MYTHOS TECHNICAL LIBRARY

STANDARDS & BENCHMARKS / MYTHOS-AI

AI Alignment, Red-Team, and Sycophancy Evaluation Standard

Defines adversarial, alignment, deception, overagreement, misuse, control, monitoring, and incident evaluations.

MYTHOS-AI-0006

Candidate Mythos Standard / 2.0.0-candidate

Published 2026-07-26 | Review 2026-10-26 | Public

Mythos Systems

Permanent Identity and Edition Record

Permanent document ID	MYTHOS-AI-0006
Canonical title	AI Alignment, Red-Team, and Sycophancy Evaluation Standard
Publication class	Standards & Benchmarks
Primary domain	MYTHOS-AI - Artificial Intelligence & Agentic Systems
Version	2.0.0-candidate
Status	Candidate Mythos Standard
Publication date	2026-07-26
Scheduled review	2026-10-26
Publisher	Mythos Systems
Owner	Aaron Stovall
Classification	Public
Prior edition	1.0.0-candidate / 2026-07-24
Supersession	This edition supersedes the prior candidate while preserving the permanent identity and historical source.

Identity doctrine

The permanent document ID is immutable. Production sequence labels are not part of the public identity. Atomic standards are authoritative; portfolios, workbooks, and machine-readable exports are generated views.

Normative Language and Applicability

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL indicate the strength of provisions in this Mythos standard. The publication is a Mythos-authored candidate standard and does not claim approval by the cited external authorities.

Applicability rule

Every control is applicable unless the organization documents a specific architectural or lifecycle reason that the subject is absent. N/A decisions MUST name the decision owner, evidence, affected scope, review date, and triggers that invalidate the exclusion.

Conformance evidence hierarchy

Direct deterministic proof: schemas, tests, signatures, state reconciliation, access decisions, build or runtime evidence.

Reproducible technical evidence: benchmark fixtures, traces, artifacts, profiles, logs, metrics, and environment manifests.

Independent assessment: qualified human or separate evaluator using an explicit rubric and disclosed uncertainty.

Attestation or supplier claim: informative unless independently verified and current for the deployed configuration.

Demonstration or narrative: insufficient by itself for a conforming score above 2.

Critical-control doctrine

Critical controls cannot be averaged away. A failed or stale critical control constrains the maximum conformance level regardless of the aggregate score.

Scope, Audience, and Engineering Principles

Purpose. Defines adversarial, alignment, deception, overagreement, misuse, control, monitoring, and incident evaluations.

In-scope systems. models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions.

Primary audience. AI safety, security, red-team, evaluation, product, policy, domain, and governance teams.

Required engineering principles

Model capability and complete system behavior MUST be measured separately.

Human, legal, production, financial, identity, and governance authority MUST remain external to model confidence or conversational fluency.

Evaluation MUST include expected, prohibited, adversarial, degraded, recovery, and lifecycle-change conditions.

Source authority, data rights, permissions, versions, and freshness MUST be visible in evidence.

Critical outcomes MUST be supported by deterministic validators or independent review wherever feasible.

The organization MUST preserve rollback, revocation, correction, deletion, and retirement paths.

Optimization for score, latency, throughput, tokens, or cost MUST NOT hide safety, quality, privacy, accessibility, or reliability regressions.

Exclusions

This standard does not certify a model, provider, framework, benchmark, dataset, or product. Conformance is limited to the declared system, use case, version, environment, evidence period, and assessment scope.

Conformance and Scoring Model

Level	Weighted threshold	Critical-control condition	Meaning
No conformance	< 50	Any state	Materially incomplete, unverified, or unsafe.
Foundation	>= 50	No critical control scored 0	Defined baseline with partial implementation and evidence.
Operational	>= 65	All applicable critical controls >= 3	Implemented and operationally tested.
Assured	>= 80	All applicable critical controls >= 4	Independently verified with adversarial and recovery evidence.
Continuously Assured	>= 90	All applicable critical controls = 5	Continuous regression, monitoring, freshness, and incident learning.

Weighted score

For applicable controls: weighted contribution = (control score / 5) x control weight. N/A controls are removed from the denominator only after an approved applicability decision. The final score is normalized to 100.

A conformance claim MUST include scope, system and model versions, assessment date, evidence window, evaluator identity, scores by family, critical-control status, unresolved findings, exceptions, limitations, and next review.

Control Score Interpretation

Score	Maturity	Required evidence state
0	Not implemented	Not implemented: the assessed control is absent, materially unknown, or contradicted by evidence.
1	Initial	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Nonconformity classes

Critical: credible risk of serious harm, unauthorized high-impact action, material security or privacy failure, false assurance, or inability to recover.

Major: requirement absent or ineffective across a meaningful part of scope, with no sufficient compensating control.

Minor: localized or low-impact deficiency that does not invalidate the overall control objective.

Observation: improvement opportunity or emerging risk without current nonconformity.

Score validity

A score expires when evidence exceeds its approved freshness period or a material change invalidates the assessment. Current operation is not inferred from a historic assessment.

Standard Architecture and Control Model

01	**Govern**	Scope, ownership, authority, policy, impacts, risk tolerance, exceptions, and lifecycle decisions.
02	**Map**	System boundaries, actors, models, data, prompts, tools, interfaces, populations, dependencies, and failure domains.
03	**Measure**	Representative and hidden evaluation, deterministic validation, human or model judging, uncertainty, performance, and cost.
04	**Manage**	Release gates, least privilege, monitoring, incident response, correction, rollback, revocation, deletion, and retirement.
05	**Prove**	Traceable evidence bundles connecting claims to configurations, tests, artifacts, effects, decisions, and user outcomes.

Assessment unit

The assessment unit is the declared AI system or workflow, not the model name alone. It includes prompts, retrieval, memory, tools, policies, interface, evaluators, infrastructure, human oversight, and operating environment.

Benchmark Dimensions and Weights

Dimension	Weight	Assessment emphasis
Hazards and authority	18%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for hazards and authority.
Truthfulness and sycophancy	16%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for truthfulness and sycophancy.
Goal and reward integrity	16%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for goal and reward integrity.
Adversarial and misuse resistance	22%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for adversarial and misuse resistance.
Control and oversight	15%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for control and oversight.
Monitoring and response	13%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for monitoring and response.

Benchmark scores MUST be reported by dimension and by critical-control status. A single aggregate ranking MUST NOT be used to conceal weak safety, authority, privacy, recovery, or evidence performance.

Contents and Control Index

Permanent identity and edition record
Normative language and applicability
Scope, audience, and engineering principles
Conformance and scoring model
Standard architecture and control model
Benchmark dimensions and weights
Control summary matrix

MYTHOS-AI-0006-C01 - Safety Objective and Hazard Contract
MYTHOS-AI-0006-C02 - Instruction Hierarchy and Authority Resolution
MYTHOS-AI-0006-C03 - Goal Integrity and Delegation Boundaries
MYTHOS-AI-0006-C04 - Impact and Autonomy Classification
MYTHOS-AI-0006-C05 - Truthfulness and Evidence Preference
MYTHOS-AI-0006-C06 - Sycophancy and User-Belief Manipulation
MYTHOS-AI-0006-C07 - Calibration, Abstention, and Epistemic Limits
MYTHOS-AI-0006-C08 - Reward Hacking and Specification Gaming
MYTHOS-AI-0006-C09 - Deception and Evaluation-Aware Behavior
MYTHOS-AI-0006-C10 - Misuse and Harmful Capability Evaluation
MYTHOS-AI-0006-C11 - Prompt Injection and Context Manipulation
MYTHOS-AI-0006-C12 - Tool Misuse and Excessive Agency
MYTHOS-AI-0006-C13 - Memory Poisoning and Persistence Abuse
MYTHOS-AI-0006-C14 - Layered Control and Safety Kernel
MYTHOS-AI-0006-C15 - Human Oversight, Approval, and Appeal
MYTHOS-AI-0006-C16 - Red-Team Program and Adaptive Campaigns
MYTHOS-AI-0006-C17 - Evaluator and Judge Independence
MYTHOS-AI-0006-C18 - Safety Monitoring and Drift Detection
MYTHOS-AI-0006-C19 - Safety Incident Response and Reassessment

Control Summary Matrix

Control	Requirement	Family	Wt.	Crit.
C01	Safety Objective and Hazard Contract	Hazards and authority	6	YES
C02	Instruction Hierarchy and Authority Resolution	Hazards and authority	6	YES
C03	Goal Integrity and Delegation Boundaries	Hazards and authority	6	YES
C04	Impact and Autonomy Classification	Truthfulness and sycophancy	6	-
C05	Truthfulness and Evidence Preference	Truthfulness and sycophancy	5	YES
C06	Sycophancy and User-Belief Manipulation	Truthfulness and sycophancy	5	YES
C07	Calibration, Abstention, and Epistemic Limits	Goal and reward integrity	6	-
C08	Reward Hacking and Specification Gaming	Goal and reward integrity	5	YES
C09	Deception and Evaluation-Aware Behavior	Goal and reward integrity	5	YES
C10	Misuse and Harmful Capability Evaluation	Adversarial and misuse resistance	6	YES

Control Summary Matrix - Continued

Control	Requirement	Family	Wt.	Crit.
C11	Prompt Injection and Context Manipulation	Adversarial and misuse resistance	6	YES
C12	Tool Misuse and Excessive Agency	Adversarial and misuse resistance	5	YES
C13	Memory Poisoning and Persistence Abuse	Adversarial and misuse resistance	5	YES
C14	Layered Control and Safety Kernel	Control and oversight	5	YES
C15	Human Oversight, Approval, and Appeal	Control and oversight	5	YES
C16	Red-Team Program and Adaptive Campaigns	Control and oversight	5	YES
C17	Evaluator and Judge Independence	Monitoring and response	5	YES
C18	Safety Monitoring and Drift Detection	Monitoring and response	4	-
C19	Safety Incident Response and Reassessment	Monitoring and response	4	YES

Weight integrity

The 19 applicable control weights sum to 100. Family weights match the benchmark dimension model. Critical controls are highlighted in the atomic control pages and assessment workbook.

MYTHOS-AI-0006-C01

Safety Objective and Hazard Contract

Family: Hazards and authority | Weight: 6 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain safety objective and hazard contract for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The operating contract must define intended users, decisions, prohibited uses, affected systems, consequences, risk tolerance, terminal conditions, and accountable authority. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that safety objective and hazard contract is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for safety objective and hazard contract covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.
- The operating contract must define intended users, decisions, prohibited uses, affected systems, consequences, risk tolerance, terminal conditions, and accountable authority.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C01

Safety Objective and Hazard Contract

Family: Hazards and authority | Weight: 6 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for safety objective and hazard contract.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- adversarial case corpus with campaign provenance
- critical-failure findings, control response, and retest evidence

Required metrics

- attack success rate
- critical control bypass rate
- safe abstention or containment rate
- time to detection
- retest closure rate

Score anchors

0	Not implemented: Safety Objective and Hazard Contract is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for safety objective and hazard contract.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C02

Instruction Hierarchy and Authority Resolution

Family: Hazards and authority | Weight: 6 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain instruction hierarchy and authority resolution for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. Authority must be enforceable outside the model through stable identities, least privilege, scoped credentials, separation of duties, revocation, and auditable approval. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that instruction hierarchy and authority resolution is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for instruction hierarchy and authority resolution covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.

Authority must be enforceable outside the model through stable identities, least privilege, scoped credentials, separation of duties, revocation, and auditable approval.

Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.

Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.

Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.

Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C02

Instruction Hierarchy and Authority Resolution

Family: Hazards and authority | Weight: 6 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for instruction hierarchy and authority resolution.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Instruction Hierarchy and Authority Resolution is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for instruction hierarchy and authority resolution.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C03

Goal Integrity and Delegation Boundaries

Family: Hazards and authority | Weight: 6 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain goal integrity and delegation boundaries for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that goal integrity and delegation boundaries is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for goal integrity and delegation boundaries covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.

The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers.

Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.

Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.

Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.

Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C03

Goal Integrity and Delegation Boundaries

Family: Hazards and authority | Weight: 6 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for goal integrity and delegation boundaries.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Goal Integrity and Delegation Boundaries is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for goal integrity and delegation boundaries.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C04

Impact and Autonomy Classification

Family: Truthfulness and sycophancy | Weight: 6 | Critical: NO

Normative requirement

The organization MUST establish, implement, verify, and maintain impact and autonomy classification for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that impact and autonomy classification is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for impact and autonomy classification covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.
- The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C04

Impact and Autonomy Classification

Family: Truthfulness and sycophancy | Weight: 6 | Critical: NO

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for impact and autonomy classification.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Impact and Autonomy Classification is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for impact and autonomy classification.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C05

Truthfulness and Evidence Preference

Family: Truthfulness and sycophancy | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain truthfulness and evidence preference for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. Evidence must correlate system identity, version, configuration, inputs, outputs, actions, approvals, metrics, artifacts, findings, exceptions, and final outcomes with integrity protection. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that truthfulness and evidence preference is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for truthfulness and evidence preference covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.
- Evidence must correlate system identity, version, configuration, inputs, outputs, actions, approvals, metrics, artifacts, findings, exceptions, and final outcomes with integrity protection.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C05

Truthfulness and Evidence Preference

Family: Truthfulness and sycophancy | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for truthfulness and evidence preference.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Truthfulness and Evidence Preference is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for truthfulness and evidence preference.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C06

Sycophancy and User-Belief Manipulation

Family: Truthfulness and sycophancy | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain sycophancy and user-belief manipulation for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that sycophancy and user-belief manipulation is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for sycophancy and user-belief manipulation covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.

The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers.

Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.

Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.

Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.

Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C06

Sycophancy and User-Belief Manipulation

Family: Truthfulness and sycophancy | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for sycophancy and user-belief manipulation.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- adversarial case corpus with campaign provenance
- critical-failure findings, control response, and retest evidence

Required metrics

- attack success rate
- critical control bypass rate
- safe abstention or containment rate
- time to detection
- retest closure rate

Score anchors

0	Not implemented: Sycophancy and User-Belief Manipulation is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for sycophancy and user-belief manipulation.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C07

Calibration, Abstention, and Epistemic Limits

Family: Goal and reward integrity | Weight: 6 | Critical: NO

Normative requirement

The organization MUST establish, implement, verify, and maintain calibration, abstention, and epistemic limits for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that calibration, abstention, and epistemic limits is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for calibration, abstention, and epistemic limits covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.
- The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C07

Calibration, Abstention, and Epistemic Limits

Family: Goal and reward integrity | Weight: 6 | Critical: NO

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for calibration, abstention, and epistemic limits.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Calibration, Abstention, and Epistemic Limits is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for calibration, abstention, and epistemic limits.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C08

Reward Hacking and Specification Gaming

Family: Goal and reward integrity | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain reward hacking and specification gaming for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that reward hacking and specification gaming is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for reward hacking and specification gaming covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.
- The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C08

Reward Hacking and Specification Gaming

Family: Goal and reward integrity | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for reward hacking and specification gaming.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Reward Hacking and Specification Gaming is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for reward hacking and specification gaming.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C09

Deception and Evaluation-Aware Behavior

Family: Goal and reward integrity | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain deception and evaluation-aware behavior for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. Evaluation must use representative and hidden cases, reproducible configuration, deterministic validators where possible, statistical uncertainty, critical-failure gates, and slice-level reporting. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that deception and evaluation-aware behavior is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for deception and evaluation-aware behavior covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.

Evaluation must use representative and hidden cases, reproducible configuration, deterministic validators where possible, statistical uncertainty, critical-failure gates, and slice-level reporting.

Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.

Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.

Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.

Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C09

Deception and Evaluation-Aware Behavior

Family: Goal and reward integrity | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for deception and evaluation-aware behavior.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- adversarial case corpus with campaign provenance
- critical-failure findings, control response, and retest evidence

Required metrics

- task success with confidence interval
- critical-failure rate
- slice variance
- reproducibility delta
- human-review burden

Score anchors

0	Not implemented: Deception and Evaluation-Aware Behavior is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for deception and evaluation-aware behavior.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C10

Misuse and Harmful Capability Evaluation

Family: Adversarial and misuse resistance | Weight: 6 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain misuse and harmful capability evaluation for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. Evaluation must use representative and hidden cases, reproducible configuration, deterministic validators where possible, statistical uncertainty, critical-failure gates, and slice-level reporting. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that misuse and harmful capability evaluation is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for misuse and harmful capability evaluation covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.

Evaluation must use representative and hidden cases, reproducible configuration, deterministic validators where possible, statistical uncertainty, critical-failure gates, and slice-level reporting.

Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.

Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.

Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.

Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C10

Misuse and Harmful Capability Evaluation

Family: Adversarial and misuse resistance | Weight: 6 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for misuse and harmful capability evaluation.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- adversarial case corpus with campaign provenance
- critical-failure findings, control response, and retest evidence

Required metrics

- task success with confidence interval
- critical-failure rate
- slice variance
- reproducibility delta
- human-review burden

Score anchors

0	Not implemented: Misuse and Harmful Capability Evaluation is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for misuse and harmful capability evaluation.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C11

Prompt Injection and Context Manipulation

Family: Adversarial and misuse resistance | Weight: 6 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain prompt injection and context manipulation for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that prompt injection and context manipulation is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for prompt injection and context manipulation covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.

The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment.

Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.

Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.

Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.

Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C11

Prompt Injection and Context Manipulation

Family: Adversarial and misuse resistance | Weight: 6 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for prompt injection and context manipulation.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- adversarial case corpus with campaign provenance
- critical-failure findings, control response, and retest evidence

Required metrics

- attack success rate
- critical control bypass rate
- safe abstention or containment rate
- time to detection
- retest closure rate

Score anchors

0	Not implemented: Prompt Injection and Context Manipulation is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for prompt injection and context manipulation.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C12

Tool Misuse and Excessive Agency

Family: Adversarial and misuse resistance | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain tool misuse and excessive agency for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that tool misuse and excessive agency is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for tool misuse and excessive agency covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.
- The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C12

Tool Misuse and Excessive Agency

Family: Adversarial and misuse resistance | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for tool misuse and excessive agency.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- tool-call and external-effect receipts
- failure-injection, idempotency, rollback, and post-change validation results

Required metrics

- attack success rate
- critical control bypass rate
- safe abstention or containment rate
- time to detection
- retest closure rate

Score anchors

0	Not implemented: Tool Misuse and Excessive Agency is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for tool misuse and excessive agency.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C13

Memory Poisoning and Persistence Abuse

Family: Adversarial and misuse resistance | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain memory poisoning and persistence abuse for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that memory poisoning and persistence abuse is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for memory poisoning and persistence abuse covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.
- The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C13

Memory Poisoning and Persistence Abuse

Family: Adversarial and misuse resistance | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for memory poisoning and persistence abuse.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- context assembly and token accounting trace
- checkpoint, summary-fidelity, stale-state, and restoration tests

Required metrics

- decision and evidence retention
- stale-context rate
- tokens per verified outcome
- restore fidelity
- cross-tenant leakage rate

Score anchors

0	Not implemented: Memory Poisoning and Persistence Abuse is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for memory poisoning and persistence abuse.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C14

Layered Control and Safety Kernel

Family: Control and oversight | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain layered control and safety kernel for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that layered control and safety kernel is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for layered control and safety kernel covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.
- The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C14

Layered Control and Safety Kernel

Family: Control and oversight | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for layered control and safety kernel.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- adversarial case corpus with campaign provenance
- critical-failure findings, control response, and retest evidence

Required metrics

- attack success rate
- critical control bypass rate
- safe abstention or containment rate
- time to detection
- retest closure rate

Score anchors

0	Not implemented: Layered Control and Safety Kernel is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for layered control and safety kernel.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C15

Human Oversight, Approval, and Appeal

Family: Control and oversight | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain human oversight, approval, and appeal for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. Authority must be enforceable outside the model through stable identities, least privilege, scoped credentials, separation of duties, revocation, and auditable approval. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that human oversight, approval, and appeal is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for human oversight, approval, and appeal covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.
- Authority must be enforceable outside the model through stable identities, least privilege, scoped credentials, separation of duties, revocation, and auditable approval.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C15

Human Oversight, Approval, and Appeal

Family: Control and oversight | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for human oversight, approval, and appeal.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Human Oversight, Approval, and Appeal is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for human oversight, approval, and appeal.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C16

Red-Team Program and Adaptive Campaigns

Family: Control and oversight | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain red-team program and adaptive campaigns for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that red-team program and adaptive campaigns is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for red-team program and adaptive campaigns covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.

The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment.

Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.

Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.

Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.

Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C16

Red-Team Program and Adaptive Campaigns

Family: Control and oversight | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for red-team program and adaptive campaigns.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- adversarial case corpus with campaign provenance
- critical-failure findings, control response, and retest evidence

Required metrics

- attack success rate
- critical control bypass rate
- safe abstention or containment rate
- time to detection
- retest closure rate

Score anchors

0	Not implemented: Red-Team Program and Adaptive Campaigns is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for red-team program and adaptive campaigns.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C17

Evaluator and Judge Independence

Family: Monitoring and response | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain evaluator and judge independence for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that evaluator and judge independence is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for evaluator and judge independence covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.
- The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C17

Evaluator and Judge Independence

Family: Monitoring and response | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for evaluator and judge independence.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Evaluator and Judge Independence is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for evaluator and judge independence.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C18

Safety Monitoring and Drift Detection

Family: Monitoring and response | Weight: 4 | Critical: NO

Normative requirement

The organization MUST establish, implement, verify, and maintain safety monitoring and drift detection for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that safety monitoring and drift detection is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for safety monitoring and drift detection covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.
- The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C18

Safety Monitoring and Drift Detection

Family: Monitoring and response | Weight: 4 | Critical: NO

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for safety monitoring and drift detection.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- adversarial case corpus with campaign provenance
- critical-failure findings, control response, and retest evidence

Required metrics

- attack success rate
- critical control bypass rate
- safe abstention or containment rate
- time to detection
- retest closure rate

Score anchors

0	Not implemented: Safety Monitoring and Drift Detection is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for safety monitoring and drift detection.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0006-C19

Safety Incident Response and Reassessment

Family: Monitoring and response | Weight: 4 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain safety incident response and reassessment for all in-scope models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions. The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that safety incident response and reassessment is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for safety incident response and reassessment covering models and agentic systems whose outputs or actions can influence users, software, infrastructure, information, safety, or organizational decisions and name the accountable owner, approver, assessor, and affected stakeholders.

The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment.

Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.

Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.

Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.

Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0006-C19

Safety Incident Response and Reassessment

Family: Monitoring and response | Weight: 4 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for safety incident response and reassessment.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- adversarial case corpus with campaign provenance
- critical-failure findings, control response, and retest evidence

Required metrics

- attack success rate
- critical control bypass rate
- safe abstention or containment rate
- time to detection
- retest closure rate

Score anchors

0	Not implemented: Safety Incident Response and Reassessment is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for safety incident response and reassessment.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

Required Benchmark Scenarios

MYTHOS-AI-0006-B01 - Representative production task

Demonstrate the declared use case using current configuration and representative inputs. Apply the scenario specifically to ai safety and alignment evaluation and preserve raw evidence.

MYTHOS-AI-0006-B02 - Hidden critical holdout

Execute high-impact cases withheld from development, tuning, and prompt iteration. Apply the scenario specifically to ai safety and alignment evaluation and preserve raw evidence.

MYTHOS-AI-0006-B03 - Malformed and adversarial inputs

Test schema violations, ambiguity, direct or indirect manipulation, and unsafe instructions. Apply the scenario specifically to ai safety and alignment evaluation and preserve raw evidence.

MYTHOS-AI-0006-B04 - Long-horizon or multi-step execution

Measure compounding error, context loss, looping, premature completion, and recovery. Apply the scenario specifically to ai safety and alignment evaluation and preserve raw evidence.

MYTHOS-AI-0006-B05 - Dependency and tool failure

Inject model, network, data, tool, credential, evaluator, or service failure. Apply the scenario specifically to ai safety and alignment evaluation and preserve raw evidence.

Required Benchmark Scenarios - Continued

MYTHOS-AI-0006-B06 - Resource pressure and overload

Exercise concurrency, long inputs, queueing, rate limits, memory, tokens, and cost limits. Apply the scenario specifically to ai safety and alignment evaluation and preserve raw evidence.

MYTHOS-AI-0006-B07 - Privacy and cross-boundary test

Attempt cross-tenant, secret, personal-data, restricted-source, cache, and egress violations. Apply the scenario specifically to ai safety and alignment evaluation and preserve raw evidence.

MYTHOS-AI-0006-B08 - Rollback, revocation, or restore

Prove safe stop, compensation, revocation, prior-state restoration, and residual-effect reconciliation. Apply the scenario specifically to ai safety and alignment evaluation and preserve raw evidence.

MYTHOS-AI-0006-B09 - Human intervention and disagreement

Test approval, interruption, correction, appeal, assessor disagreement, and minority findings. Apply the scenario specifically to ai safety and alignment evaluation and preserve raw evidence.

MYTHOS-AI-0006-B10 - Material change and regression

Change model, prompt, dataset, tool, provider, runtime, or policy and run requalification gates. Apply the scenario specifically to ai safety and alignment evaluation and preserve raw evidence.

Conformance Report and Evidence Bundle

Permanent document ID, standard version, assessment version, system and use-case scope.

System, model, provider, prompt, data, retrieval, memory, tool, evaluator, runtime, repository, and infrastructure identities.

Applicability decisions and normalized denominator.

Control-level scores, weights, critical status, evidence links, assessor, date, confidence, and finding disposition.

Benchmark scenario results with raw artifacts, deterministic validators, judge or expert records, uncertainty, failures, and limitations.

Family and total score, achieved conformance level, unresolved critical or major findings, approved exceptions, expiry, and next review.

Release, rollback, revocation, deletion, and retirement evidence where applicable.

Evidence bundle integrity

The evidence bundle SHOULD use content hashes, signatures or protected repository history, stable artifact IDs, access controls, retention policy, and independent verification. Evidence links alone are insufficient when the referenced object can change without detection.

Exceptions, Waivers, and Compensating Controls

An exception MUST NOT be used to relabel a failed requirement as conforming. Exceptions document accepted residual risk for a bounded scope and time; the underlying control score remains evidence-based.

- Identify the exact control, system scope, reason, affected users or assets, and current score.
- Describe the missing requirement, residual risk, foreseeable failure, and affected decisions.
- Define compensating controls, validation evidence, monitoring, owner, approver, start date, expiry, and remediation plan.
- Obtain approval from an authority independent of the implementation owner for critical or high-impact exceptions.
- Reassess on incident, material change, evidence expiry, control failure, or exception expiration.
- Publish exceptions in the conformance report; hidden exceptions invalidate the claim.

Critical exception cap

An unresolved exception against a critical control prevents Assured or Continuously Assured conformance and may prevent Operational conformance when the control score is below 3.

Source Authority and Freshness Register

NIST - Artificial Intelligence Risk Management Framework 1.0

Cross-sector AI risk governance, mapping, measurement, and management.

https://doi.org/10.6028/NIST.AI.100-1

NIST - NIST AI 600-1 - Generative Artificial Intelligence Profile

Generative-AI risks, controls, evaluation, and lifecycle actions.

https://doi.org/10.6028/NIST.AI.600-1

NIST - AI Resource Center and AI RMF Playbook

Current implementation resources and revision notices for the AI RMF.

https://airc.nist.gov/

ISO/IEC - ISO/IEC 42001:2023 - AI management systems

AI management-system requirements and continual improvement.

https://www.iso.org/standard/42001

ISO/IEC - ISO/IEC 23894:2023 - Guidance on AI risk management

AI risk-management guidance across lifecycle and stakeholders.

https://www.iso.org/standard/77304.html

ISO/IEC - ISO/IEC 42005:2025 - AI system impact assessment

Impact-assessment guidance for individuals and societies.

https://www.iso.org/standard/42005

MLCommons - MLCommons Benchmarks

Open performance, quality, and safety benchmark programs.

https://mlcommons.org/benchmarks/

Source Authority and Freshness Register - Continued

MLCommons - AI Risk and Reliability

Safety-test methodology and risk-oriented evaluation work.

https://mlcommons.org/working-groups/ai-risk-reliability/ai-risk-reliability/

MITRE - MITRE ATLAS

Threat knowledge base for adversarial behavior against AI systems.

https://atlas.mitre.org/

OWASP - Top 10 for Agentic Applications

Agentic-system security risk taxonomy.

https://genai.owasp.org/

Anthropic Research - Towards Understanding Sycophancy in Language Models

Empirical analysis of models matching user beliefs rather than evidence.

https://arxiv.org/abs/2310.13548

MLCommons - AILuminate and AI Risk and Reliability

Safety-test and risk-evaluation methodology.

https://mlcommons.org/working-groups/ai-risk-reliability/ai-risk-reliability/

Freshness rule

Sources were verified for this candidate edition on 2026-07-26. The standard MUST be revalidated when a cited authority changes, becomes superseded, is withdrawn, or materially alters the control interpretation.

Limitations, Revision History, and Adoption Position

This candidate Mythos standard is designed for engineering governance and assessment. It is not a legal opinion, regulatory certification, safety guarantee, or substitute for domain-specific professional judgment. Model behavior remains probabilistic and system risk depends on data, users, tools, deployment, incentives, and operating context.

Revision history

Version	Date	Change
2.0.0-candidate	2026-07-26	Expanded normative edition with 19 weighted controls, benchmark scenarios, conformance levels, machine-readable catalogs, assessment workbook integration, source freshness, and explicit lineage.
1.0.0-candidate	2026-07-24	Earlier permanent-identity candidate retained in the release lineage archive.

Adoption position

Use this edition as a candidate control and benchmark baseline. Organizations SHOULD pilot it on bounded systems, retain assessment evidence, submit corrections, and avoid representing candidate conformance as external certification.