PENAXTRA SECURITY RESEARCH

Open LLM Security Posture Mapping for NIST CSF 2.0

A vendor-neutral mapping of LLM and generative AI security risks to the NIST Cybersecurity Framework 2.0.

VERSION	**1.0**
PUBLICATION TYPE	**Security Mapping**
FRAMEWORK	**NIST Cybersecurity Framework 2.0**
STATUS	**Published**
LAST UPDATED	**2026-07-13**
RESOURCE	**https://penaxtra.com/research/llm-security-posture-mapping**

Abstract

This resource maps the security risks of LLM and generative AI systems to the outcomes of the NIST Cybersecurity Framework 2.0. It defines a taxonomy of 49 security concerns with stable identifiers, organized into nine categories that span AI asset inventory, model endpoint exposure, shadow AI, prompt injection, system prompt exposure, sensitive data in prompt and response flows, retrieval-augmented generation security, Model Context Protocol server and tool security, AI agent execution, monitoring, and response. Each concern is related to NIST CSF 2.0 outcomes through two labeled relationship types, primary and supporting, with a rationale for each of the 145 relationships. The mapping uses official CSF 2.0 Function, Category, and Subcategory identifiers and is presented as a community resource for security assessment. It is not an official NIST crosswalk. Its purpose is to let an organization assess LLM and generative AI exposure with the framework it already applies to the rest of its environment, supported by an explicit model of the assessment evidence that demonstrates each outcome.

Keywords: LLM security; generative AI security; NIST Cybersecurity Framework 2.0; AI security posture; prompt injection; RAG security; MCP security; AI agent security; model endpoint; security assessment

Contents

1 Overview

Traditional cybersecurity asset inventories and control models do not always describe the relationships between LLM applications, model endpoints, AI agents, RAG pipelines, MCP servers, external model providers, prompt flows, and tool execution paths. A control model built for hosts, networks, and applications can record that an AI system exists without capturing what it can read, what it can do, and which untrusted inputs influence its behavior.

This resource establishes a structured security taxonomy for LLM and generative AI systems and relates the identified security concerns to NIST Cybersecurity Framework 2.0 outcomes. The taxonomy provides an assessment lens: a set of stable identifiers, each describing a security objective, the condition that undermines it, the outcome that resolves it, and the evidence that demonstrates the outcome.

The mapping is a community-developed resource. It does not modify NIST CSF 2.0, and it does not create new NIST requirements. It relates LLM security concerns to existing CSF 2.0 outcomes so that an organization can assess LLM exposure using the framework it already applies to the rest of its environment. Organizations should interpret the mapping according to system context, risk tolerance, architecture, and regulatory obligations.

> This resource is not endorsed by NIST. It is an independent mapping. See the Limitations and the publication note for the full statement.

2 Purpose

The resource supports security work that requires a structured view of LLM and generative AI exposure. It is intended for use in:

- LLM security assessments
- AI asset inventory reviews
- Security posture reviews
- Security architecture reviews
- Technical control gap analysis
- AI governance evidence collection
- LLM deployment reviews
- AI agent risk reviews
- MCP security reviews
- RAG security reviews

The mapping can be used independently of any commercial product. The taxonomy identifiers, the CSF relationships, and the assessment evidence descriptions are sufficient to structure an assessment using an organization's own tooling and records.

3 Scope

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

- Hosted LLM APIs and self-hosted models
- Enterprise AI applications, internal AI assistants, and copilots
- AI agents and agentic workflows
- RAG systems and the vector databases that participate in retrieval flows
- MCP servers, MCP clients, and MCP tools
- Model gateways and prompt gateways
- Model endpoints and AI application APIs
- Cloud AI services and external model providers
- Model dependencies and AI-related deployment dependencies where they bear on supply chain risk

The following are outside scope:

- General-purpose machine learning risk not associated with LLM or generative AI systems
- Model quality benchmarking
- AI ethics scoring and bias measurement methodology
- General privacy law interpretation
- Formal certification and NIST compliance certification

The taxonomy is deliberately bounded to LLM and generative AI system security. It does not attempt to cover the full breadth of AI or machine learning risk.

4 Mapping Methodology

Each concern in the taxonomy carries a stable identifier with the prefix PNX-LLM, followed by a two- or three-letter category code and a sequence number, for example PNX-LLM-RA-03. These identifiers are assigned for this taxonomy. They are not drawn from an external standard and do not correspond to OWASP, MITRE ATLAS, or NIST identifiers. Their purpose is to let an assessment reference a concern precisely and consistently across versions. Where a concern relates to an external catalog, that relationship is expressed through the mapping rather than through the identifier.

Each concern is evaluated against six attributes: the security objective it serves, the asset or process it affects, the threat or failure condition that undermines it, the expected security outcome, the assessment evidence that demonstrates the outcome, and the relevant NIST CSF 2.0 outcome.

Relationships are selected according to the intended outcome of the CSF 2.0 Category or Subcategory, not according to shared vocabulary. A concern is related to a CSF outcome when achieving that outcome contributes to the concern's expected result. Where the relationship is uncertain, the mapping selects the narrower defensible relationship rather than adding several weak ones.

A single concern often relates to more than one CSF outcome because addressing it involves several activities: identifying the risk, applying a protective control, monitoring for the condition at runtime, analyzing events, and responding. The mapping distinguishes two relationship types.

- Primary Mapping: a NIST CSF 2.0 outcome directly associated with the security objective of the concern.
- Supporting Mapping: a NIST CSF 2.0 outcome that contributes to the implementation, monitoring, or response for the concern.

The CSF 2.0 relationships were additionally cross-checked against NIST IR 8596, the Cybersecurity Framework Profile for Artificial Intelligence (Cyber AI Profile), which organizes AI cybersecurity risk using the same CSF 2.0 outcomes. That document is an initial preliminary draft and is used here only as a reference point; it does not set the relationships in this resource, which remain the maintainer's own determination.

These relationship labels appear in the mapping table, the downloadable datasets, and the machine-readable file. The mapping is not an official NIST crosswalk. It is a community mapping intended to be defensible to a security architect reviewing it against the CSF 2.0 core.

5 LLM Security Risk Taxonomy

The taxonomy contains 49 entries across 9 categories. Each entry carries a stable identifier, a security objective, the risk condition it addresses, the expected outcome, the assessment evidence that demonstrates the outcome, and its primary and supporting NIST CSF 2.0 relationships.

5.1 Asset Management

PNX-LLM-AM-01 AI Asset Inventory

Security objective. Maintain an inventory of LLM applications, model endpoints, agent runtimes, RAG systems, MCP servers, and the external model services they depend on.

An organization cannot assess LLM exposure without identifying the systems, endpoints, data stores, and external services involved. An AI asset inventory records each LLM application and its supporting components, the environment it runs in, its owner, and the authentication boundary that governs access to it.

Risk condition. AI systems are deployed by product and engineering teams without a corresponding security inventory, so the model endpoints, agents, and integrations in production are not fully known to the teams responsible for assessing them.

Expected outcome. The set of LLM applications and their supporting components is identified, attributed to an owner, and kept current as systems are added, changed, and retired.

ASSESSMENT EVIDENCE

- AI application inventory
- Model endpoint inventory
- Agent and MCP server inventory
- Ownership and environment records
- System lifecycle records

Implementation considerations. Different discovery methods reveal different parts of the surface. Cloud service inventory, source dependencies, API gateway configuration, and DNS each expose a subset. Treat the inventory as a reconciliation of several sources rather than the output of one.

Primary CSF 2.0 mapping	ID.AM-02
Supporting CSF 2.0 mapping	ID.AM-05, ID.AM-08

PNX-LLM-AM-02 Model Endpoint Discovery

Security objective. Identify the model API endpoints reachable from applications, agents, and internal networks, including test, development, and administrative endpoints.

A model endpoint is the network interface through which an application submits prompts and receives responses. Endpoints exist for hosted provider APIs, self-hosted models, and model gateways. Discovery establishes which endpoints exist, where they are reachable from, and how they authenticate.

Risk condition. Model endpoints are exposed on internal or public networks without a record of their existence, or non-production endpoints remain reachable after their intended use has ended.

Expected outcome. Reachable model endpoints are enumerated and associated with an application, an environment, and an access boundary.

ASSESSMENT EVIDENCE

- Model endpoint inventory
- Network reachability records
- Endpoint authentication configuration
- Gateway routing configuration

Implementation considerations. Endpoint discovery based only on documented architecture will miss endpoints introduced during development. Combine configuration review with observed network flows where available.

Primary CSF 2.0 mapping	ID.AM-02
Supporting CSF 2.0 mapping	ID.AM-03, ID.RA-01

PNX-LLM-AM-03 External Model Service Identification

Security objective. Identify external model providers and hosted AI services that process organizational prompts and data.

Applications commonly send prompts to external model providers. Each provider is a supplier that processes organizational data. Identification records which providers are in use, which applications use them, and what data classification reaches them.

Risk condition. External model providers are integrated at the application layer without being recorded as suppliers, so the data that leaves the organization and the terms that govern it are not tracked.

Expected outcome. External model providers and hosted AI services are identified, attributed to the applications that use them, and treated as suppliers within existing supplier management.

ASSESSMENT EVIDENCE

- External model provider inventory
- Application-to-provider mapping
- Data classification reaching each provider
- Provider agreement references

Implementation considerations. A single application may call several providers, and a provider relationship may be introduced through a dependency or an SDK rather than a direct integration. Reconcile the provider list against outbound network destinations.

Primary CSF 2.0 mapping	ID.AM-04
Supporting CSF 2.0 mapping	GV.SC-04, ID.AM-02

PNX-LLM-AM-04 Shadow AI Identification

Security objective. Identify LLM and generative AI usage that is not recorded in the approved inventory.

Shadow AI is the use of model APIs, embedded SDKs, browser-based AI services, SaaS AI features, application plugins, or developer tooling that is not recorded in the approved inventory. It is defined by the absence of a governance and security record rather than by the identity of the service.

Risk condition. Model integrations reach production through SaaS features, embedded SDKs, plugins, or undocumented endpoints referenced in code and configuration, and these integrations are not represented in the inventory used for assessment.

Expected outcome. Previously unrecorded AI usage is identified, evaluated, and either brought under management or removed.

ASSESSMENT EVIDENCE

- Source and configuration references to model endpoints
- SaaS AI feature inventory
- Dependency and SDK inventory
- Outbound connections to model provider domains
- Browser and endpoint telemetry where available

Implementation considerations. Scoping matters. Treating all consumer use of a public assistant as shadow AI produces noise. Focus on integrations that process organizational data or connect to organizational systems.

Primary CSF 2.0 mapping	ID.AM-02
Supporting CSF 2.0 mapping	DE.CM-09, ID.AM-04

PNX-LLM-AM-05 AI Dependency Inventory

Security objective. Maintain an inventory of the software dependencies that compose LLM systems, including SDKs, agent frameworks, MCP servers, and model gateways.

LLM systems are assembled from model provider SDKs, agent frameworks, MCP server implementations, model gateways, and supporting libraries. A dependency inventory records these components and their versions so that a known vulnerability or a compromised package can be traced to the systems that use it.

Risk condition. AI-specific dependencies are introduced without inventory, so a vulnerability disclosure in an agent framework or an MCP implementation cannot be mapped to affected systems.

Expected outcome. The components that compose LLM systems are inventoried at a version granularity that supports vulnerability and change tracking.

ASSESSMENT EVIDENCE

- Dependency manifests
- Agent framework and SDK versions
- MCP server implementation records
- Model gateway component records
- Container image contents

Implementation considerations. AI dependencies change quickly and are sometimes pinned to pre-release versions. Record the resolved versions in use rather than only the declared ranges.

Primary CSF 2.0 mapping	ID.AM-02
Supporting CSF 2.0 mapping	GV.SC-04, ID.RA-09

5.2 Risk Assessment

PNX-LLM-RA-01 LLM Application Risk Assessment

Security objective. Assess the security risk of an LLM application based on its function, data exposure, integrations, and execution capability.

An LLM application risk assessment combines the sensitivity of the data it processes, the systems it can reach, the actions it can take, and the trust placed in its inputs. The result informs the depth of testing and the controls applied.

Risk condition. LLM applications are deployed at uniform assurance regardless of the data they process or the actions they can perform, so high-impact applications receive the same review as low-impact ones.

Expected outcome. Each LLM application has a recorded risk determination that reflects data exposure, integration reach, and execution capability, and that determination guides control selection.

ASSESSMENT EVIDENCE

- Application data flow records
- Integration and tool inventory
- Execution capability records
- Recorded risk determination

Implementation considerations. Risk changes when an application gains a new tool, a new data source, or a new agent capability. Reassessment should be tied to those changes rather than to a fixed calendar alone.

Primary CSF 2.0 mapping	ID.RA-05
Supporting CSF 2.0 mapping	ID.RA-03, ID.RA-04

PNX-LLM-RA-02 Model Endpoint Exposure

Security objective. Assess the exposure of model endpoints, including reachability, authentication strength, and the presence of test or administrative interfaces.

Model endpoint exposure describes the conditions under which an endpoint can be reached and used. It covers public and internal reachability, the strength of the access boundary, and the presence of development or administrative endpoints that were not intended for production traffic.

Risk condition. A model endpoint is reachable without an adequate access boundary, or a test, development, or administrative endpoint is exposed on a network segment that untrusted parties can reach.

Expected outcome. Model endpoint reachability and access boundaries are assessed and recorded, and endpoints without an adequate boundary are remediated.

ASSESSMENT EVIDENCE

- Endpoint reachability records
- Authentication and authorization configuration
- Gateway exposure configuration
- Rate and quota configuration

Implementation considerations. Rate control limits abuse but does not establish an access boundary. Treat quotas as supporting evidence alongside authentication and authorization rather than as a replacement for them.

Primary CSF 2.0 mapping	ID.RA-01
Supporting CSF 2.0 mapping	PR.AA-05, PR.IR-01

PNX-LLM-RA-03 AI Attack Path Analysis

Security objective. Analyze how an input source, an LLM application, an agent, its tools, and downstream systems combine into a reachable path to a sensitive operation.

AI attack path analysis treats the LLM system as a graph. It examines how an entry point, a prompt or content source, an agent, its available tools, the identity those tools run as, and a downstream system connect. The analysis identifies paths where untrusted input can reach a consequential operation.

Risk condition. Individual components are assessed in isolation, so a path that connects an untrusted input to a high-impact operation through an agent and its tools is not identified.

Expected outcome. Paths from untrusted input to consequential operations are identified and recorded, and the paths that carry the greatest impact are prioritized for control.

ASSESSMENT EVIDENCE

- Integration and tool graph
- Identity and permission records for tool execution
- Data flow records
- Recorded attack path analysis

Implementation considerations. The value of the analysis depends on accurate effective-permission data. A path is only as constrained as the identity its final tool call runs as.

Primary CSF 2.0 mapping	ID.RA-04
Supporting CSF 2.0 mapping	ID.RA-03, ID.RA-05

PNX-LLM-RA-04 Agent Capability Risk

Security objective. Assess the risk represented by an agent's effective execution capability rather than by its stated purpose.

An agent's risk is a function of what it can do, not what it is intended to do. Capability risk assessment records the tools an agent can invoke, the identity those invocations use, and the operations reachable through them.

Risk condition. An agent is assessed by its described role, while the tools and identities available to it permit operations well beyond that role.

Expected outcome. Agent risk is assessed against effective capability, and capability that exceeds the intended function is reduced or justified.

ASSESSMENT EVIDENCE

- Agent tool inventory
- Effective execution identity records
- Reachable operation records
- Recorded capability risk determination

Implementation considerations. Capability is often broader than intended because a tool inherits the permissions of the service identity behind it. Assess the service identity behind the tool as well as the tool description.

Primary CSF 2.0 mapping	ID.RA-05
Supporting CSF 2.0 mapping	PR.AA-05, ID.RA-04

PNX-LLM-RA-05 AI Supply Chain Risk

Security objective. Assess the security risk contributed by model providers, model artifacts, agent frameworks, MCP servers, and other AI system suppliers.

AI supply chain risk covers the providers and components an AI system depends on to function: model providers, model artifacts, SDKs, agent frameworks, MCP servers, and model gateways. Assessment considers the security ownership of each and the consequences of a compromise or a change.

Risk condition. AI system suppliers and components are adopted without assessment, so a compromised model artifact, framework, or connected server is not recognized as a security dependency.

Expected outcome. AI system suppliers and components are assessed before adoption and monitored over the relationship, with critical dependencies identified.

ASSESSMENT EVIDENCE

- Supplier and component inventory
- Provenance and integrity records
- Supplier assessment records
- Change and version records

Implementation considerations. AI supply chain risk overlaps general software supply chain risk but adds model artifacts and connected servers that execute or influence application behavior. Keep the assessment specific to those components.

Primary CSF 2.0 mapping	GV.SC-07
Supporting CSF 2.0 mapping	ID.RA-09, GV.SC-04

5.3 Prompt Security

PNX-LLM-PR-01 Direct Prompt Injection

Security objective. Preserve the application's control instructions when a user submits input intended to override them.

Direct prompt injection is input submitted by a user of the application that attempts to override the application's instructions or policy. The relevant boundary is between application-controlled context and user-controlled input. An LLM does not enforce an instruction hierarchy on its own, so the application cannot assume that its instructions take precedence over user text.

Risk condition. User input alters the application's intended behavior, causing it to disregard its instructions, disclose restricted context, or take actions outside its intended scope.

Expected outcome. The application's control instructions are preserved under adversarial user input, and deviations are detected.

ASSESSMENT EVIDENCE

- Prompt construction and context handling design
- Adversarial prompt test results
- Runtime prompt security events
- Instruction and policy separation records

Implementation considerations. Pattern filtering removes a subset of known phrasings and does not establish the boundary. Treat filtering as one supporting control alongside context isolation, output handling, and constrained execution.

Primary CSF 2.0 mapping	PR.DS-10
Supporting CSF 2.0 mapping	ID.RA-01, DE.CM-09

PNX-LLM-PR-02 Indirect Prompt Injection

Security objective. Preserve the application's control instructions when instructions are embedded in content the model reads, such as retrieved documents, web pages, or tool outputs.

Indirect prompt injection places adversarial instructions in content that the model consumes rather than in direct user input. Sources include retrieved documents, web content, files, and tool outputs. Because the application places this content into the model's context, retrieved content becomes an input trust boundary.

Risk condition. Content read by the model carries instructions that alter application behavior, and the application treats retrieved or tool-provided content as trusted context.

Expected outcome. Content consumed by the model is treated as untrusted input, and its influence on application behavior is bounded and observable.

ASSESSMENT EVIDENCE

- Retrieval and content ingestion design
- Content provenance records
- Adversarial content test results
- Runtime prompt security events

Implementation considerations. The exposure grows with the number of content sources an application reads and with the actions available once content is in context. Assess ingestion, retrieval, and the actions reachable afterward together.

Primary CSF 2.0 mapping	PR.DS-10
Supporting CSF 2.0 mapping	PR.DS-01, DE.CM-09

PNX-LLM-PR-03 System Prompt Exposure

Security objective. Protect the confidentiality of application-controlled instructions and the security-relevant behavior they encode.

System prompt exposure occurs through explicit extraction, partial disclosure, behavioral inference, error paths, logging, debugging interfaces, downstream tool context, and multi-turn leakage. Preventing literal disclosure of the system prompt and protecting the security-relevant application behavior it encodes are related but not identical objectives.

Risk condition. The system prompt is disclosed in full or in part, or its security-relevant contents are inferred, exposing embedded rules, references, or credentials.

Expected outcome. Application-controlled instructions are protected from disclosure, and security decisions do not depend on the secrecy of the prompt alone.

ASSESSMENT EVIDENCE

- Prompt handling and logging design
- Error and debug path review
- Extraction and inference test results
- Downstream context propagation records

Implementation considerations. A system prompt that carries a secret or a credential remains a risk even when literal extraction is prevented. Move security-relevant enforcement out of the prompt where the design allows.

Primary CSF 2.0 mapping	PR.DS-10
Supporting CSF 2.0 mapping	ID.RA-01, DE.CM-09

PNX-LLM-PR-04 Prompt Control Bypass

Security objective. Maintain the application's safety and policy controls against inputs designed to circumvent them.

Prompt control bypass is the circumvention of an application's safety, policy, or guardrail controls through crafted input. The concern is whether the controls hold under adversarial conditions, independent of the specific phrasing used to test them.

Risk condition. Application controls that restrict behavior are circumvented through crafted input, so the application performs actions or produces output that the controls were intended to prevent.

Expected outcome. Safety and policy controls resist circumvention under adversarial input, and bypass conditions are identified and corrected.

ASSESSMENT EVIDENCE

- Guardrail and policy configuration
- Bypass test results
- Runtime policy enforcement events
- Control improvement records

Implementation considerations. A control that depends only on input pattern matching is bypassable by construction. Evaluate controls against a range of phrasings and encodings and treat single-pattern controls as incomplete.

Primary CSF 2.0 mapping	PR.DS-10
Supporting CSF 2.0 mapping	PR.PS-06, ID.IM-02

PNX-LLM-PR-05 Cross-Context Prompt Manipulation

Security objective. Prevent input in one user, session, or tenant context from influencing another.

Cross-context prompt manipulation covers cases where context, memory, or retrieved content from one user, session, or tenant influences another. It applies to shared memory stores, cached context, and multi-tenant retrieval where separation is assumed but not enforced.

Risk condition. Context or content from one user, session, or tenant reaches another, so the application acts on data outside the current requester's boundary.

Expected outcome. Context is isolated across users, sessions, and tenants, and cross-context influence is prevented and detectable.

ASSESSMENT EVIDENCE

- Context and memory isolation design
- Multi-tenant separation records
- Cross-context test results
- Runtime context access events

Implementation considerations. Shared memory and caching introduce cross-context paths that are invisible in a single-session test. Assess separation with concurrent, multi-identity scenarios.

Primary CSF 2.0 mapping	PR.DS-10
Supporting CSF 2.0 mapping	PR.AA-05, DE.CM-09

5.4 Data Security

PNX-LLM-DS-01 Sensitive Data in Prompts

Security objective. Control the sensitive data that enters prompts and is transmitted to model endpoints and providers.

Sensitive data reaches prompts through user input, application context, and retrieved content. It may include PII, credentials, API keys, internal documents, source code, customer data, regulated data, and security telemetry. The concern is the confidentiality of that data as it is transmitted to the model and processed by the provider.

Risk condition. Sensitive data is placed into prompts and transmitted to a model endpoint or external provider without a control that reflects its classification.

Expected outcome. The sensitive data that enters prompts is identified and controlled according to its classification and the destination that processes it.

ASSESSMENT EVIDENCE

- Prompt data flow records
- Data classification reaching model endpoints
- Provider processing terms
- Redaction and minimization configuration

Implementation considerations. The applicable controls differ between a self-hosted model and an external provider. Assess the destination and its data handling as part of the control decision.

Primary CSF 2.0 mapping	PR.DS-02
Supporting CSF 2.0 mapping	PR.DS-10, ID.AM-07

PNX-LLM-DS-02 Sensitive Data in Model Responses

Security objective. Control the disclosure of sensitive data in model responses returned to users or downstream systems.

Model responses can contain sensitive data drawn from training, retrieval, tool outputs, or context. The concern is the disclosure of that data to a recipient who is not authorized for it, whether an end user, a downstream system, or a log.

Risk condition. A model response returns sensitive data to a recipient that is not authorized for it, or to a downstream system or log that is not controlled for that classification.

Expected outcome. Sensitive data in model responses is controlled according to the recipient and the destination.

ASSESSMENT EVIDENCE

- Response handling design
- Output classification records
- Response monitoring events
- Downstream destination records

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

Primary CSF 2.0 mapping	PR.DS-02
Supporting CSF 2.0 mapping	PR.DS-10, DE.CM-09

PNX-LLM-DS-03 Prompt and Response Retention Exposure

Security objective. Control the retention of prompts and responses in application logs, provider logs, and analytics stores.

Prompts and responses are frequently retained for debugging, analytics, and model improvement. Retention creates data-at-rest that carries the sensitivity of the original content. The concern is the confidentiality and access control of that retained data.

Risk condition. Prompts and responses are retained in logs or analytics stores without controls that reflect the sensitivity of their content, or without a defined retention boundary.

Expected outcome. Retained prompts and responses are protected and access-controlled according to their classification, with a defined retention boundary.

ASSESSMENT EVIDENCE

- Prompt and response logging configuration
- Retention configuration
- Access control on log and analytics stores
- Data inventory for retained content

Implementation considerations. Retained prompt and response data often outlives the application that produced it and is copied into analytics pipelines. Follow the data to every store it reaches.

Primary CSF 2.0 mapping	PR.DS-01
Supporting CSF 2.0 mapping	PR.AA-05, ID.AM-07

PNX-LLM-DS-04 External Model Data Transfer

Security objective. Control the transfer of organizational data to external model providers and hosted AI services.

External model data transfer is the flow of prompts, context, and retrieved content to a provider outside the organization. The concern is the confidentiality of that data in transit, the terms under which the provider processes it, and the classification of what is sent.

Risk condition. Organizational data is transferred to an external provider without a control that reflects its classification or the provider's processing and retention terms.

Expected outcome. Data transferred to external providers is controlled in transit and governed by supplier terms appropriate to its classification.

ASSESSMENT EVIDENCE

- Outbound data flow records
- Provider agreements and processing terms
- Data classification per provider
- Transport protection configuration

Implementation considerations. The provider agreement is part of the control. Assess processing, retention, and sub-processor terms alongside the transport protection.

Primary CSF 2.0 mapping	PR.DS-02
Supporting CSF 2.0 mapping	GV.SC-05, GV.SC-07

PNX-LLM-DS-05 AI Logging Data Exposure

Security objective. Control the sensitive data that AI systems write to logs, traces, and telemetry.

AI systems generate logs, traces, and telemetry that can contain prompts, responses, tool inputs, credentials, and identifiers. The concern is the exposure of that sensitive data through logging that is broadly readable or retained without control.

Risk condition. AI logging and telemetry capture sensitive data that is then readable or retained without controls that reflect its sensitivity.

Expected outcome. AI logging captures the records needed for monitoring while sensitive content is controlled at generation and at rest.

ASSESSMENT EVIDENCE

- Logging and telemetry configuration
- Access control on log stores
- Sensitive field handling in logs
- Retention configuration

Implementation considerations. Logging that supports detection and logging that leaks sensitive data are in tension. Configure what is captured and who can read it as one decision.

Primary CSF 2.0 mapping	PR.DS-01
Supporting CSF 2.0 mapping	PR.PS-04, PR.AA-05

5.5 RAG Security

PNX-LLM-RAG-01 RAG Data Source Inventory

Security objective. Maintain an inventory of the data sources indexed into retrieval pipelines and the classification of their content.

A RAG system answers using content retrieved from indexed sources. A data source inventory records which sources feed the index, who owns them, and the classification of their content, so that retrieval scope and authorization can be assessed against known sources.

Risk condition. Sources are indexed into a retrieval pipeline without an inventory, so the content that a model can retrieve and its sensitivity are not fully known.

Expected outcome. The data sources feeding retrieval are inventoried, classified, and attributed to owners.

ASSESSMENT EVIDENCE

- RAG data source inventory
- Content classification records
- Indexing pipeline configuration
- Source ownership records

Implementation considerations. Indexes accumulate sources over time and across teams. Reconcile the inventory against the indexing pipeline configuration rather than against documentation alone.

Primary CSF 2.0 mapping	ID.AM-07
Supporting CSF 2.0 mapping	ID.AM-02, ID.AM-05

PNX-LLM-RAG-02 Retrieval Authorization Exposure

Security objective. Enforce that retrieval returns only content the requester is authorized to access.

Retrieval authorization determines whether a retrieval respects the requester's access rights. When authorization is applied at the application rather than at retrieval, a user can receive content through the model that they could not access directly.

Risk condition. Retrieval returns content without applying the requester's access rights, so a user obtains restricted content through generated answers.

Expected outcome. Retrieval enforces the requester's authorization, and access decisions are applied at retrieval rather than only at the application boundary.

ASSESSMENT EVIDENCE

- Retrieval authorization design
- Access control configuration on retrieval
- Tenant and user scoping records
- Authorization test results

Implementation considerations. Oversharing is often introduced when retrieval runs with a broad service identity. Assess the identity retrieval executes as, not only the application's front-door controls.

Primary CSF 2.0 mapping	PR.AA-05
Supporting CSF 2.0 mapping	PR.DS-01, ID.AM-05

PNX-LLM-RAG-03 RAG Poisoning

Security objective. Preserve the integrity of indexed content against malicious or unauthorized modification.

RAG poisoning is the introduction of malicious or misleading content into a retrieval source or index. It can occur through a compromised source, unauthorized document insertion, or the exploitation of a trusted-source assumption. Poisoned content can carry indirect prompt injection or bias generated answers.

Risk condition. Content in a retrieval source or index is modified or inserted by an unauthorized party, and the pipeline treats indexed content as trusted.

Expected outcome. The integrity and provenance of indexed content are protected, and unauthorized modification is detectable.

ASSESSMENT EVIDENCE

- Source provenance records
- Index write access controls
- Content change monitoring
- Re-indexing and validation records

Implementation considerations. Provenance and change monitoring matter because poisoning frequently exploits a source the pipeline already trusts. Assess integrity at the source and at re-indexing.

Primary CSF 2.0 mapping	PR.DS-01
Supporting CSF 2.0 mapping	DE.CM-09, PR.DS-10

PNX-LLM-RAG-04 Untrusted Retrieved Content

Security objective. Treat retrieved content as untrusted input to the model rather than as trusted context.

Retrieved content enters the model's context and can carry instructions, misleading data, or references that alter behavior. The trust placed in retrieved content is a security decision. Content that originates outside the organization or from user-writable sources warrants the same scrutiny as direct user input.

Risk condition. Retrieved content is placed into the model context as trusted, so instructions or data embedded in it influence application behavior.

Expected outcome. Retrieved content is handled as untrusted input, and its influence on behavior and actions is bounded.

ASSESSMENT EVIDENCE

- Retrieval handling design
- Content provenance and trust records
- Adversarial content test results
- Action constraints after retrieval

Implementation considerations. The risk is proportional to what the application can do after retrieval. Constrain the actions reachable once retrieved content is in context.

Primary CSF 2.0 mapping	PR.DS-10
Supporting CSF 2.0 mapping	PR.DS-01, DE.CM-09

PNX-LLM-RAG-05 Vector Store Exposure

Security objective. Protect the vector store and its metadata from unauthorized access and modification.

A vector store holds embedded representations of source content along with chunk metadata. Exposure covers unauthorized read access to embeddings and metadata, unauthorized write access that supports poisoning, and network reachability of the store itself.

Risk condition. A vector store is reachable or modifiable without an adequate access boundary, exposing embedded content, metadata, or the index to unauthorized parties.

Expected outcome. Access to the vector store and its metadata is authenticated, authorized, and bounded to the systems that require it.

ASSESSMENT EVIDENCE

- Vector store access configuration
- Network reachability records
- Metadata handling records
- Store inventory

Implementation considerations. Chunk metadata can carry source paths, identifiers, and access hints that are sensitive independent of the embeddings. Assess metadata exposure alongside the embeddings.

Primary CSF 2.0 mapping	PR.DS-01
Supporting CSF 2.0 mapping	PR.AA-05, PR.IR-01

5.6 MCP Security

PNX-LLM-MCP-01 MCP Server Inventory

Security objective. Maintain an inventory of MCP servers connected to LLM applications and agents, including their origin and the tools they expose.

An MCP server exposes tools, resources, and prompts to an LLM application or agent. An inventory records which servers are connected, where they originate, and what they expose, so that the capability reaching applications through MCP is known.

Risk condition. MCP servers are connected to applications and agents without an inventory, so the tools and resources reachable through MCP are not fully known.

Expected outcome. Connected MCP servers are inventoried and attributed to the applications and agents that use them, with their origin recorded.

ASSESSMENT EVIDENCE

- MCP server inventory
- Server origin records
- Tool and resource definitions
- Application-to-server mapping

Implementation considerations. MCP servers can be added by developers during integration work and can proxy to further servers. Reconcile the inventory against actual client configuration.

Primary CSF 2.0 mapping	ID.AM-02
Supporting CSF 2.0 mapping	ID.AM-04, ID.AM-05

PNX-LLM-MCP-02 MCP Server Authentication

Security objective. Authenticate MCP servers and the connections to them, and manage the credentials involved.

MCP connections carry tool invocations that can reach organizational systems. Authentication establishes that a connection is to the intended server and that the server is authorized to be used. It covers the credentials the client presents, the credentials the server holds for downstream services, and the transport that carries them.

Risk condition. An MCP connection is established without authenticating the server or the client, or downstream credentials held by the server are not managed.

Expected outcome. MCP servers and connections are authenticated, and the credentials involved are managed and protected.

ASSESSMENT EVIDENCE

- MCP authentication configuration
- Credential management records
- Transport configuration
- Downstream credential inventory

Implementation considerations. The credentials an MCP server holds for downstream services are frequently more sensitive than the client connection. Assess both directions of authentication.

Primary CSF 2.0 mapping	PR.AA-03
Supporting CSF 2.0 mapping	PR.AA-01, PR.AA-04

PNX-LLM-MCP-03 MCP Tool Permission Exposure

Security objective. Limit MCP-connected tools to the permissions and operations required by the approved use case.

MCP tools may expose file access, command execution, data retrieval, API operations, or administrative functions to an LLM application or agent. Tool availability alone is not sufficient evidence that the resulting execution boundary is appropriate. The effective permission of a tool is determined by the identity it runs as and the operations reachable through its parameters.

Risk condition. A connected MCP tool can perform operations outside the intended application scope, or permissions are not reviewed when server capabilities change.

Expected outcome. MCP tool capabilities and effective permissions are identified, reviewed, and restricted according to the approved system function.

ASSESSMENT EVIDENCE

- MCP server configuration
- Tool definitions
- Authentication configuration
- Effective service permissions
- Allowed operations
- External service access
- Change records

Implementation considerations. Tool permissions should be evaluated as effective permissions, not only as documented tool descriptions. Review the identity used by the MCP server, downstream service permissions, and operations available through parameterized tool calls.

Primary CSF 2.0 mapping	PR.AA-05
Supporting CSF 2.0 mapping	ID.AM-02, PR.PS-01

PNX-LLM-MCP-04 Untrusted MCP Server Connection

Security objective. Establish the trust basis of an MCP server before connecting an application or agent to it.

Connecting to an MCP server extends application capability to whatever that server exposes and to whatever it can reach downstream. Connecting to a server whose origin, integrity, and behavior are not established places that trust in an unverified party.

Risk condition. An application or agent connects to an MCP server whose origin and integrity have not been established, or to a server that can change its exposed capability after connection.

Expected outcome. MCP servers are evaluated before connection, and connections are limited to servers with an established trust basis.

ASSESSMENT EVIDENCE

- Server origin and provenance records
- Integrity verification records
- Connection approval records
- Change monitoring configuration

Implementation considerations. An MCP server can present benign tool descriptions while its implementation reaches further systems. Establish trust from the server's origin and observed behavior.

Primary CSF 2.0 mapping	GV.SC-06
Supporting CSF 2.0 mapping	ID.RA-09

PNX-LLM-MCP-05 MCP Tool Input Manipulation

Security objective. Preserve the integrity of MCP tool invocations against manipulation through model-generated parameters.

MCP tool calls are constructed from model output. When untrusted input influences that output, the parameters passed to a tool can be manipulated to reach unintended operations. The concern is the integrity of the tool invocation, including the operation selected and the parameters supplied.

Risk condition. Model-generated tool parameters are manipulated through untrusted input, causing a tool to perform an unintended operation within its available permissions.

Expected outcome. Tool invocations reflect the intended operation, and manipulation of parameters is constrained and detectable.

ASSESSMENT EVIDENCE

- Tool parameter handling design
- Invocation constraint configuration
- Tool invocation records
- Manipulation test results

Implementation considerations. A tool with broad parameters is manipulable even when its permissions are correct. Constrain the operations and parameter ranges a tool accepts, and record invocations for review.

Primary CSF 2.0 mapping	PR.DS-10
Supporting CSF 2.0 mapping	PR.AA-05, DE.CM-09

PNX-LLM-MCP-06 MCP Capability Change Detection

Security objective. Detect changes to the tools, resources, and permissions an MCP server exposes.

An MCP server's exposed capability can change after connection through server updates, configuration changes, or dynamic tool registration. Capability change detection establishes that the tools and permissions in use match what was reviewed, and surfaces changes for reassessment.

Risk condition. An MCP server changes the tools or permissions it exposes after connection, and the change is not detected or reassessed.

Expected outcome. Changes to MCP-exposed capability are detected, recorded, and reassessed against the approved configuration.

ASSESSMENT EVIDENCE

- Baseline capability records
- Capability change records
- Change monitoring configuration
- Reassessment records

Implementation considerations. Capability change is a monitoring concern as well as a change-management concern. A server that adds a tool has changed the application's execution surface.

Primary CSF 2.0 mapping	DE.CM-09
Supporting CSF 2.0 mapping	ID.RA-07, PR.PS-01

5.7 Agent Security

PNX-LLM-AG-01 Agent Tool Inventory

Security objective. Maintain an inventory of the tools available to each agent and the operations those tools expose.

An agent acts by invoking tools. An agent tool inventory records which tools each agent can invoke and the operations reachable through them, so that the agent's capability can be assessed against its intended function.

Risk condition. Agents are deployed with tool sets that are not inventoried, so the operations an agent can perform are not fully known.

Expected outcome. The tools available to each agent are inventoried and attributed to the operations they expose.

ASSESSMENT EVIDENCE

- Agent tool inventory
- Tool operation records
- Agent-to-tool mapping
- Tool source records

Implementation considerations. Tool sets change as agents are extended. Tie the inventory to the agent configuration that is actually deployed rather than to the intended design.

Primary CSF 2.0 mapping	ID.AM-02
Supporting CSF 2.0 mapping	ID.AM-05, PR.AA-05

PNX-LLM-AG-02 Unsafe Tool Execution

Security objective. Constrain the operations an agent can execute so that consequential actions occur within an approved boundary.

Agent tools can perform file operations, command execution, API actions, administrative operations, database actions, messaging actions, cloud operations, and identity operations. Unsafe tool execution is the ability of an agent to perform a consequential operation without a control appropriate to its impact.

Risk condition. An agent executes a consequential operation without an approval boundary, a constrained identity, an operation allowlist, or an observable record.

Expected outcome. Consequential agent operations occur within a defined boundary, run under a constrained identity, and are observable.

ASSESSMENT EVIDENCE

- Tool execution constraints
- Execution identity records
- Operation allowlists
- Tool execution records

Implementation considerations. Human approval is one boundary among several and is not always the right one. Constrained identities, operation allowlists, and observable execution can bound impact where approval is impractical.

Primary CSF 2.0 mapping	PR.AA-05
Supporting CSF 2.0 mapping	DE.CM-09

PNX-LLM-AG-03 Excessive Agent Permissions

Security objective. Align an agent's granted permissions with the operations its function requires.

Excessive agent permissions arise when the identity an agent uses can perform operations beyond the agent's function. The concern is the gap between granted permission and required permission, which determines the impact of a compromise or a manipulation.

Risk condition. An agent runs with an identity whose permissions exceed the operations its function requires.

Expected outcome. Agent permissions are scoped to the required operations, and excess permission is removed or justified.

ASSESSMENT EVIDENCE

- Agent execution identity records
- Granted permission records
- Required operation records
- Permission review records

Implementation considerations. Excess permission usually comes from reusing a broad service identity. Assess the effective permission of the identity behind the agent.

Primary CSF 2.0 mapping	PR.AA-05
Supporting CSF 2.0 mapping	PR.AA-01, ID.RA-05

PNX-LLM-AG-04 Unbounded Agent Action

Security objective. Bound the volume, scope, and duration of the actions an agent can take without intervention.

An agent can take repeated or expanding actions in response to input or its own intermediate output. Unbounded action is the absence of limits on how many actions, of what scope, and over what duration an agent can take, which affects both security impact and resource consumption.

Risk condition. An agent takes actions without limits on volume, scope, or duration, so a single input can produce a large or expanding set of operations.

Expected outcome. Agent actions are bounded in volume, scope, and duration, and boundary conditions are observable.

ASSESSMENT EVIDENCE

- Action limit configuration
- Scope constraint records
- Agent action records
- Resource consumption records

Implementation considerations. Bounds serve both security and availability. Assess the limits on action count and scope alongside the resource and cost controls.

Primary CSF 2.0 mapping	PR.AA-05
Supporting CSF 2.0 mapping	DE.CM-09, PR.IR-04

PNX-LLM-AG-05 Agent Identity and Authentication

Security objective. Assign and authenticate distinct identities for agents so that their actions are attributable and access-controlled.

An agent that acts on systems does so under an identity. Agent identity and authentication establish that the agent presents a distinct, managed identity to the systems it reaches, so that its access is controlled and its actions are attributable.

Risk condition. An agent acts under a shared, unmanaged, or unauthenticated identity, so its access cannot be controlled and its actions cannot be attributed.

Expected outcome. Agents present distinct, managed identities that are authenticated to the systems they reach.

ASSESSMENT EVIDENCE

- Agent identity records
- Authentication configuration
- Credential management records
- Attribution records

Implementation considerations. A shared identity across agents removes attribution and makes least privilege impractical. Assign distinct identities where the systems support it.

Primary CSF 2.0 mapping	PR.AA-01
Supporting CSF 2.0 mapping	PR.AA-03, PR.AA-05

PNX-LLM-AG-06 Cross-System Agent Attack Path

Security objective. Identify paths where an agent connects an untrusted input to a consequential operation on a downstream system.

A cross-system agent attack path connects an entry point, an LLM application, a content source, an agent, its tools, an execution identity, and a downstream system into a reachable route to a consequential operation. The path analysis records how these elements combine and where impact concentrates.

Risk condition. An untrusted input reaches a consequential operation on a downstream system through an agent and its tools, and the path is not identified.

Expected outcome. Cross-system agent paths are identified and recorded, and paths carrying the greatest impact are prioritized for control.

ASSESSMENT EVIDENCE

- Agent tool and identity graph
- Downstream system reachability records
- Data flow records
- Recorded path analysis

Implementation considerations. Two paths illustrate the pattern. An untrusted retrieved document leads to indirect prompt injection, then to agent instruction manipulation, then to an MCP tool invocation under a service identity, then to an internal ticketing or source control operation. A user message leads to a tool call, then to a cloud operation under the agent identity, then to a configuration change. The impact of each path is set by the identity of its final operation.

Primary CSF 2.0 mapping	ID.RA-04
Supporting CSF 2.0 mapping	PR.AA-05, DE.AE-03

5.8 Monitoring and Detection

PNX-LLM-DE-01 Prompt Attack Monitoring

Security objective. Monitor prompt flows for input that attempts to manipulate application behavior.

Prompt attack monitoring observes the input side of LLM traffic for manipulation attempts, including injection and control-bypass patterns. It provides the runtime signal that a point-in-time test cannot, because prompt content changes continuously in production.

Risk condition. Prompt manipulation attempts occur in production without a monitoring signal, so injection and bypass activity is not observed.

Expected outcome. Prompt flows are monitored for manipulation, and detected attempts are available for analysis.

ASSESSMENT EVIDENCE

- Prompt security event records
- Monitoring configuration
- Detection coverage records
- Model endpoint request records

Implementation considerations. Runtime monitoring does not replace architecture review. It observes what reaches the application; it does not establish whether the application's boundaries are correct.

Primary CSF 2.0 mapping	DE.CM-09
Supporting CSF 2.0 mapping	DE.AE-02, DE.CM-01

PNX-LLM-DE-02 Model Response Security Monitoring

Security objective. Monitor model responses for sensitive data disclosure and policy-violating output.

Model response monitoring observes the output side of LLM traffic for sensitive data disclosure and output that violates policy. It complements input monitoring because some concerns are only visible in the response.

Risk condition. Sensitive or policy-violating output is returned in production without a monitoring signal, so disclosure through responses is not observed.

Expected outcome. Model responses are monitored for disclosure and policy violation, and detected events are available for analysis.

ASSESSMENT EVIDENCE

- Response security event records
- Monitoring configuration
- Output classification records
- Log records

Implementation considerations. Response monitoring must be configured to avoid retaining the sensitive content it detects in a broadly readable store. Configure detection and retention together.

Primary CSF 2.0 mapping	DE.CM-09
Supporting CSF 2.0 mapping	DE.AE-02, PR.PS-04

PNX-LLM-DE-03 AI Configuration Change Monitoring

Security objective. Monitor changes to model, gateway, agent, and RAG configuration that affect the security boundary.

AI configuration change monitoring observes changes to model settings, gateway routing, agent policy, and RAG indexing that alter the security boundary. A configuration change can expand exposure without any change to code.

Risk condition. A change to AI configuration expands the security boundary without detection, so the exposure introduced by the change is not observed or reassessed.

Expected outcome. Security-relevant AI configuration changes are detected, recorded, and available for reassessment.

ASSESSMENT EVIDENCE

- Configuration change records
- Baseline configuration records
- Change monitoring configuration
- Reassessment records

Implementation considerations. Point-in-time testing does not replace change monitoring. A configuration that was assessed as sound can be changed the day after the assessment.

Primary CSF 2.0 mapping	DE.CM-09
Supporting CSF 2.0 mapping	ID.RA-07, PR.PS-01

PNX-LLM-DE-04 MCP Capability Change Monitoring

Security objective. Monitor connected MCP servers for changes to the tools and permissions they expose.

MCP capability change monitoring observes connected servers for changes to their exposed tools, resources, and permissions. Because a server can change its capability after connection, this monitoring maintains an accurate view of the execution surface reaching applications through MCP.

Risk condition. A connected MCP server changes its exposed capability, and the change is not observed against the approved baseline.

Expected outcome. Changes to MCP-exposed capability are monitored and available for reassessment.

ASSESSMENT EVIDENCE

- Baseline capability records
- Capability change events
- External service monitoring records
- Change monitoring configuration

Implementation considerations. MCP capability change monitoring overlaps external service monitoring where the server is operated by a third party. Include both the server behavior and the downstream services it reaches.

Primary CSF 2.0 mapping	DE.CM-09
Supporting CSF 2.0 mapping	DE.CM-06, ID.RA-07

PNX-LLM-DE-05 Agent Execution Monitoring

Security objective. Monitor agent tool execution for actions that fall outside expected operation.

Agent execution monitoring observes the operations agents perform through their tools, including the operation, the identity, and the outcome. It provides the record needed to detect and investigate agent actions that fall outside expected operation.

Risk condition. An agent performs operations that fall outside expected operation without a monitoring signal, so anomalous or unsafe execution is not observed.

Expected outcome. Agent tool execution is monitored, and operations outside expected behavior are available for analysis.

ASSESSMENT EVIDENCE

- Agent action records
- Tool execution records
- Execution identity records
- Monitoring configuration

Implementation considerations. Execution monitoring depends on the agent and its tools producing attributable records. Assess whether the execution path generates the records the monitoring requires.

Primary CSF 2.0 mapping	DE.CM-09
Supporting CSF 2.0 mapping	DE.CM-03, DE.AE-02

PNX-LLM-DE-06 AI Security Event Correlation

Security objective. Correlate AI security events across prompt, response, tool, configuration, and identity sources.

AI security events originate from prompts, responses, tool executions, configuration changes, and identity activity. Correlation combines these sources so that a multi-step sequence, such as a poisoned retrieval followed by a tool invocation, can be understood as a single chain rather than as unrelated events.

Risk condition. AI security events are observed in isolation, so a multi-step sequence that crosses prompt, retrieval, tool, and identity sources is not recognized.

Expected outcome. AI security events are correlated across sources so that related activity is analyzed together.

ASSESSMENT EVIDENCE

- Event source inventory
- Correlation configuration
- Correlated event records
- Analysis records

Implementation considerations. Correlation quality depends on consistent identifiers across sources. Assess whether prompt, tool, and identity events can be joined on a common request or session identifier.

Primary CSF 2.0 mapping	DE.AE-03
Supporting CSF 2.0 mapping	DE.AE-02, DE.AE-04

5.9 Response

PNX-LLM-RS-01 AI Security Event Analysis

Security objective. Analyze AI security events to establish what took place and the root cause.

AI security event analysis establishes what happened during an AI security event and why. It draws on the correlated event record to determine the sequence, the entry point, the systems reached, and the underlying cause.

Risk condition. An AI security event is handled without analysis of what took place and why, so the root cause is not established and recurrence is likely.

Expected outcome. AI security events are analyzed to establish the sequence of activity and the root cause.

ASSESSMENT EVIDENCE

- Correlated event records
- Analysis records
- Root cause determination
- Impact records

Implementation considerations. Analysis depends on the evidence preserved at the time of the event. The quality of the analysis is bounded by the runtime and logging records available.

Primary CSF 2.0 mapping	RS.AN-03
Supporting CSF 2.0 mapping	DE.AE-02, RS.AN-08

PNX-LLM-RS-02 Model Endpoint Isolation

Security objective. Isolate a model endpoint to contain an active AI security event.

Model endpoint isolation removes or restricts access to an endpoint involved in an active event, containing the activity while analysis and remediation proceed. It is a containment action applied to the model interface.

Risk condition. An active event involving a model endpoint continues because there is no ready means to isolate the endpoint.

Expected outcome. A model endpoint involved in an event can be isolated to contain the activity.

ASSESSMENT EVIDENCE

- Endpoint isolation procedure
- Access control records
- Isolation action records
- Containment outcome records

Implementation considerations. Isolation is more effective when the endpoint's callers and dependencies are known in advance. The asset inventory and data flow records support a targeted isolation.

Primary CSF 2.0 mapping	RS.MI-01
Supporting CSF 2.0 mapping	RS.MA-03, PR.AA-05

PNX-LLM-RS-03 MCP Connection Revocation

Security objective. Revoke an MCP connection and the associated credentials to contain an event involving a connected server.

MCP connection revocation removes a server connection and invalidates the credentials associated with it, containing activity that reaches systems through that server. It addresses both the connection and the downstream credentials the server holds.

Risk condition. An event involving a connected MCP server continues because the connection and its credentials cannot be revoked promptly.

Expected outcome. An MCP connection and its associated credentials can be revoked to contain an event.

ASSESSMENT EVIDENCE

- Connection revocation procedure
- Credential inventory
- Revocation action records
- Containment outcome records

Implementation considerations. Revoking the connection without invalidating the downstream credentials the server holds may leave the reachable operations available. Address both.

Primary CSF 2.0 mapping	RS.MI-01
Supporting CSF 2.0 mapping	PR.AA-01, RS.MI-02

PNX-LLM-RS-04 Agent Capability Restriction

Security objective. Restrict an agent's tools or identity to contain an event involving agent activity.

Agent capability restriction reduces the tools available to an agent or constrains the identity it uses, containing activity that the agent is performing. It is a containment action applied to the agent's execution capability.

Risk condition. An event involving an agent continues because there is no ready means to restrict the tools or identity it uses.

Expected outcome. An agent's capability can be restricted to contain an event.

ASSESSMENT EVIDENCE

- Capability restriction procedure
- Agent identity and tool records
- Restriction action records
- Containment outcome records

Implementation considerations. The effectiveness of restriction depends on the agent's identity and tools being known and controllable at the time of the event. The agent inventory supports a targeted restriction.

Primary CSF 2.0 mapping	RS.MI-01
Supporting CSF 2.0 mapping	PR.AA-05, RS.MI-02

PNX-LLM-RS-05 Prompt Security Incident Investigation

Security objective. Investigate a prompt security incident using preserved prompt, response, and execution records.

Prompt security incident investigation reconstructs a prompt-related event from the preserved records of prompts, responses, and the actions that followed. It establishes what input was involved, how the application responded, and what operations resulted.

Risk condition. A prompt security incident cannot be reconstructed because the prompt, response, and execution records needed for investigation were not preserved.

Expected outcome. Prompt security incidents can be investigated from preserved records, and the sequence of activity is established.

ASSESSMENT EVIDENCE

- Preserved prompt and response records
- Execution records
- Investigation records
- Triage and validation records

Implementation considerations. Investigation depends on records that may themselves contain sensitive content. Preserve the records under access control appropriate to their sensitivity.

Primary CSF 2.0 mapping	RS.AN-03
Supporting CSF 2.0 mapping	RS.AN-06, RS.MA-02

PNX-LLM-RS-06 AI Security Evidence Preservation

Security objective. Preserve AI security event data and metadata with integrity and provenance for analysis and investigation.

AI security evidence preservation retains the prompt, response, tool execution, configuration, and identity records associated with an event, along with their metadata, so that analysis and investigation can rely on them. Preservation addresses both the completeness and the integrity of the records.

Risk condition. The records needed to analyze and investigate an AI security event are not preserved, or their integrity and provenance are not maintained.

Expected outcome. AI security event data and metadata are collected and preserved with integrity and provenance.

ASSESSMENT EVIDENCE

- Evidence collection procedure
- Record integrity controls
- Provenance records
- Retention configuration

Implementation considerations. Evidence quality matters when the mapping is used for control review as well as for incident response. Records that lack integrity or provenance limit both analysis and assurance.

Primary CSF 2.0 mapping	RS.AN-07
Supporting CSF 2.0 mapping	RS.AN-06, PR.PS-04

6 LLM Security Risk Areas

6.1 AI Asset and Service Visibility

An organization cannot assess LLM exposure without identifying the systems, endpoints, data stores, and external services involved. AI asset and service visibility records the LLM applications, hosted model APIs, self-hosted models, cloud AI services, model gateways, AI application endpoints, agent runtimes, RAG infrastructure, MCP servers, and external AI dependencies in use, together with their ownership, environment, data classification context, and authentication boundary.

No single discovery method provides complete visibility. DNS and network data expose reachable endpoints. A configuration management database records declared systems. Cloud service inventory reveals managed AI services. Source code dependencies expose SDKs and frameworks. API gateway configuration reveals routed endpoints. Application configuration reveals provider integrations. Each source describes a different part of the surface, and the inventory is a reconciliation of several sources.

Ownership and data classification context belong to the inventory itself. An endpoint without an owner cannot be assessed or remediated, and an application whose data classification is unknown cannot be prioritized.

6.2 Model Endpoint Exposure

A model endpoint is the interface through which prompts are submitted and responses returned. Exposure describes the conditions under which an endpoint can be reached and used: public and internal reachability, the presence and strength of authentication, and the presence of test, development, or administrative endpoints that were not intended for production traffic.

Model gateways concentrate exposure. A gateway that fronts several models presents one reachable interface whose access boundary governs access to all of them. Test and development endpoints are a recurring source of exposure because they are often created without the access controls applied to production and are not removed when their purpose ends.

> Rate control limits abuse volume but does not establish an access boundary. Treat quotas and rate limits as supporting evidence alongside a real access control.

6.3 Shadow AI

Shadow AI is LLM and generative AI usage that is not recorded in the approved inventory. It is defined by the absence of a security and governance record rather than by the identity of the service. The category includes unapproved model APIs, embedded SDKs, browser-based AI services, SaaS AI features, application plugins, developer tooling, undocumented model integrations, and external endpoints referenced in code or configuration.

A precise definition keeps the assessment focused. Treating every use of a public assistant as shadow AI produces noise that obscures the cases that matter: integrations that process organizational data or connect to or-

ganizational systems. Those integrations expand the attack surface and the data exposure, and they are the ones an inventory needs to capture.

6.4 Prompt Injection

Prompt injection is the manipulation of application behavior through input that the model treats as instruction. It divides into direct injection, where the input comes from a user of the application, and indirect injection, where the input is embedded in content the model reads, such as retrieved documents, web content, files, or tool outputs.

The relevant boundary is between application-controlled context and other input. An LLM does not enforce an instruction hierarchy on its own, so the application cannot assume that its instructions take precedence over user text or over retrieved content. Content that the application places into the model's context becomes an input trust boundary, whether it originates from a user, a document, or a tool.

The consequence of injection depends on what the application can do once its behavior is altered. An application that only returns text has a bounded exposure. An application with an agent and tools can be driven to take actions, which is where injection connects to tool execution and attack path analysis.

> Pattern filtering removes a subset of known phrasings and does not establish the boundary. Treat it as one supporting control alongside context isolation, output handling, and constrained execution.

6.5 System Prompt Exposure

System prompt exposure covers explicit extraction, partial disclosure, behavioral inference, error paths, logging, debugging interfaces, downstream tool context, and multi-turn leakage. The system prompt can be disclosed directly or reconstructed from the model's behavior across interactions.

Preventing literal disclosure of the system prompt and protecting security-relevant application behavior are related but not identical objectives. A design that prevents extraction of the prompt text still carries risk if the prompt contains a secret, a credential, or a reference whose value does not depend on the exact wording. Where the design allows, security-relevant enforcement belongs outside the prompt, so that the security of the application does not rest on the secrecy of a string.

6.6 Sensitive Data in Prompt and Response Flows

Sensitive data enters and leaves LLM systems through prompts and responses. On the input side it arrives through user input, application context, and retrieved content, and can include PII, credentials, API keys, internal documents, source code, customer data, regulated data, security telemetry, and incident data. On the output side it can be disclosed in responses. In both directions it can be retained in prompt and response logs and processed by an external model provider.

The applicable controls depend on the destination. Data sent to a self-hosted model stays within the organization's boundary. Data sent to an external provider crosses that boundary and is governed by the provider's processing and retention terms. Retention in logs creates data-at-rest whose sensitivity matches the original content and which often outlives the application that produced it.

This resource does not provide legal conclusions about any data type. It records the data flows and retention points so that an organization can apply its own classification and obligations.

6.7 RAG Security

A RAG system answers using content retrieved from indexed sources. Its security spans the data source inventory, the access controls on those sources, the retrieval scope, the indexing pipeline, document provenance, vector store access, chunk metadata, tenant boundaries, retrieval authorization, and the context that reaches the generated answer.

Retrieval authorization is the control that most often fails quietly. When authorization is enforced at the application boundary rather than at retrieval, and retrieval runs with a broad service identity, a user can receive content through generated answers that they could not access directly. Assessing retrieval requires the identity that retrieval executes as, not only the application's front-door controls.

6.8 RAG Poisoning

RAG poisoning is the introduction of malicious or misleading content into a retrieval source or index. It occurs through malicious source content, a compromised data source, unauthorized document insertion, or the exploitation of a trusted-source assumption. Poisoned content can carry indirect prompt injection into every interaction that retrieves it, and it can bias generated answers without any change to the model or the application.

Provenance and change monitoring are central because poisoning usually exploits a source the pipeline already trusts. Content that is stale or has been replaced can be as damaging as content that is malicious. Assessing poisoning requires integrity at the source, at insertion, and at re-indexing, and a means to detect unauthorized change.

6.9 MCP Server Security

The Model Context Protocol connects LLM applications and agents to servers that expose tools, resources, and prompts. An MCP server is a security boundary and an integration point. Its security covers the server inventory, its origin, authentication, transport exposure, the tools and resources it exposes, any prompts it provides, the downstream credentials it holds, its effective service identity, its external network access, and changes to its capability.

The credentials an MCP server holds for downstream services are frequently more sensitive than the connection to the server. A server that reaches a source control system, a ticketing system, or a cloud API does so under an identity whose permissions determine the impact of a compromise. Assessing an MCP server means assessing both directions: the connection to it and the systems it can reach.

6.10 MCP Tool Permission Exposure

A tool's declared behavior and its effective downstream permission are different properties. A tool described as reading a record may run under a service identity that can also modify or delete records, and a tool with a narrow description may accept parameters that reach high-impact operations. The declared behavior is what the tool is meant to do; the effective permission is what it can do.

A benign tool description therefore does not establish the execution boundary. The boundary is set by the identity the tool runs as and by the operations reachable through its parameters. Assessing tool permission means evaluating effective permission: the service identity, the downstream permissions of that identity, and the operations available through parameterized calls.

6.11 Unsafe Agent Tool Execution

Agent tools can perform file operations, shell or command execution, API actions, administrative operations, database actions, messaging actions, cloud operations, and identity operations. Unsafe tool execution is the ability of an agent to perform a consequential operation without a control appropriate to its impact.

Several controls can bound impact: an approval boundary for the highest-impact operations, a constrained identity that cannot exceed the intended scope, an operation allowlist that limits reachable actions, and observable execution that records what was done. These controls are not equivalent, and human approval is not always the right one. Approval fits operations that are rare and high-impact; for frequent operations, constrained identities and allowlists bound impact without making the agent impractical.

6.12 AI Agent Attack Paths

Attack path analysis treats an agent deployment as a chain: an entry point, an LLM application, a prompt or content source, an agent, its available tools, the effective identity those tools use, a downstream system, and the data or administrative capability reachable there. A path is a security concern when it connects an untrusted input to a consequential operation.

Two neutral examples illustrate the pattern. In the first, an untrusted retrieved document carries an indirect prompt injection, which manipulates an agent's instructions, which leads to an MCP tool invocation, which runs under a service account permission, which reaches an internal ticketing or source control operation. In the second, a user message produces a tool call that performs a cloud operation under the agent's identity, which changes a configuration. In each case the impact is set by the identity of the final operation, which is why effective-permission data is central to the analysis.

6.13 Model and AI Supply Chain Risk

An LLM system is composed from model providers, model artifacts, SDKs, agent frameworks, MCP servers, container images, dependencies, plugins, model gateway dependencies, hosted AI services, and CI/CD configuration. Each is a component whose security and whose ownership contribute to the system's risk.

This section is specific to AI system composition rather than to general software supply chain risk. The components that distinguish it are the ones that execute or influence application behavior: a model artifact that runs, an agent framework that orchestrates tool calls, and an MCP server that reaches downstream systems. For these, the security question is not only provenance and integrity but also who owns the security of the component in operation.

7 NIST CSF 2.0 Mapping

The table relates each taxonomy entry to its primary and supporting NIST CSF 2.0 outcomes. Primary and supporting relationships are defined in the methodology. This is a community mapping and not an official NIST crosswalk. Full outcome wording and per-relationship rationales are given in the accompanying datasets.

Taxonomy ID	Security Concern	Category	Primary CSF 2.0	Supporting CSF 2.0
PNX-LLM-AM-01	AI Asset Inventory	Asset Management	ID.AM-02	ID.AM-05, ID.AM-08
PNX-LLM-AM-02	Model Endpoint Discovery	Asset Management	ID.AM-02	ID.AM-03, ID.RA-01
PNX-LLM-AM-03	External Model Service Identification	Asset Management	ID.AM-04	GV.SC-04, ID.AM-02
PNX-LLM-AM-04	Shadow AI Identification	Asset Management	ID.AM-02	DE.CM-09, ID.AM-04
PNX-LLM-AM-05	AI Dependency Inventory	Asset Management	ID.AM-02	GV.SC-04, ID.RA-09
PNX-LLM-RA-01	LLM Application Risk Assessment	Risk Assessment	ID.RA-05	ID.RA-03, ID.RA-04
PNX-LLM-RA-02	Model Endpoint Exposure	Risk Assessment	ID.RA-01	PR.AA-05, PR.IR-01
PNX-LLM-RA-03	AI Attack Path Analysis	Risk Assessment	ID.RA-04	ID.RA-03, ID.RA-05
PNX-LLM-RA-04	Agent Capability Risk	Risk Assessment	ID.RA-05	PR.AA-05, ID.RA-04
PNX-LLM-RA-05	AI Supply Chain Risk	Risk Assessment	GV.SC-07	ID.RA-09, GV.SC-04
PNX-LLM-PR-01	Direct Prompt Injection	Prompt Security	PR.DS-10	ID.RA-01, DE.CM-09
PNX-LLM-PR-02	Indirect Prompt Injection	Prompt Security	PR.DS-10	PR.DS-01, DE.CM-09
PNX-LLM-PR-03	System Prompt Exposure	Prompt Security	PR.DS-10	ID.RA-01, DE.CM-09
PNX-LLM-PR-04	Prompt Control Bypass	Prompt Security	PR.DS-10	PR.PS-06, ID.IM-02
PNX-LLM-PR-05	Cross-Context Prompt Manipulation	Prompt Security	PR.DS-10	PR.AA-05, DE.CM-09
PNX-LLM-DS-01	Sensitive Data in Prompts	Data Security	PR.DS-02	PR.DS-10, ID.AM-07

Taxonomy ID	Security Concern	Category	Primary CSF 2.0	Supporting CSF 2.0
PNX-LLM-DS-02	Sensitive Data in Model Responses	Data Security	PR.DS-02	PR.DS-10, DE.CM-09
PNX-LLM-DS-03	Prompt and Response Retention Exposure	Data Security	PR.DS-01	PR.AA-05, ID.AM-07
PNX-LLM-DS-04	External Model Data Transfer	Data Security	PR.DS-02	GV.SC-05, GV.SC-07
PNX-LLM-DS-05	AI Logging Data Exposure	Data Security	PR.DS-01	PR.PS-04, PR.AA-05
PNX-LLM-RAG-01	RAG Data Source Inventory	RAG Security	ID.AM-07	ID.AM-02, ID.AM-05
PNX-LLM-RAG-02	Retrieval Authorization Exposure	RAG Security	PR.AA-05	PR.DS-01, ID.AM-05
PNX-LLM-RAG-03	RAG Poisoning	RAG Security	PR.DS-01	DE.CM-09, PR.DS-10
PNX-LLM-RAG-04	Untrusted Retrieved Content	RAG Security	PR.DS-10	PR.DS-01, DE.CM-09
PNX-LLM-RAG-05	Vector Store Exposure	RAG Security	PR.DS-01	PR.AA-05, PR.IR-01
PNX-LLM-MCP-01	MCP Server Inventory	MCP Security	ID.AM-02	ID.AM-04, ID.AM-05
PNX-LLM-MCP-02	MCP Server Authentication	MCP Security	PR.AA-03	PR.AA-01, PR.AA-04
PNX-LLM-MCP-03	MCP Tool Permission Exposure	MCP Security	PR.AA-05	ID.AM-02, PR.PS-01
PNX-LLM-MCP-04	Untrusted MCP Server Connection	MCP Security	GV.SC-06	ID.RA-09
PNX-LLM-MCP-05	MCP Tool Input Manipulation	MCP Security	PR.DS-10	PR.AA-05, DE.CM-09
PNX-LLM-MCP-06	MCP Capability Change Detection	MCP Security	DE.CM-09	ID.RA-07, PR.PS-01
PNX-LLM-AG-01	Agent Tool Inventory	Agent Security	ID.AM-02	ID.AM-05, PR.AA-05
PNX-LLM-AG-02	Unsafe Tool Execution	Agent Security	PR.AA-05	DE.CM-09
PNX-LLM-AG-03	Excessive Agent Permissions	Agent Security	PR.AA-05	PR.AA-01, ID.RA-05
PNX-LLM-AG-04	Unbounded Agent Action	Agent Security	PR.AA-05	DE.CM-09, PR.IR-04

Taxonomy ID	Security Concern	Category	Primary CSF 2.0	Supporting CSF 2.0
PNX-LLM-AG-05	Agent Identity and Authentication	Agent Security	PR.AA-01	PR.AA-03, PR.AA-05
PNX-LLM-AG-06	Cross-System Agent Attack Path	Agent Security	ID.RA-04	PR.AA-05, DE.AE-03
PNX-LLM-DE-01	Prompt Attack Monitoring	Monitoring and Detection	DE.CM-09	DE.AE-02, DE.CM-01
PNX-LLM-DE-02	Model Response Security Monitoring	Monitoring and Detection	DE.CM-09	DE.AE-02, PR.PS-04
PNX-LLM-DE-03	AI Configuration Change Monitoring	Monitoring and Detection	DE.CM-09	ID.RA-07, PR.PS-01
PNX-LLM-DE-04	MCP Capability Change Monitoring	Monitoring and Detection	DE.CM-09	DE.CM-06, ID.RA-07
PNX-LLM-DE-05	Agent Execution Monitoring	Monitoring and Detection	DE.CM-09	DE.CM-03, DE.AE-02
PNX-LLM-DE-06	AI Security Event Correlation	Monitoring and Detection	DE.AE-03	DE.AE-02, DE.AE-04
PNX-LLM-RS-01	AI Security Event Analysis	Response	RS.AN-03	DE.AE-02, RS.AN-08
PNX-LLM-RS-02	Model Endpoint Isolation	Response	RS.MI-01	RS.MA-03, PR.AA-05
PNX-LLM-RS-03	MCP Connection Revocation	Response	RS.MI-01	PR.AA-01, RS.MI-02
PNX-LLM-RS-04	Agent Capability Restriction	Response	RS.MI-01	PR.AA-05, RS.MI-02
PNX-LLM-RS-05	Prompt Security Incident Investigation	Response	RS.AN-03	RS.AN-06, RS.MA-02
PNX-LLM-RS-06	AI Security Evidence Preservation	Response	RS.AN-07	RS.AN-06, PR.PS-04

8 Assessment Evidence

The mapping is only useful for control review when it is supported by evidence. The evidence model below groups the records that demonstrate the outcomes in the taxonomy. The categories are a way to organize evidence collection; the specific records vary by architecture. No category requires a single tool, and no single tool produces every category.

Evidence Category	Description	Example Types	Related IDs
Inventory Evidence	Records that establish which AI systems, endpoints, tools, and external services exist and who owns them.	Model endpoint inventory; AI application inventory; MCP server inventory; Agent tool inventory; External model provider inventory; RAG data source inventory	PNX-LLM-AM-01, PNX-LLM-AM-02, PNX-LLM-AM-03, PNX-LLM-AM-05, PNX-LLM-RAG-01, PNX-LLM-MCP-01, PNX-LLM-AG-01
Configuration Evidence	Configuration that determines the security boundary of an AI system, independent of its code.	Model gateway configuration; MCP server configuration; RAG indexing configuration; Agent policy configuration; Prompt gateway configuration	PNX-LLM-RA-02, PNX-LLM-MCP-03, PNX-LLM-MCP-06, PNX-LLM-DE-03, PNX-LLM-AG-04
Identity Evidence	Records of the identities under which AI systems, tools, and agents act, used to assess effective permission.	Service principals; API identities; Cloud roles; MCP service identities; Agent execution identities	PNX-LLM-MCP-02, PNX-LLM-MCP-03, PNX-LLM-AG-03, PNX-LLM-AG-05, PNX-LLM-RAG-02
Network Evidence	Records of where AI endpoints and services are reachable from and where organizational data flows.	Endpoint reachability records; Network data flow records; External destination records; Transport configuration	PNX-LLM-AM-02, PNX-LLM-RA-02, PNX-LLM-DS-04, PNX-LLM-RAG-05
Data Flow Evidence	Records of the data that enters prompts, returns in responses, and moves through retrieval and external transfer.	Prompt data flow records; Response data flow records; Retrieval flow records; Data classification per destination	PNX-LLM-DS-01, PNX-LLM-DS-02, PNX-LLM-DS-04, PNX-LLM-RAG-02, PNX-LLM-PR-05
Runtime Evidence	Records generated during operation that support detection, correlation, and investigation.	Prompt and response security events; Tool execution records; Model endpoint requests; Agent action history; MCP tool invocation history	PNX-LLM-DE-01, PNX-LLM-DE-02, PNX-LLM-DE-05, PNX-LLM-MCP-05, PNX-LLM-AG-02
Logging Evidence	Log records generated and made available for monitoring, configured to support detection without exposing sensitive content.	Application log records; Model gateway logs; Agent execution logs; Audit records	PNX-LLM-DS-03, PNX-LLM-DS-05, PNX-LLM-DE-02, PNX-LLM-RS-06
Change Evidence	Records of changes to AI configuration, capability, and dependencies used to detect boundary changes and drift.	Configuration change records; Capability change records; Dependency version changes; Baseline records	PNX-LLM-DE-03, PNX-LLM-DE-04, PNX-LLM-MCP-06, PNX-LLM-RAG-03

Evidence Category	Description	Example Types	Related IDs
Dependency Evidence	Records of the components and suppliers that compose AI systems, used to assess supply chain risk and integrity.	Dependency manifests; Model artifact provenance; SDK and framework versions; Container image contents; Supplier assessment records	`PNX-LLM-AM-05, PNX-LLM-RA-05, PNX-LLM-MCP-04`

9 Implementation Considerations

Organizations will implement the same CSF outcomes differently depending on architecture. The considerations below apply across the taxonomy.

- SaaS model use and self-hosted model use have different evidence sources. A provider relationship supplies contract and processing terms; a self-hosted deployment supplies configuration and runtime records.
- Runtime monitoring does not replace architecture review. Monitoring observes what reaches a system; it does not establish whether the system's boundaries are correct.
- Point-in-time testing does not replace change monitoring. A configuration assessed as sound can be changed after the assessment.
- Asset discovery does not by itself establish ownership. An inventory without owners cannot drive remediation.
- Tool inventory does not establish effective permission boundaries. A tool's permission is set by the identity it runs as.
- Prompt filtering is not a complete prompt injection control. It removes known phrasings and must be combined with context isolation, output handling, and constrained execution.
- RAG data controls must be evaluated at source, indexing, retrieval, and response. A control at one stage does not cover the others.
- AI agents should be assessed by effective execution capability rather than by stated purpose.
- Evidence quality determines how far the mapping can support a control review. Records that lack integrity or provenance limit both assurance and investigation.

10 Limitations

- This is not an official NIST publication and not an official NIST crosswalk.
- This resource does not provide NIST certification and does not modify NIST CSF 2.0.
- Mapping relationships require organizational context to apply. The right relationship for a given system depends on its architecture and risk.
- The taxonomy is focused on LLM and generative AI system security and does not cover the full breadth of AI or machine learning risk.
- The resource does not replace threat modeling. It provides a lens rather than a threat model for a specific system.
- The resource does not replace legal or regulatory analysis and does not provide legal conclusions.
- The resource does not define model safety or model quality evaluation methodology.
- The taxonomy and mapping may change as LLM integration patterns and security boundaries change.

11 Version History

Version	Date	Change
1.0	2026-07-13	Initial publication of the LLM security taxonomy and NIST CSF 2.0 mapping.

12 References

[1] **The NIST Cybersecurity Framework (CSF) 2.0**. National Institute of Standards and Technology. NIST CSWP 29. https://doi.org/10.6028/NIST.CSWP.29. Source of truth for the CSF 2.0 Functions, Categories, Subcategories, and outcome wording used in this mapping.

[2] **NIST Cybersecurity Framework 2.0 Reference Tool**. National Institute of Standards and Technology. https://csrc.nist.gov/projects/cybersecurity-framework/filters. Searchable reference for CSF 2.0 Core identifiers.

[3] **Cybersecurity Framework Profile for Artificial Intelligence (Cyber AI Profile)**. National Institute of Standards and Technology. NIST IR 8596, Initial Preliminary Draft, December 2025. https://csrc.nist.gov/pubs/ir/8596/iprd. NIST's preliminary draft profile that organizes AI cybersecurity risk using NIST CSF 2.0 outcomes. Used as a cross-check reference point for the CSF 2.0 relationships in this resource. As an initial preliminary draft, it is subject to change.

[4] **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. Referenced for generative AI risk context. Not modified by this resource.

[5] **OWASP Top 10 for Large Language Model Applications**. OWASP Foundation. https://genai.owasp.org/. Referenced for LLM application risk categories.

[6] **MITRE ATLAS**. MITRE Corporation. https://atlas.mitre.org/. Referenced for adversarial machine learning tactics and techniques.

[7] **Model Context Protocol Specification**. Model Context Protocol. https://modelcontextprotocol.io/. Referenced for MCP server, tool, and transport definitions.

13 Citation

Penaxtra Security Research. "Open LLM Security Posture Mapping for NIST CSF 2.0." Version 1.0. Penaxtra. https://penaxtra.com/research/llm-security-posture-mapping

This resource is an independent community-developed mapping maintained by Penaxtra Security Research. Penaxtra is not affiliated with or endorsed by the National Institute of Standards and Technology. References to NIST and the NIST Cybersecurity Framework are provided for mapping and implementation purposes. This resource does not modify the NIST Cybersecurity Framework 2.0 and does not create NIST requirements or certification.