tools.

The operating model defines how an assessment team moves from system context to evidence, measurements, and treatment decisions using the AI RMF functions. It is intended for use by security architects, AI security engineering teams, AI platform owners, technical risk teams, and red and purple teams working on AI system assessment. It is usable without any specific product.

- Establish decision authority and the approvals that already apply to a system.
- Map the components, trust boundaries, effective identities, and attack paths.
- Collect the evidence that establishes the state of each component.
- Apply security measurements and record results with their interpretation boundaries.
- Convert findings into treatment decisions with an accountable owner.
- Reassess when component, identity, or capability changes invalidate prior evidence.

3 Scope

This model applies to the systems that make up an LLM or generative AI deployment and the components they depend on:

- LLM applications, internal assistants, and copilots
- Hosted model APIs and self-hosted models
- Model endpoints, model gateways, and prompt gateways
- RAG pipelines, retrieval services, and vector stores
- MCP servers, MCP clients, and MCP tools
- AI agents and agent runtimes
- External model providers and downstream services reached through tools
- Service identities under which tools and agents operate

The following are outside scope. The model is a security assessment method, not a general AI evaluation method.

- General AI safety evaluation
- Bias or fairness measurement methodology
- Model quality or accuracy benchmarking
- Legal interpretation of any obligation
- Certification against any standard

4 Use Case Context

The use case is an Enterprise LLM Security Assessment Workflow. The environment is a generalized enterprise that runs several kinds of AI systems at once: internal AI assistants, SaaS AI functionality, hosted model APIs, self-hosted models, RAG applications with enterprise search integrations, MCP servers and MCP-connected tools, AI agents, cloud AI services, and external model providers. Through tools, those systems can reach source control, ticketing, internal APIs, messaging, and cloud administration.

The assessment team does not treat this estate as one system. It selects a system boundary, records the components inside it, and assesses the paths that connect untrusted input to a consequential operation. The workflow is the same for a single assistant and for a fleet of agents; the difference is the number of components and paths in scope.

> This use case describes a generalized enterprise assessment environment. It does not represent a specific Penaxtra customer or claim deployment within a named organization.

5 Assessment Model

The LLM Security Assessment Operating Model is a sequence of stages, each of which produces evidence the next stage uses. The order below is the model's implementation order for a technical security assessment. It is not a claim about the AI RMF, and it is not presented as a mandatory sequence. The AI RMF Playbook is a companion resource of suggested actions, not a sequential checklist.

In practice the stages iterate. Mapping a new component during evidence collection sends the team back to context; a measurement result can change the treatment of an earlier finding. The value of the model is that every stage names the record it produces and the record it consumes, so an assessment can be repeated and its state can be reviewed.

#	Stage	Function	Detail
1	System context	`MAP`	Record the systems in scope and the reason each is being assessed. An LLM feature is treated as a set of related components rather than a single object.
2	Governance and decision authority	`GOVERN`	Record the owners, the assessment requirement, the risk acceptance authority, and the approvals that already apply to the system: model providers, MCP servers, and agent capabilities.
3	Component and trust boundary mapping	`MAP`	Enumerate the components, the trust boundaries between them, the effective identities that operations run as, and the paths that connect untrusted input to a consequential operation.
4	Assessment evidence collection	`MAP`, `MEASURE`	Collect the records that establish the state of each component: inventory, configuration, identity, network, data flow, dependency, and logging evidence.
5	Security measurement	`MEASURE`	Apply the inventory, exposure, control, runtime, and change measurements to the mapped components and record results with their interpretation boundaries.
6	Risk analysis	`MEASURE`, `MANAGE`	Combine measurement results with system context and impact to determine which conditions represent findings that require a decision.
7	Treatment decision	`MANAGE`	For each finding, an accountable authority selects a treatment: accept, restrict, isolate, remove, monitor, or reassess, with the rationale recorded.
8	Change monitoring	`MANAGE`	Watch for the component, identity, capability, and configuration changes that invalidate prior evidence and would change a prior decision.
9	Reassessment	`MANAGE`, `MAP`	When a change trigger fires or evidence becomes stale, run the appropriate reassessment scope: full, targeted, evidence refresh, or decision review.

6 Relationship to the NIST AI RMF

The implementation model uses the AI RMF Core functions as risk management anchors. It does not redefine the functions or replace their official meaning. It describes how a security assessment activity relates to each function, and where an official subcategory identifier is linked to an activity, the relationship is stated explicitly and the identifier is used with its published wording.

For this model, GOVERN establishes decision authority, ownership, risk tolerance, assessment requirements, exception handling, and evidence responsibilities. MAP establishes system context, boundaries, component relationships, data flows, external dependencies, effective identities, tools, and attack paths. MEASURE selects and applies assessment methods and security measurements to the risks identified in context. MANAGE prioritizes findings and selects treatment based on context, evidence, risk tolerance, and system impact.

The referenced subcategories, such as Measure 2.7 for security and resilience evaluation and Manage 4.1 for post-deployment monitoring and change management, are used with the wording published in the AI RMF Core. Every referenced identifier was checked against current official material and retained only where the assessment activity genuinely relates to the outcome.

Function	Role in this model
GOVERN	A cross-cutting function that establishes and sustains a culture of risk management, informing the other three functions.
MAP	Establishes context and frames the risks related to an AI system.
MEASURE	Uses quantitative, qualitative, or mixed methods to analyze, assess, benchmark, and monitor AI risk and related impacts.
MANAGE	Allocates resources to mapped and measured risks and acts on treatment decisions.

7 GOVERN

GOVERN in this model is an operating model, not a policy document. Its output is a set of recorded decisions and named authorities that the rest of the assessment relies on. Before a component is measured, the assessment needs to know who owns the system, who can accept its residual risk, and which providers, servers, and capabilities are already approved for it.

The section answers operational questions by turning them into record fields and named authorities rather than into prose. Who identifies the system boundary, who owns the downstream service identity, who can approve a new external model provider or MCP server, and who can accept the risk of a high-impact tool capability are all questions the Governance Decision Record resolves with a named party.

Capability approval and effective permission are separate reviews. An approval can name a repository set or a provider; the effective permission is whatever the identity behind a tool can actually reach, which the MAP stage establishes independently. A system whose approval and effective permission disagree is a finding, not a governance formality.

Operating questions the record resolves

- Who identifies the system boundary for assessment.
- Who owns the AI application and its approved use.
- Who owns the service identity that tools and agents run as.
- Who can approve a new external model provider for the system.
- Who can approve a new MCP server connection.
- Who can accept the risk of a high-impact tool capability.
- Who is notified when an agent gains a new operation.
- Who decides whether a change requires reassessment, and at what scope.

Governance Decision Record

Field	Description
`system_id`	Stable identifier for the assessed system.
`system_owner`	Accountable owner of the AI use case and its outcomes.
`technical_owner`	Team or individual responsible for the system's implementation and configuration.
`security_assessment_owner`	Party responsible for performing and maintaining the security assessment.
`risk_acceptance_authority`	Role authorized to accept residual risk for this system.
`approved_use_case`	The use the system is approved to perform.
`deployment_environment`	Where the system runs: environment and hosting model.

Field	Description
`data_classification`	Highest classification of data the system processes.
`approved_model_providers`	External and internal model services approved for this system.
`approved_mcp_servers`	MCP servers approved for connection.
`approved_agent_capabilities`	Agent tools and operations approved for use.
`assessment_frequency`	Baseline cadence for reassessment independent of change triggers.
`last_assessment_date`	Date the most recent assessment completed.
`next_review_date`	Scheduled date of the next baseline review.
`evidence_owner`	Party responsible for the assessment evidence and its retention.
`exception_reference`	Identifier of any accepted exception that applies.

8 MAP

MAP is the deepest technical stage. It is a security system mapping process that produces the component inventory, the trust boundaries, the effective identities, and the attack paths the rest of the assessment measures against. It begins with discovery and ends with a set of records that describe not just what exists but how the parts connect.

Discovery reconciles several sources because no single method is complete. AI application inventory, model endpoint and gateway configuration, RAG source and indexing records, MCP client configuration, agent configuration, and the identities behind tools each describe a part of the surface. Application discovery based only on documented architecture misses integrations added during development, so configuration review is combined with observed flows where available.

The stage records effective identities deliberately. An operation's reach is set by the identity it runs as, so a tool that reads a record may run under a service identity that can also modify or delete records. The identity is where impact concentrates, which is why the identity inventory is prerequisite to the permission reviews in MEASURE.

Security impact often depends on the complete relationship path rather than a single component finding. An untrusted document that is indexed, retrieved as context, and allowed to influence an agent that invokes a tool under a broad service identity is a path, and the path is what carries the impact. The relationship model below is a method for documenting these paths so they can be measured and reviewed.

Component types

Type	Definition
`llm_application`	An application that constructs prompts and consumes model output.
`hosted_model_api`	A model served by an external provider over an API.
`self_hosted_model`	A model the organization runs on its own infrastructure.
`model_endpoint`	A network interface through which prompts are sent and responses returned.
`model_gateway`	A component that fronts one or more model endpoints.
`prompt_gateway`	A component that applies policy to prompts and responses in the request path.
`rag_pipeline`	The indexing and retrieval path that supplies retrieved context to a model.
`vector_store`	A store of embedded content and chunk metadata used for retrieval.
`rag_data_source`	A source whose content is indexed for retrieval.
`mcp_client`	A client that connects an application or agent to MCP servers.
`mcp_server`	A server that exposes tools, resources, and prompts over the Model Context Protocol.
`mcp_tool`	A tool exposed by an MCP server that performs an operation.

Type	Definition
agent_runtime	The runtime that plans and invokes tools on behalf of an agent.
agent_tool	A tool available to an agent.
external_model_provider	A supplier that processes prompts and returns model output.
service_identity	The identity a component, tool, or agent authenticates as.
downstream_service	A system reached through a tool or an agent operation.
logging_source	A source of runtime records for a component.

Relationship types

Relationship	Meaning
USES_MODEL	An application uses a model.
CALLS_ENDPOINT	A component calls a model endpoint.
SENDS_PROMPT_TO	A component sends prompt content to a model or gateway.
RECEIVES_RESPONSE_FROM	A component receives model output.
RETRIEVES_FROM	A retrieval step reads content from a store or service.
INDEXES_FROM	An indexing pipeline reads content from a source.
CONNECTS_TO_MCP	A client or agent connects to an MCP server.
EXPOSES_TOOL	An MCP server exposes a tool.
INVOKES_TOOL	An application or agent invokes a tool.
EXECUTES_AS	An operation runs under a service identity.
CALLS_SERVICE	A tool or agent calls a downstream service.
READS_FROM	An operation reads from a downstream system.
WRITES_TO	An operation writes to a downstream system.
ADMINISTERS	An operation performs an administrative action on a system.
LOGS_TO	A component sends records to a logging source.
DEPENDS_ON	A component depends on another component or supplier to function.

Example relationship path

```
RAG indexing pipeline INDEXES_FROM Untrusted document
LLM application RETRIEVES_FROM Retrieval service
LLM application SENDS_PROMPT_TO Model endpoint
LLM application RECEIVES_RESPONSE_FROM Model endpoint
```

```
Agent runtime CONNECTS_TO_MCP MCP server
MCP server EXPOSES_TOOL Repository tool
Agent runtime INVOKES_TOOL Repository tool
Repository tool EXECUTES_AS Service account
Repository tool WRITES_TO Source control system
```

MAP outputs

- AI System Inventory
- Model Endpoint Inventory
- External Model Provider Inventory
- RAG Source Inventory
- MCP Server Inventory
- MCP Tool Inventory
- Agent Capability Inventory
- Effective Identity Inventory
- Data Flow Record
- Trust Boundary Record
- External Dependency Record
- AI Attack Path Record

9 MEASURE

MEASURE applies assessment methods to the components and paths that MAP produced. It does not reduce to a statement that risks should be assessed regularly. It defines five measurement classes, each with a distinct evidence source and a distinct interpretation boundary, so that a result is read for what it establishes and not for more.

- Inventory measurements size the gap between what is discovered and what is recorded and owned.
- Exposure measurements establish the conditions under which a component can be reached or used.
- Control measurements establish how a control behaved under a defined assessment method.
- Runtime measurements establish what was observed during operation over a defined period.
- Change measurements establish which changes have invalidated prior evidence.

Results are not all pass or fail. A measurement may produce a count, a boolean, a pass or fail, a coverage percentage, an observed or not-observed state, a severity distribution, a control gap, or a contextual finding. Each result type carries its own limitation: a count sizes a gap without judging severity, a coverage percentage is bounded by what the assessment enumerated, and observed runtime activity is evidence of behavior during the observation period rather than proof that unobserved paths are safe.

10 MANAGE

MANAGE converts findings and measurement results into decisions. Each finding is reviewed against system context and impact, and an accountable authority selects a treatment. The treatments in this model are decision labels used by the assessment, not official NIST treatment categories. They are recorded in a Risk Decision Record with the evidence, the effective identity, the selected treatment, and a reassessment trigger.

A treatment is only meaningful with an owner and a condition. Acceptance is recorded against a named authority and a review date; monitoring is a treatment only when the records it depends on exist and a change condition is defined, and is otherwise deferral. Where a finding does not involve an attack path or an effective identity, those fields are marked non-applicable rather than filled with irrelevant evidence.

11 Assessment Evidence Model

Every measurement rests on evidence, and evidence has limits. The model groups evidence into twelve classes. Each class states its purpose, gives concrete examples, and names the limitation that most often causes a result to be misread. The recurring limitation is that a record establishes one thing and is often read as establishing another: a documented service identity establishes which identity is used, not what it can do; a log that records the model endpoint does not support execution review; observed runtime activity describes a period, not all possible behavior.

`Inventory Evidence`

Purpose. Establish which AI applications, endpoints, providers, RAG sources, MCP servers, tools, and agents exist and who owns them.

Examples. AI application inventory; Model endpoint inventory; External provider records; MCP server and tool inventory; Agent and tool inventory

Limitation. An inventory is a reconciliation of several discovery sources. A single source is incomplete, and an entry without an owner cannot be assessed or remediated.

`Architecture Evidence`

Purpose. Establish how components connect, where trust boundaries sit, and which paths carry untrusted input to consequential operations.

Examples. Component and data flow diagrams; Trust boundary records; Attack path records; Prompt and context boundary description

Limitation. A design diagram records the intended architecture. It should be reconciled against the deployed configuration, which can differ.

`Configuration Evidence`

Purpose. Establish the settings that determine a component's security boundary independent of its code.

Examples. Model gateway configuration; Prompt gateway policy; RAG indexing configuration; Agent policy configuration

Limitation. Configuration changes without a code change. Evidence reflects the state at collection time and can be stale by the next review.

`Identity Evidence`

Purpose. Establish the identities that components, tools, and agents authenticate as, so effective permission can be reviewed.

Examples. Service principals; Cloud roles; API identities; MCP service identities; Agent execution identities; Downstream access policies

Limitation. A documented service identity does not establish effective permission. Downstream role assignment and operation scope must also be reviewed.

Network Evidence

Purpose. Establish where endpoints and services are reachable from and where organizational data flows.

Examples. Endpoint reachability records; Network data flow records; External destination records

Limitation. Reachability observed from one vantage point does not describe reachability from every network position an attacker may hold.

Data Flow Evidence

Purpose. Establish the data that enters prompts, returns in responses, and moves through retrieval and external transfer.

Examples. Prompt data flow records; Response data flow records; Retrieval flow records; Data classification per destination

Limitation. A data flow record captures known paths. Content that reaches a model through a new integration is outside a record that predates it.

Dependency Evidence

Purpose. Establish the suppliers and components an AI system depends on to function.

Examples. Model provider records; Model artifact provenance; SDK and agent framework versions; MCP server implementation records

Limitation. A dependency list records declared dependencies. Active runtime use of a dependency is a separate question from its presence in a manifest.

Test Evidence

Purpose. Establish how a control behaved under a defined assessment method.

Examples. Prompt injection test results; Retrieval authorization test results; System prompt disclosure test results; Tool restriction test results

Limitation. A test result describes behavior under the methods used. It does not establish that untested inputs or paths behave the same way.

Runtime Evidence

Purpose. Establish what was observed during operation over a defined period.

Examples. Prompt and response security events; Tool invocation records; Agent action records; Policy decision records

Limitation. Observed activity is evidence of behavior during the observation period. It is not proof that unobserved execution paths do not exist.

Logging Evidence

Purpose. Establish which records are generated and available for review, and whether they are sufficient to reconstruct an operation.

Examples. Application log records; Model gateway logs; Agent execution logs; Audit records

Limitation. A log that records only the model endpoint is insufficient for execution review. Completeness is assessed against the fields an investigation needs.

`Change Evidence`

Purpose. Establish the component, identity, capability, and configuration changes that affect a prior assessment.

Examples. Configuration change records; Capability change records; Dependency version changes; Baseline records

Limitation. Change evidence is only as good as the change sources it monitors. A change made outside a monitored path is invisible until the next full review.

`Decision Evidence`

Purpose. Establish which findings were reviewed, what was decided, by whom, and under what conditions.

Examples. Risk decision records; Treatment action records; Exception records; Reassessment trigger records

Limitation. A decision record captures the decision at a point in time. A decision made on stale evidence carries the staleness of that evidence.

12 Measurement Catalog

The catalog lists the measurements that feed the MEASURE stage. Each entry carries a stable identifier, the security question it answers, the evidence it draws on, the method, the result type, the interpretation, its limitations, the related system components, and the AI RMF outcomes it relates to. The identifiers are grouped by class: INV for inventory, EXP for exposure, CTL for control, RUN for runtime, and CHG for change.

It favors a set of measurements that each carry distinct technical content over a larger set that repeats the same wording with substituted nouns. A team applies the measurements that fit the components in scope; not every measurement applies to every system.

Inventory measurements

INV-001 AI application inventory coverage

Security question. Which LLM and generative AI applications are known and attributed to an owner.

Evidence. AI application inventory reconciled against source dependencies, gateway configuration, and outbound provider destinations.

Method. Reconcile discovery sources and compare against the approved inventory.

Result type. Count

Interpretation. A gap between discovered and inventoried applications identifies systems that are running without a security record. It does not by itself indicate a vulnerability.

Limitations. Discovery based on documented architecture alone misses integrations added during development.

AI RMF. `Map 1.1, Map 2.1`

INV-002 Model endpoint inventory coverage

Security question. Which model endpoints are reachable from applications, agents, and internal networks.

Evidence. Model endpoint inventory, gateway routing configuration, and observed network flows.

Method. Enumerate endpoints and associate each with an application, environment, and access boundary.

Result type. Count

Interpretation. Endpoints present in traffic but absent from the inventory are unmanaged interfaces to review. The count sizes the gap, not its severity.

Limitations. Endpoints introduced during development are missed by configuration review alone; observed flows are needed to close the gap.

AI RMF. `Map 1.1, Map 2.1`

INV-003 External model provider inventory

Security question. Which external model providers process organizational prompts and data.

Evidence. Application-to-provider mapping and outbound network destinations.

Method. List providers, the applications that use them, and the data classification reaching each.

Result type. Count

Interpretation. Each provider is a supplier that processes organizational data. A provider absent from supplier management is a gap in third-party oversight.

Limitations. A provider relationship can arrive through a dependency or SDK rather than a direct integration.

AI RMF. `Map 1.1, Govern 6.1`

INV-004 Self-hosted model inventory

Security question. Which models the organization runs on its own infrastructure and where.

Evidence. Deployment records, model gateway configuration, and container image contents.

Method. Record each self-hosted model, its host, and the endpoints that serve it.

Result type. Count

Interpretation. Self-hosted models keep prompts within the organizational boundary but still expose endpoints and artifacts that require the same inventory discipline.

Limitations. A model artifact present on disk is not necessarily in active serving use.

AI RMF. `Map 1.1, Map 2.1`

INV-005 RAG system and source inventory

Security question. Which retrieval pipelines exist and which sources feed their indexes.

Evidence. RAG data source inventory, indexing pipeline configuration, and source ownership records.

Method. List retrieval systems and reconcile indexed sources against the indexing configuration.

Result type. Count

Interpretation. Indexes accumulate sources across teams over time. Sources present in the pipeline but absent from the inventory define the content a model can retrieve without a record.

Limitations. Reconciliation against documentation alone misses sources added directly to the pipeline.

AI RMF. `Map 1.1, Map 2.1`

INV-006 MCP server inventory

Security question. Which MCP servers are connected to applications and agents, and where they originate.

Evidence. MCP client configuration and MCP server records.

Method. Reconcile connected servers against client configuration and record each server's origin.

Result type. Count

Interpretation. Connected servers extend application capability to whatever they expose and reach downstream. A connected server without a record is unassessed capability.

Limitations. MCP servers can be added by developers during integration and can proxy to further servers.

AI RMF. `Map 1.1, Govern 6.1`

INV-007 MCP tool inventory

Security question. Which tools connected MCP servers expose and what operations they perform.

Evidence. Tool definitions from connected MCP servers.

Method. Enumerate tools and record the operation each performs and the server that exposes it.

Result type. Count

Interpretation. The tool set is the operation surface an application or agent can reach through MCP. An unlisted tool is an operation nobody reviewed.

Limitations. A server can register tools dynamically, so the inventory reflects the tools present at collection time.

AI RMF. `Map 2.1`

INV-008 Agent capability inventory

Security question. Which tools each agent can invoke and which operations those tools expose.

Evidence. Agent configuration and tool definitions for each deployed agent.

Method. Record the tools available to each agent and the operations reachable through them.

Result type. Count

Interpretation. An agent's operation surface is the union of its tools, not its described role. The inventory establishes what the agent can do.

Limitations. Tool sets change as agents are extended; the inventory must track the deployed configuration, not the intended design.

AI RMF. `Map 2.1`

INV-009 Effective execution identity inventory

Security question. Which identities components, tools, and agents authenticate as.

Evidence. Service principals, cloud roles, API identities, and agent execution identities.

Method. Record the identity each component operation runs as and link it to its downstream authorizations.

Result type. Count

Interpretation. The identity is where impact concentrates. An operation's reach is set by the identity it runs as, so the identity inventory is prerequisite to permission review.

Limitations. An identity record establishes which identity is used, not what it can do; downstream scope is a separate review.

AI RMF. `Map 2.1`

INV-010 Unowned AI asset count

Security question. How many AI assets in the inventory have no accountable owner.

Evidence. AI system inventory and ownership records.

Method. Count inventory entries without a recorded owner.

Result type. Count

Interpretation. A non-zero value identifies systems requiring ownership resolution. It does not establish technical vulnerability; it establishes that the system cannot yet be assessed or remediated by an accountable party.

Limitations. Ownership recorded in the inventory may be stale; a named owner is not the same as an owner who accepts the responsibility.

AI RMF. `Govern 2.1, Map 1.1`

Exposure measurements

EXP-001 Publicly reachable model endpoints

Security question. Which model endpoints are reachable from outside the intended network boundary.

Evidence. Endpoint reachability records and network configuration.

Method. Test reachability of each endpoint from its intended and adjacent network positions.

Result type. Count

Interpretation. A publicly reachable endpoint is an interface to review for access control. Reachability is a condition to assess, not a finding on its own.

Limitations. Reachability observed from one position does not describe reachability from every network position.

AI RMF. `Measure 2.7`

EXP-002 Model endpoints without expected authentication

Security question. Which reachable endpoints lack the authentication their boundary requires.

Evidence. Endpoint authentication configuration and reachability records.

Method. For each reachable endpoint, verify the presence and type of the required access control.

Result type. Count

Interpretation. An endpoint reachable without an adequate access boundary is a direct exposure. Rate limiting is not an access boundary and does not close this measurement.

Limitations. Configuration review establishes declared authentication; a live check confirms it is enforced.

AI RMF. `Measure 2.7`

EXP-003 Non-production endpoints reachable outside intent

Security question. Which development or test endpoints are reachable beyond their intended boundary.

Evidence. Endpoint inventory tagged by environment and reachability records.

Method. Identify non-production endpoints and test their reachability outside the intended scope.

Result type. Count

Interpretation. Non-production endpoints are a recurring exposure because they are often created without production controls and not removed when their use ends.

Limitations. Environment tagging must be accurate for this measurement to be meaningful.

AI RMF. `Measure 2.7`

EXP-004 External model provider data egress

Security question. Which data classifications leave the organization to external providers.

Evidence. Outbound data flow records and provider processing terms.

Method. Map the classification of data sent to each external provider and compare against approvals.

Result type. Contextual Finding

Interpretation. Data crossing the organizational boundary to a provider is governed by that provider's processing and retention terms. Whether it is acceptable depends on classification and provider terms, which are organization-specific.

Limitations. The provider agreement is part of the control; transport protection alone does not address processing terms.

AI RMF. `Govern 6.1`

EXP-005 MCP servers reachable outside approved boundary

Security question. Which MCP servers are reachable from outside the approved network boundary.

Evidence. MCP server reachability records and network configuration.

Method. Test each server's reachability against its approved network boundary.

Result type. Count

Interpretation. A server reachable beyond its approved boundary widens who can drive the operations it exposes and the downstream systems it holds credentials for.

Limitations. The credentials a server holds downstream are frequently more sensitive than the connection to it; both directions matter.

AI RMF. `Measure 2.7`

EXP-006 Tools exposing administrative operations

Security question. Which connected tools can perform administrative operations on downstream systems.

Evidence. Tool definitions and the effective permissions of the identities behind them.

Method. Review each tool's reachable operations against the impact of those operations.

Result type. Count

Interpretation. Administrative operations reachable through a tool raise the impact of any manipulation of that tool. The count identifies tools whose effective operations warrant a boundary review.

Limitations. A tool's declared description does not establish the operations reachable through its parameters and identity.

AI RMF. `Measure 2.7`

EXP-007 Agents using high-impact service identities

Security question. Which agents run under identities that can perform high-impact operations.

Evidence. Agent execution identity records and downstream authorization records.

Method. Compare each agent's execution identity permissions against the operations its function requires.

Result type. Contextual Finding

Interpretation. An agent's risk follows its effective identity. Where the identity can perform operations beyond the agent's function, the gap is the exposure. What counts as high-impact is organization-specific.

Limitations. Excess permission usually comes from reusing a broad service identity rather than an intentional grant.

AI RMF. `Measure 2.7`

EXP-008 RAG sources crossing tenant or classification boundaries

Security question. Whether retrieval can return content across tenant or classification boundaries.

Evidence. Retrieval authorization design and source classification records.

Method. Assess whether retrieval enforces the requester's authorization or runs with a broad service identity.

Result type. Observed/Not Observed

Interpretation. Oversharing appears when retrieval runs with a broad identity and authorization is applied only at the application boundary. The condition is present or absent for a given pipeline.

Limitations. Assessment requires the identity retrieval executes as, not only the application's front-door controls.

AI RMF. `Measure 2.7`

EXP-009 Vector stores reachable outside intended boundary

Security question. Which vector stores and their metadata are reachable beyond the intended access boundary.

Evidence. Vector store access configuration and network reachability records.

Method. Test store reachability and review read and write access against the intended boundary.

Result type. Count

Interpretation. Chunk metadata can carry source paths and access hints that are sensitive independent of the embeddings. Store exposure includes read, write, and network reach.

Limitations. Write access to a store also enables poisoning, so read exposure is not the only concern.

AI RMF. `Measure 2.7`

Control measurements

CTL-001 Direct prompt injection resistance

Security question. Whether the application preserves its control instructions under adversarial user input.

Evidence. Adversarial prompt test results and prompt construction design.

Method. Apply an adversarial prompt suite against the application and score deviations from intended behavior.

Result type. Pass/Fail

Interpretation. The result describes behavior under the methods used. A pass under one method set is not a guarantee against untested phrasings; a fail identifies a boundary that did not hold.

Limitations. Pattern filtering removes a subset of phrasings and does not establish the boundary.

AI RMF. `Measure 2.7, Measure 2.1`

CTL-002 Indirect prompt injection resistance

Security question. Whether instructions embedded in retrieved or tool-provided content alter application behavior.

Evidence. Adversarial content test results and retrieval handling design.

Method. Introduce adversarial content through the ingestion and retrieval path and score its influence on behavior and actions.

Result type. Pass/Fail

Interpretation. The exposure grows with the actions reachable once content is in context. A fail is most consequential where the application can take actions after retrieval.

Limitations. Coverage depends on exercising the content sources the application actually reads.

AI RMF. `Measure 2.7, Measure 2.1`

CTL-003 System prompt disclosure

Security question. Whether application-controlled instructions can be disclosed or reconstructed.

Evidence. Extraction and inference test results and prompt handling design.

Method. Attempt disclosure through direct extraction, error paths, and multi-turn inference.

Result type. Observed/Not Observed

Interpretation. Preventing literal extraction is not the same as protecting a secret the prompt contains. Where the prompt carries a credential, disclosure risk remains even when extraction is prevented.

Limitations. Behavioral inference can reconstruct security-relevant content without literal disclosure.

AI RMF. `Measure 2.7`

CTL-004 Sensitive data in prompt handling

Security question. Whether sensitive data placed into prompts is controlled for its classification and destination.

Evidence. Prompt data flow records and redaction configuration.

Method. Trace sensitive data into prompts and review controls against the destination that processes it.

Result type. Coverage Percentage

Interpretation. The applicable controls differ between a self-hosted model and an external provider. Coverage is measured against the classified data types that reach prompts.

Limitations. Coverage reflects the data types the assessment enumerated; an unmapped data path is outside the percentage.

AI RMF. `Measure 2.7`

CTL-005 Sensitive data in response handling

Security question. Whether sensitive data in responses is controlled for the recipient and destination.

Evidence. Response handling design and output monitoring records.

Method. Review response handling and any output controls against the recipients and downstream destinations.

Result type. Coverage Percentage

Interpretation. Output handling is a separate control from input handling. A response safe for one recipient may not be safe to write to a shared log or pass to another system.

Limitations. A response can carry sensitive data drawn from retrieval or tool output, not only from the model.

AI RMF. `Measure 2.7`

CTL-006 Retrieval authorization

Security question. Whether retrieval returns only content the requester is authorized to access.

Evidence. Retrieval authorization test results and access control configuration.

Method. Test retrieval as different identities and verify results respect the requester's authorization.

Result type. Pass/Fail

Interpretation. A fail means a user can obtain restricted content through generated answers. The decision point is whether authorization is applied at retrieval or only at the application boundary.

Limitations. The identity retrieval executes as determines the result more than the application's front-door controls.

AI RMF. `Measure 2.7, Measure 2.1`

CTL-007 RAG source provenance

Security question. Whether indexed content has recorded provenance and integrity controls.

Evidence. Source provenance records and index write access controls.

Method. Review provenance at the source, at insertion, and at re-indexing, and confirm write controls.

Result type. Coverage Percentage

Interpretation. Poisoning frequently exploits a source the pipeline already trusts. Coverage measures the indexed content whose provenance and integrity are established.

Limitations. Stale content that has been replaced can be as damaging as content that is malicious.

AI RMF. `Measure 2.7, Govern 6.1`

CTL-008 Untrusted retrieved content handling

Security question. Whether retrieved content is treated as untrusted input and its influence bounded.

Evidence. Retrieval handling design and action constraints after retrieval.

Method. Review whether retrieved content is placed into context as trusted and what actions are reachable afterward.

Result type. Observed/Not Observed

Interpretation. The risk is proportional to what the application can do after retrieval. The condition is whether retrieved content is bounded and observable once in context.

Limitations. Content from user-writable sources warrants the same scrutiny as direct user input.

AI RMF. `Measure 2.7`

CTL-009 MCP server authentication

Security question. Whether MCP servers and connections are authenticated and their credentials managed.

Evidence. MCP authentication configuration and downstream credential inventory.

Method. Review authentication in both directions: the client connection and the server's downstream credentials.

Result type. Pass/Fail

Interpretation. A connection established without authenticating the server, or downstream credentials that are unmanaged, are direct gaps. Both directions are in scope.

Limitations. The credentials a server holds for downstream services are frequently more sensitive than the client connection.

AI RMF. `Measure 2.7, Govern 6.1`

CTL-010 MCP effective permission review

Security question. What operations a tool can actually perform, given the identity it runs as.

Evidence. Tool definitions, service identity records, and downstream authorization records.

Method. Evaluate effective permission: the service identity, its downstream permissions, and operations reachable through parameters.

Result type. Control Gap

Interpretation. A benign tool description does not establish the execution boundary. The gap is the difference between the tool's declared behavior and its effective permission.

Limitations. Effective permission must be reviewed against downstream role assignment, not the tool description.

AI RMF. `Measure 2.7, Map 2.1`

CTL-011 Tool input manipulation constraint

Security question. Whether tool invocations are constrained against manipulation through model-generated parameters.

Evidence. Tool parameter handling design and invocation constraint configuration.

Method. Review the operations and parameter ranges a tool accepts and whether invocations are recorded.

Result type. Observed/Not Observed

Interpretation. A tool with broad parameters is manipulable even when its permissions are correct. The condition is whether the reachable operations and parameters are constrained.

Limitations. Least privilege on the identity bounds impact but does not by itself constrain which operation a manipulated call selects.

AI RMF. `Measure 2.7`

CTL-012 Agent operation restriction

Security question. Whether consequential agent operations occur within a defined boundary.

Evidence. Tool execution constraints, operation allowlists, and execution identity records.

Method. Review the boundary applied to consequential operations: approval, constrained identity, allowlist, or observable execution.

Result type. Control Gap

Interpretation. Human approval is one boundary among several and is not universally required. The gap is any consequential operation reachable without a boundary appropriate to its impact.

Limitations. Approval fits rare high-impact operations; constrained identities and allowlists bound frequent operations without making the agent impractical.

AI RMF. `Measure 2.7, Govern 3.2`

CTL-013 Effective identity least privilege

Security question. Whether the identities behind tools and agents are scoped to required operations.

Evidence. Service identity records and downstream authorization records.

Method. Compare granted permissions against required operations for each execution identity.

Result type. Control Gap

Interpretation. The gap between granted and required permission determines the impact of a compromise or manipulation. Excess permission is the finding, not the identity's existence.

Limitations. A shared identity across agents removes attribution and makes least privilege impractical.

AI RMF. `Measure 2.7`

CTL-014 Logging completeness for execution review

Security question. Whether records are sufficient to reconstruct a prompt, tool, or agent operation.

Evidence. Application, gateway, and agent execution logs.

Method. Compare recorded fields against the fields an investigation needs: invoking application, tool, effective identity, operation, target, and result.

Result type. Coverage Percentage

Interpretation. A log that records only the model endpoint is insufficient for execution review. Coverage measures the operations whose records contain the fields an investigation needs.

Limitations. Records must be attributable across sources on a common request or session identifier to support reconstruction.

AI RMF. `Measure 3.1, Manage 4.1`

Runtime measurements

RUN-001 Prompt security events

Security question. What prompt manipulation activity was observed during the observation period.

Evidence. Prompt security event records from the runtime path.

Method. Review recorded prompt security events over the defined period.

Result type. Severity Distribution

Interpretation. Observed events are evidence of runtime behavior during the period. They do not prove that unobserved injection paths do not exist.

Limitations. Monitoring observes what reaches the application; it does not establish whether the application's boundaries are correct.

AI RMF. `Measure 3.1, Measure 2.4`

RUN-002 Response security events

Security question. What sensitive-disclosure or policy-violating output was observed in responses.

Evidence. Response security event records from the runtime path.

Method. Review recorded response security events over the defined period.

Result type. Severity Distribution

Interpretation. Response monitoring complements input monitoring because some conditions are only visible in the response. Observed events describe the period, not all possible output.

Limitations. Response monitoring must avoid retaining the sensitive content it detects in a broadly readable store.

AI RMF. `Measure 3.1, Measure 2.4`

RUN-003 MCP tool invocation records

Security question. What tool invocations occurred and whether each is attributable.

Evidence. Tool invocation records including tool, parameters summary, identity, and outcome.

Method. Review invocation records for attribution and for operations outside expected use.

Result type. Observed/Not Observed

Interpretation. Attributable invocation records support detection and investigation. Their absence means tool activity cannot be reconstructed, independent of whether misuse occurred.

Limitations. Records reflect invocations that were logged; an unlogged path produces no record.

AI RMF. `Measure 3.1, Manage 4.1`

RUN-004 Agent action trace completeness

Security question. Whether agent actions can be traced from trigger to downstream result.

Evidence. Agent action records linking the invoking application, tool, identity, operation, target, and result.

Method. Trace a sample of agent actions end to end and record missing fields.

Result type. Coverage Percentage

Interpretation. A trace that identifies only the model endpoint is insufficient. Coverage measures the actions whose records support end-to-end reconstruction.

Limitations. Trace completeness depends on the agent and its tools producing attributable records.

AI RMF. `Measure 3.1, Manage 4.1`

RUN-005 Security policy decision records

Security question. What allow and block decisions the runtime path recorded.

Evidence. Policy decision records from the prompt or runtime gateway.

Method. Review policy decision records and reason codes over the period.

Result type. Count

Interpretation. Decision records show what the runtime path allowed and blocked. A block record is evidence of a decision, not evidence that the tested control set is complete.

Limitations. Decision records describe enforcement at runtime, not the correctness of the policy being enforced.

AI RMF. `Measure 3.1`

RUN-006 Downstream operation attribution

Security question. Whether downstream operations initiated through tools are attributable to the initiating request.

Evidence. Downstream service logs correlated with tool invocation records.

Method. Join downstream operations to the tool invocation and identity that initiated them.

Result type. Coverage Percentage

Interpretation. Attribution across the boundary is what allows a downstream change to be traced back to an agent action. Coverage measures the operations that can be joined.

Limitations. Correlation quality depends on consistent identifiers across the tool and downstream logs.

AI RMF. `Measure 3.1, Manage 4.1`

Change measurements

CHG-001 New or changed model endpoint

Security question. Whether a new or changed model endpoint has appeared since the last assessment.

Evidence. Endpoint inventory baseline and change records.

Method. Compare the current endpoint set against the assessed baseline.

Result type. Boolean

Interpretation. A new endpoint changes the assessed surface. The measurement flags that a targeted reassessment of the endpoint is due; it does not itself judge the endpoint.

Limitations. Detection depends on the change sources monitored; an endpoint added outside a monitored path is invisible until a full review.

AI RMF. `Manage 4.1`

CHG-002 New or changed model provider

Security question. Whether the external model providers in use have changed.

Evidence. Provider inventory baseline and outbound destination records.

Method. Compare current providers and provider configuration against the baseline.

Result type. Boolean

Interpretation. A provider change alters where organizational data is processed and under whose terms. It triggers a targeted reassessment of data egress and provider terms.

Limitations. A provider change can arrive through an SDK or dependency update rather than a configuration change.

AI RMF. `Manage 3.1, Govern 6.1`

CHG-003 New RAG data source or trust source

Security question. Whether a new source has been indexed since the last assessment.

Evidence. RAG source inventory baseline and indexing pipeline change records.

Method. Compare the current indexed source set against the baseline.

Result type. Boolean

Interpretation. A new source changes what the model can retrieve and the trust the pipeline places in it. It triggers a targeted reassessment of provenance and retrieval scope.

Limitations. Sources added directly to the pipeline are missed if only documentation is compared.

AI RMF. `Manage 4.1, Map 1.1`

CHG-004 Retrieval authorization change

Security question. Whether retrieval authorization has changed since the last assessment.

Evidence. Retrieval authorization configuration baseline and change records.

Method. Compare current retrieval authorization against the assessed baseline.

Result type. Boolean

Interpretation. A change to retrieval authorization can introduce oversharing without any code change. It triggers a targeted reassessment of the retrieval authorization control.

Limitations. Authorization applied through a shared service identity can change downstream without a visible pipeline change.

AI RMF. `Manage 4.1`

CHG-005 New MCP server or tool

Security question. Whether a new MCP server or tool has been connected since the last assessment.

Evidence. MCP inventory baseline and client configuration change records.

Method. Compare the current server and tool set against the baseline.

Result type. Boolean

Interpretation. A new server or tool extends the operation surface reachable through MCP. It triggers a targeted reassessment of authentication and effective permission.

Limitations. Servers can proxy to further servers, so a single connection can change more of the surface than it appears.

AI RMF. `Manage 4.1, Govern 6.1`

CHG-006 MCP capability change

Security question. Whether a connected server changed the tools or permissions it exposes after connection.

Evidence. MCP capability baseline and change records.

Method. Compare current exposed capability against the approved baseline.

Result type. Boolean

Interpretation. A server that adds a tool has changed the application's execution surface after review. It triggers a targeted reassessment against the approved configuration.

Limitations. Capability change is a monitoring concern as well as a change-management concern.

AI RMF. `Manage 4.1`

CHG-007 Effective identity or downstream permission change

Security question. Whether an execution identity or its downstream permissions have changed.

Evidence. Identity and downstream authorization baseline and change records.

Method. Compare current identity permissions against the assessed baseline.

Result type. Boolean

Interpretation. A permission increase on an execution identity raises the impact of any operation that runs as it. It triggers a targeted reassessment of the identity and the operations it reaches.

Limitations. Downstream permission changes are made outside the AI system and can be missed by AI-scoped monitoring.

AI RMF. `Manage 4.1`

CHG-008 Logging loss or degradation

Security question. Whether logging for the system has been removed or reduced below the assessed level.

Evidence. Logging configuration baseline and change records.

Method. Compare current logging against the assessed completeness baseline.

Result type. Boolean

Interpretation. A loss of logging reduces the evidence available for runtime review and investigation. It triggers an evidence refresh and a review of the logging completeness measurement.

Limitations. Degradation can be partial: records may still be produced but missing the fields an investigation needs.

AI RMF. `Manage 4.1`

CHG-009 Assessment evidence staleness

Security question. Whether the assessment evidence has passed its validity period.

Evidence. Evidence collection dates and validity periods.

Method. Compare evidence collection dates against the defined validity period for each evidence class.

Result type. Boolean

Interpretation. Evidence past its validity period can no longer support a current decision. It triggers an evidence refresh for the affected classes and a review of the decisions that relied on it.

Limitations. Validity periods are organization-specific and depend on how quickly the underlying component changes.

AI RMF. `Manage 4.1, Measure 3.1`

13 Risk Treatment Model

The treatment model defines six decision labels for this implementation model: accept, restrict, isolate, remove, monitor, and reassess. They describe what the assessment decided to do about a finding. They are not official NIST labels and they do not redefine enterprise risk terminology; they name the action recorded in the Risk Decision Record.

The distinction between restrict, isolate, and remove is the reach that is changed. Restrict reduces the operations, permissions, data scope, or endpoints available to a component. Isolate separates a component or path from the systems, data, identities, or networks that create unacceptable impact without removing the component. Remove takes the component out of the approved context and ends both the exposure and the function it provided.

Decision	Definition
ACCEPT	The identified condition remains within documented risk tolerance and no additional treatment is selected at the time of review. Acceptance is recorded against a named authority and a review date. It is a decision, not the absence of one.
RESTRICT	Reduce the operations, permissions, data scope, endpoints, sources, or execution capability available to the component. Restriction targets effective capability. Reducing a tool's parameters or narrowing a service identity are both restrictions.
ISOLATE	Separate the component or execution path from the systems, data, identities, or network boundaries that create unacceptable impact. Isolation changes what a compromised or manipulated path can reach, without removing the component.
REMOVE	Remove the component, dependency, integration, capability, data source, tool, or endpoint from the approved system context. Removal ends the exposure. It also ends the function the component provided, so it is a use-case decision as well as a security one.
MONITOR	Continue operation under explicitly defined evidence and change conditions. Monitoring is only a treatment when the records it depends on exist and a change condition is defined. Without those, it is deferral.
REASSESS	Perform a new or targeted assessment because system context, evidence, or capability has materially changed. Reassessment is selected when the current evidence can no longer support a decision, not as a way to postpone one.

Risk Decision Record fields

`finding_id, system_id, finding_title, affected_components, attack_path, evidence, measurement_results, security_impact, business_context, current_controls, control_boundary, effective_identity, risk_owner, risk_acceptance_authority, selected_treatment, treatment_actions, reassessment_trigger, target_review_date, decision_date, decision_status`

14 Change and Reassessment Triggers

A baseline annual review does not capture the changes that matter most in integrated LLM systems. A new MCP tool, a widened service identity, or a new indexed source can change the reachable operation surface between reviews without any change to the model. The model defines event-driven triggers so that a material change reaches an assessment decision when it happens.

Not every trigger requires a full assessment. The model defines four response types and selects by the scope of what changed. A full reassessment repeats the assessment across the system context and is reserved for changes such as a security incident. A targeted reassessment covers the affected components and the paths that reach them. An evidence refresh re-collects the affected evidence without re-running the full method set. A decision review re-examines an existing decision against the change without new collection.

- Full reassessment: repeat the assessment across the system context.
- Targeted reassessment: assess the affected components and the paths that reach them.
- Evidence refresh: re-collect the affected evidence without re-running the full method set.
- Decision review: re-examine the existing decision against the change without new collection.

ID	Trigger	Component	Response	Required evidence
TRG-01	Model provider change	external_model_provider	Targeted	Data egress and provider terms evidence
TRG-02	Deployment environment change	llm_application	Targeted	Configuration and network evidence
TRG-03	New model endpoint	model_endpoint	Targeted	Endpoint reachability and authentication evidence
TRG-04	Authentication boundary change	model_endpoint	Targeted	Authentication configuration evidence
TRG-05	New RAG data source	rag_data_source	Targeted	Source provenance and classification evidence
TRG-06	New indexed content trust source	rag_pipeline	Targeted	Indexing configuration and provenance evidence
TRG-07	Retrieval authorization change	rag_pipeline	Targeted	Retrieval authorization test evidence
TRG-08	New MCP server	mcp_server	Targeted	Server origin, authentication, and effective permission evidence
TRG-09	New MCP tool	mcp_tool	Targeted	Tool definition and effective permission evidence
TRG-10	MCP capability definition change	mcp_server	Targeted	Capability baseline and change evidence
TRG-11	Service identity change	service_identity	Targeted	

ID	Trigger	Component	Response	Required evidence
				Identity and downstream authorization evidence
TRG-12	Downstream permission increase	downstream_service	Targeted	Downstream role and scope evidence
TRG-13	New agent tool	agent_tool	Targeted	Agent capability and identity evidence
TRG-14	New administrative operation reachable	agent_tool	Targeted	Operation restriction and impact evidence
TRG-15	Network exposure change	model_endpoint	Targeted	Reachability evidence
TRG-16	Data classification change	llm_application	Decision review	Data classification record
TRG-17	Logging loss or degradation	logging_source	Evidence refresh	Logging completeness evidence
TRG-18	Security incident involving the system	llm_application	Full	Incident record and affected-path evidence
TRG-19	Material prompt architecture change	prompt_gateway	Targeted	Prompt and context boundary evidence
TRG-20	Risk exception expiration	llm_application	Decision review	Exception record
TRG-21	Assessment evidence past validity period	llm_application	Evidence refresh	Prior evidence and validity records

15 Worked Example

The worked example applies the model to a single path: an indirect prompt injection that reaches a repository write through a RAG-to-agent execution path. It is a security assessment path, not an offensive procedure, and it contains no injection payloads, bypass strings, or exploitation steps. Its purpose is to show how GOVERN, MAP, MEASURE, and MANAGE apply to one concrete path from an untrusted document to a source control operation.

The path runs from an untrusted document, through an indexing pipeline and retrieval service, into an LLM application and an agent runtime, to an MCP server that exposes a repository tool, which executes under a service account with repository write capability. The example shows why capability approval and effective permission need separate review: the approval names a repository set, while the service account can reach a broader set, and the impact follows the identity's reach.

> This example describes a generalized enterprise assessment environment. It does not represent a specific Penaxtra customer or claim deployment within a named organization. Where a value is shown, it is illustrative and labeled as an assessment result, not telemetry.

Path

```
RAG indexing pipeline INDEXES_FROM Untrusted document
LLM application RETRIEVES_FROM Retrieval service
LLM application SENDS_PROMPT_TO Model endpoint
LLM application RECEIVES_RESPONSE_FROM Model endpoint
Agent runtime CONNECTS_TO_MCP MCP server
MCP server EXPOSES_TOOL Repository tool
Agent runtime INVOKES_TOOL Repository tool
Repository tool EXECUTES_AS Service account
Repository tool WRITES_TO Source control system
```

GOVERN

System owner	Owner of the internal engineering assistant use case.
Technical owner	Team operating the assistant and its agent runtime.
Security assessment owner	Team performing the assessment.
Risk acceptance authority	Authority authorized to accept residual risk for the assistant.
Approved use case	Answer engineering questions and open pull requests on approved repositories.
Approved MCP capability	Read repository metadata and open pull requests on a named repository set.
Approved repository scope	The named repositories the use case requires.
Assessment requirement	Assess the complete retrieval-to-tool path, not the model in isolation.

Evidence responsibility	Assessment owner collects and retains the path evidence.

Capability approval and effective permission are reviewed separately. The approval names a repository set; the effective permission is whatever the service account can reach, which the MAP stage establishes independently.

MAP

Component	Type	Note
Untrusted document	`rag_data_source`	Untrusted; content originates outside the organization's control.
RAG indexing pipeline	`rag_pipeline`	Reads the document and writes it into the retrieval store.
Retrieval service	`rag_pipeline`	Returns retrieved content as model context.
LLM application	`llm_application`	Constructs prompts, retrieves context, and consumes model output.
Model endpoint	`model_endpoint`	The interface the application sends prompts to and receives responses from.
Agent runtime	`agent_runtime`	Plans and invokes tools based on model output.
MCP server	`mcp_server`	Exposes the repository tool to the agent.
Repository tool	`mcp_tool`	Reads metadata and opens pull requests; the component that performs the repository operation.
Service account	`service_identity`	The identity the repository tool runs as.
Source control system	`downstream_service`	The repository system the operation reaches.
Logging sources	`logging_source`	Records for the retrieval, tool, and repository operations.

Assessment path

1. Untrusted document content is indexed and later retrieved as context.
2. Retrieved content influences how the model or agent processes instructions.
3. The influenced processing reaches an available tool decision boundary.
4. A repository operation is invoked through the tool.
5. The operation executes under the service account identity.
6. Repository state may change within whatever the service account can reach.

MEASURE

Measurement	Illustrative result
Document source provenance coverage (CTL-007)	Illustrative result: provenance recorded for approved sources; the affected source set is not covered.

Measurement	Illustrative result
RAG source approval status (INV-005, CHG-003)	Illustrative result: the source is present in the index but not in the approved source inventory.
Indirect prompt injection resistance (CTL-002)	Illustrative result: Fail on the retrieval-to-agent path under the method used.
Retrieved content trust boundary documented (CTL-008)	Illustrative result: Not Observed; retrieved content is placed into context as trusted.
Agent tool inventory coverage (INV-008)	Illustrative result: the repository tool is inventoried.
MCP effective permission review (CTL-010)	Illustrative result: 2 repositories within approved scope; 11 additional repositories reachable through the same service identity.
Repository operation restriction coverage (CTL-011)	Illustrative result: the tool exposes write operations without an operation constraint.
Service identity repository scope (CTL-013, EXP-007)	Illustrative result: the service account holds write across a broader repository set than the use case requires.
Tool invocation logging completeness (CTL-014, RUN-003)	Illustrative result: invocations are recorded with tool and identity; the target repository is not consistently recorded.
Downstream operation attribution (RUN-006)	Illustrative result: repository operations can be joined to the tool invocation for the approved repositories only.

MANAGE: RESTRICT

The path connects an untrusted source to a repository write under a service identity whose scope exceeds the approved repository set. The impact is set by the identity's effective reach, not by the tool's description. Restriction reduces that reach and constrains the reachable operation while the assistant continues to serve its use case.

- Separate the read identity from the write identity so retrieval and metadata reads do not run as an identity that can write.
- Limit the service account to the approved repository set so the reachable operation cannot extend beyond the use case.
- Constrain the repository tool to the specific operations the use case requires.
- Record MCP capability changes so a new tool or permission triggers reassessment.
- Record the target repository on every tool invocation so downstream operations are attributable.
- Require a targeted reassessment when the repository scope or the service identity changes, and retest the complete retrieval-to-agent path.

16 Implementation Considerations

SaaS model services and self-hosted models expose different evidence. A provider relationship supplies contract and processing terms; a self-hosted deployment supplies configuration and runtime records. Cloud resource inventory may enumerate managed AI services but will not identify browser-based AI use, and source code dependency analysis may identify a model SDK without establishing that the SDK is in active runtime use.

Runtime monitoring records observed behavior, not all possible capability, and point-in-time testing can become stale after an agent or MCP capability change. Asset discovery does not establish ownership, tool inventory does not establish effective permissions, and a declared MCP tool description does not establish the downstream service boundary the tool can reach.

Prompt filtering is not a complete prompt injection control; it removes a subset of phrasings and must be combined with context isolation, output handling, and constrained execution. Human approval is not universally required for every agent operation, and approval boundaries should be based on effective operation and impact rather than applied uniformly. RAG controls must be assessed across source, indexing, retrieval, and generated context, because a control at one stage does not cover the others.

Service identities must be reviewed against downstream authorization, not their descriptions. Security measurements require interpretation in system context, and every risk treatment needs an accountable decision owner. A measurement without an interpretation boundary and a decision without an owner are both incomplete records.

17 Limitations

- This is an independent community-developed implementation model. It is not an official NIST publication, an official NIST AI RMF profile, or a NIST-validated use case.
- NIST does not endorse Penaxtra or this methodology, and this model does not modify the AI RMF.
- The model is not a certification methodology.
- The model is focused on LLM and generative AI security assessment. It does not define general AI safety evaluation, and it does not define bias or fairness measurement methodology.
- The model does not provide legal interpretation and does not replace organization-specific risk tolerance decisions.
- The model does not establish universal measurement thresholds. Where a threshold is organization-specific, it is stated as such.
- A given system may require additional evidence or measurements beyond those cataloged here.
- The use case is generalized and does not represent a specific organization.

18 Version History

Version	Date	Change
1.0	2026-07-14	Initial publication of the NIST AI RMF implementation model for LLM security assessments.

19 References

[1] **Artificial Intelligence Risk Management Framework (AI RMF 1.0)**. National Institute of Standards and Technology. NIST AI 100-1. https://doi.org/10.6028/NIST.AI.100-1. Source of the AI RMF Core functions, categories, and subcategory outcomes referenced in this model. AI RMF 1.0 (2023) is the published version; NIST has stated the framework is subject to ongoing update.

[2] **NIST AI RMF Core**. National Institute of Standards and Technology (AIRC). https://airc.nist.gov/airmf-resources/airmf/5-sec-core/. Source of truth for the GOVERN, MAP, MEASURE, and MANAGE subcategory identifiers and wording used here.

[3] **NIST AI RMF Playbook**. National Institute of Standards and Technology (AIRC). https://airc.nist.gov/airmf-resources/playbook/. Companion voluntary resource with suggested actions for the AI RMF subcategories. The Playbook is not a mandatory sequential checklist and is subject to update.

[4] **NIST AIRC AI RMF Use Cases**. National Institute of Standards and Technology (AIRC). https://airc.nist.gov/airmf-resources/usecases/. AIRC page describing AI RMF use cases and the submission channel. NIST does not validate or endorse individual organizations or their approaches to using the AI RMF.

[5] **Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile**. National Institute of Standards and Technology. NIST AI 600-1. https://doi.org/10.6028/NIST.AI.600-1. Cross-sectoral profile referenced for generative AI risk context.

[6] **Open LLM Security Posture Mapping for NIST CSF 2.0**. Penaxtra Security Research. Version 1.0. https://penaxtra.com/research/llm-security-posture-mapping. Related Penaxtra Security Research. A community mapping of LLM security concerns to NIST Cybersecurity Framework 2.0 outcomes, and a source of an existing LLM security taxonomy. It is not a NIST publication.

20 Citation

Penaxtra Security Research. "Operationalizing the NIST AI RMF for LLM Security Assessments." Version 1.0. Penaxtra. https://penaxtra.com/research/operationalizing-ai-rmf-for-llm-security

This resource is an independent community-developed implementation model maintained by Penaxtra Security Research. Penaxtra is developed and operated by Seccops Cyber Security Technologies Inc. Penaxtra is not affiliated with or endorsed by the National Institute of Standards and Technology. References to NIST and the NIST AI Risk Management Framework are provided for implementation and research purposes. This implementation model does not represent an official NIST profile or a NIST-validated use case, and it does not modify the AI RMF.