MYTHOS SYSTEMS

PERMANENT LIBRARY PUBLICATION

ENGINEERING STANDARDS LIBRARY / CYBERSECURITY AND NETWORK SECURITY

API Security and Token Lifecycle Standard

Defines API authorization, tokens, scopes, rotation, validation, monitoring, and revocation.

PERMANENT DOCUMENT ID
MYTHOS-SEC-0011

PUBLICATION
Candidate Standard

ASSURANCE SURFACE
6 Atomic Criteria / 6 Families

PERMANENT PUBLICATION IDENTITY

API Security and Token Lifecycle Standard

Defines API authorization, tokens, scopes, rotation, validation, monitoring, and revocation.

DOCUMENT ID
MYTHOS-SEC-0011

VERSION
1.0.0-candidate

STATUS
Candidate Standard

PUBLISHER
Mythos Systems

PUBLICATION DATE
2026-07-23

SCHEDULED REVIEW
2027-01-23

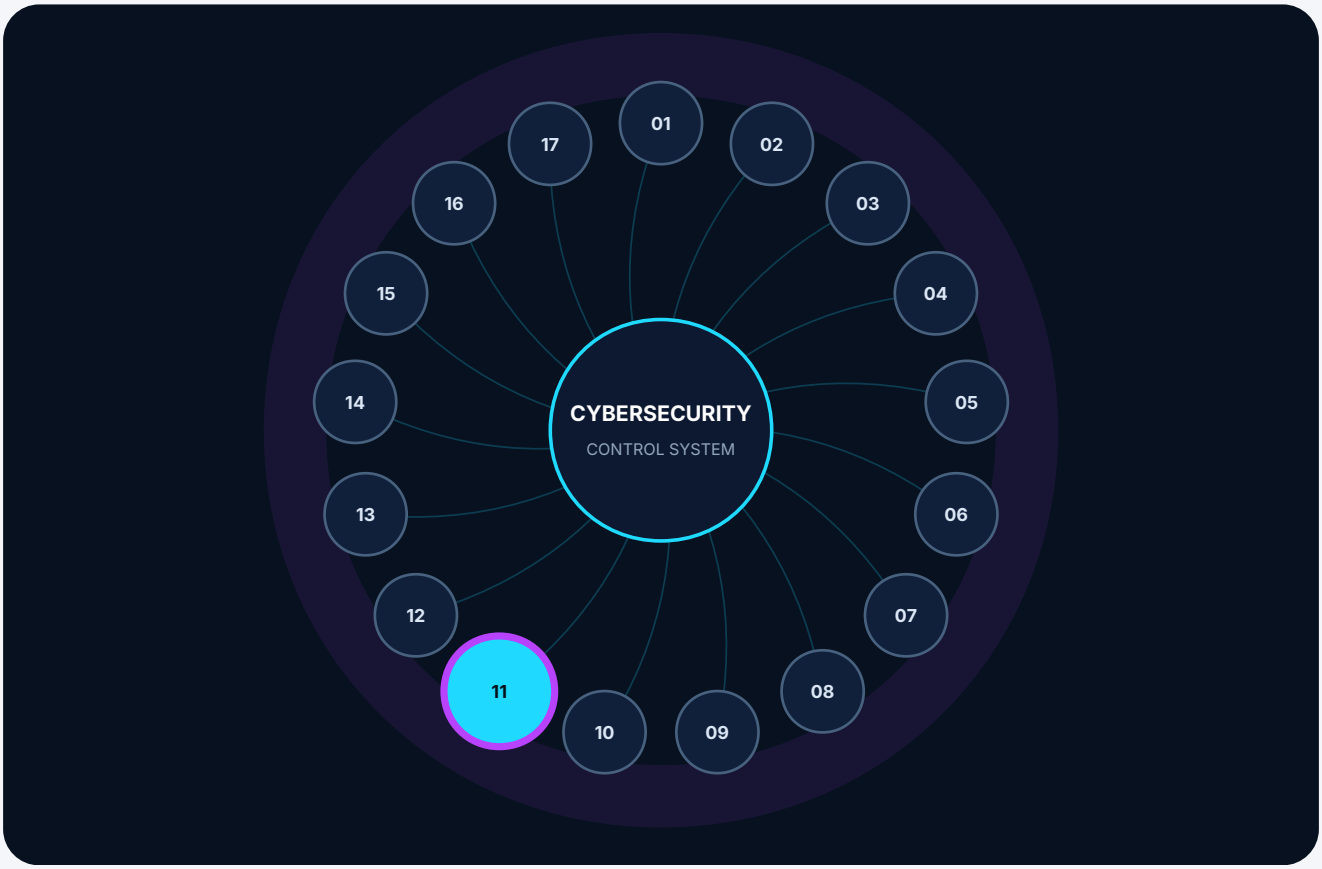
CLASSIFICATION
Public

6	6	4	1
ATOMIC CRITERIA	CAPABILITY FAMILIES	ASSESSMENT PROFILES	PERMANENT IDENTITY

Publication doctrine. This standard establishes durable requirements for its defined scope. Interpretation must preserve the permanent document identity, recorded evidence, and formal revision history.

DOMAIN CONTROL SYSTEM

MYTHOS-SEC-0011 inside the cybersecurity and network security standard constellation



Connected assurance

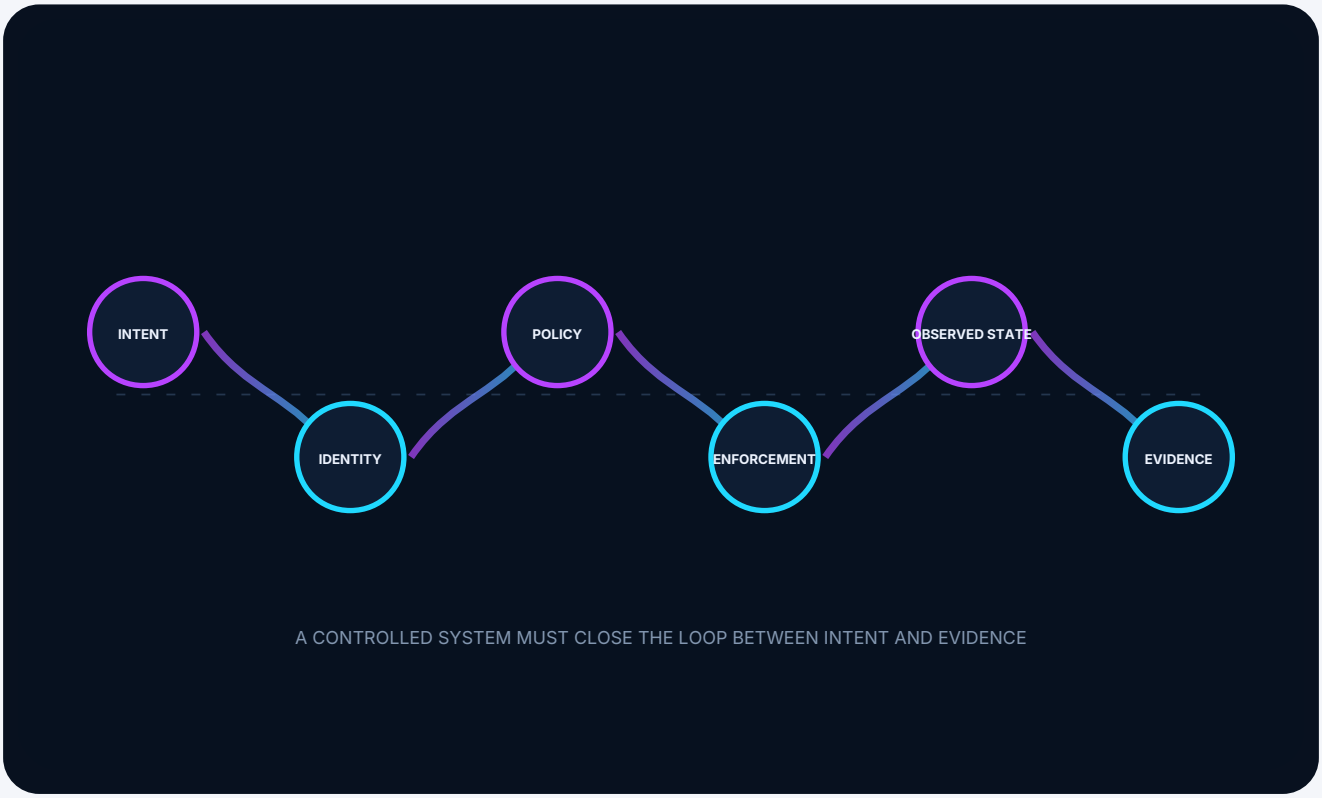
Architecture, policy, implementation, operations, and evidence are evaluated as one controlled system.

Specialization rule

Companion standards may add precision, but they cannot silently weaken this publication's requirements.

CLOSED-LOOP ARCHITECTURE

Intent becomes trustworthy only when observed state returns as evidence



State separation

Desired, approved, deployed, observed, historical, and simulated states must remain distinguishable.

Recoverable change

Material change requires validation, authorization, monitoring, containment, and a credible recovery path.

EVALUATION TOPOLOGY

The capability families and atomic criteria of MYTHOS-SEC-0011

Capability families

Families expose the major engineering surfaces and preserve the source weighting model.

Atomic criteria

Each criterion carries a question, maturity spectrum, evidence expectations, assessment method, and operational signals.

NAVIGATION

Contents

ENGINEERING DOCTRINE

0. Executive Mandate

The architecture of API security has failed.

Organizations expose microservices and AI agent tool endpoints behind API gateways, believing that TLS and a Bearer token are sufficient defenses. They rely on network perimeters and Web Application Firewalls (WAFs) to stop injection attacks, while their endpoints blindly trust client-side payloads. They issue long-lived JSON Web Tokens (JWTs) that cannot be instantly revoked, and they fail to enforce object-level authorization, allowing any authenticated user to read any other user's data.

This standard exists because:

1. **Authentication is Not Authorization:** Just because a user is logged in does not mean they are authorized to access a specific resource. Broken Object Level Authorization (BOLA/IDOR) is the #1 API threat. Systems MUST enforce strict, per-object, per-request authorization checks.
2. **Bearer Tokens are Stolen Credentials:** A standard OAuth 2.0 Bearer token can be intercepted and replayed by anyone who possesses it. Systems MUST adopt OAuth 2.1 DPoP (Demonstrating Proof-of-Possession) or mTLS-bound tokens, cryptographically binding the token to the client's private key.
3. **Unvalidated Payloads are Injection Vectors:** Accepting arbitrary JSON payloads and passing them to downstream services or LLM tool-calling APIs guarantees mass assignment, SQL injection, or prompt injection. Systems MUST enforce strict, schema-first payload validation at the gateway layer.
4. **Token Revocation Cannot Wait for Expiry:** Relying on short-lived JWTs is insufficient; a compromised token can do catastrophic damage in 15 minutes. Systems MUST implement real-time token introspection or revocation lists (e.g., token blacklists via Redis) for immediate termination.
5. **APIs Must Defend Against Resource Exhaustion:** A single complex GraphQL query or an unbounded REST loop can consume all database resources. Systems MUST implement adaptive rate limiting, query complexity analysis, and strict timeout budgets.

This document establishes the **Mythos API Security Standard (MAPIS)**, a comprehensive, evidence-based methodology for evaluating API architectures against payload integrity, token non-repudiation, and zero-trust authorization.

ENGINEERING DOCTRINE

Part I. Scope and Applicability

1.1 In Scope

This standard covers the evaluation of:

- API Gateways and edge proxies (Kong, Envoy, APISIX, Tyk)
- REST, GraphQL, and gRPC security architectures
- OWASP API Security Top 10 (BOLA, Broken Authentication, Injection, etc.)
- OAuth 2.1, OpenID Connect (OIDC), and JWT lifecycle management
- Token binding (DPoP, mTLS) and real-time revocation
- Schema validation, payload sanitization, and mass assignment prevention
- Rate limiting, throttling, and query complexity analysis

1.2 Out of Scope

- General identity and PKI infrastructure (covered in MYTHOS-ID-0001)
- Internal service-to-service zero-trust (covered in MYTHOS-SEC-0003)
- General API contract design (covered in MYTHOS-DST-0002)

1.3 Audience

- Security architects and application security engineers
- Platform engineers managing API gateways and service meshes
- Backend engineers designing microservice and AI tool APIs
- DevSecOps teams implementing CI/CD API testing
- Standards bodies seeking a reference API security methodology

---

ENGINEERING DOCTRINE

Part II. Definitions and Taxonomy

2.1 API Security Dimensions

Dimension	Definition
BOLA (Broken Object Level Authorization)	A vulnerability where an API endpoint fails to verify if the authenticated user has the right to access a specific object ID (e.g., `/users/123`), allowing IDOR (Insecure Direct Object Reference).
DPoP (Demonstrating Proof-of-Possession)	An OAuth 2.1 extension that cryptographically binds a token to the client's public key, preventing token replay even if intercepted.
Schema-First Validation	An architectural requirement where the API gateway rejects any request payload that does not strictly match a predefined, compiled OpenAPI/Protobuf schema.
Mass Assignment	A vulnerability where an API blindly maps client-provided JSON keys to internal object properties, allowing a user to overwrite fields they are not authorized to change (e.g., `{"role": "admin"}`).
Query Complexity Analysis	A GraphQL security mechanism that assigns a computational cost to schema fields and rejects queries that exceed a maximum cost threshold, preventing resource exhaustion.

2.2 The API Security Doctrine

> A perimeter WAF cannot save a vulnerable endpoint. A Bearer token is a stolen key waiting to be used. An API that trusts client JSON is an injection vector. If the authorization does not check the object ID, it is just authentication.

ENGINEERING DOCTRINE

Part III. Evaluation Methodology

3.1 Scoring Model

Each capability is scored from 0 to 5.

Score	Meaning
0	Fails basic task; BOLA vulnerable, Bearer tokens only, no schema validation
1	Basic authentication, but flat authorization and no token revocation
2	Functional RBAC, but lacks DPoP or strict payload validation
3	Solid production performance; strict schemas, DPoP, real-time revocation, BOLA prevention
4	Strong performance; adaptive rate limiting, query complexity bounds, mTLS-bound tokens
5	Best-in-class; sets the standard for mathematically verified, zero-trust API architecture

Weighting

Category	Weight	Rationale
Object-Level Authorization (BOLA/IDOR)	20%	The #1 critical API threat vector
Token Binding & Non-Repudiation	20%	Bearer tokens are trivially replayable
Schema Validation & Payload Integrity	15%	Unvalidated JSON causes injection and mass assignment
Token Revocation & Lifecycle	15%	Long-lived tokens cannot be secured
DoS Protection & Rate Limiting	15%	Unbounded queries exhaust compute resources
Transport & Edge Security	15%	Internal API traffic must be encrypted and inspected

Total: 100%

A system MUST score at least 3.0 in Object-Level Authorization and Token Binding to be considered for production use.

EVALUATION AND ASSURANCE SYSTEM

Capability claims are bounded by evidence, context, and reproducible assessment.

6 atomic criteria across 6 capability families define the evaluation surface of this standard.

4.1 / CAPABILITY FAMILY

Object-Level Authorization (BOLA/ IDOR)

SOURCE WEIGHT 20%

4.1.1
Resource Access Control

MYTHOS-SEC-0011-EVAL-4.1.1

Resource Access Control

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Object-Level Authorization (BOLA/ IDOR)	4.1.1	20%	Evidence-led

Does the API enforce per-object authorization for every request?

0
Relies solely on authentication; if you have a token, you can access any object ID

1
Basic role checks (e.g., "is user an admin"), but no object-level ownership checks

2
Object-level checks exist, but implemented inconsistently across endpoints

3
Strict, centralized authorization policy (e.g., OPA/Rego) evaluating user ID vs. object owner for every request

4
Attribute-Based Access Control (ABAC) dynamically evaluating context (time, IP, resource state)

5
Leading authorization: formally verified access control logic, zero BOLA/IDOR vulnerabilities possible, mathematical proof of tenant isolation

ENGINEERING INTERPRETATION

This criterion evaluates whether resource access control is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- source-of-truth objects, Git history, observed-state snapshots, and reconciliation evidence
- approved architecture or policy record
- current configuration or platform export

ASSESSMENT PROCEDURE

1. introduce controlled drift and verify detection, adjudication, and restoration
2. Examine authoritative design, policy, configuration, and lifecycle records.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- multiple authorities, silent manual changes, stale inventory, and destructive auto-remediation
- The control exists only in diagrams or policy text.

OPERATIONAL MEASURES

- drift detection time
- reconciliation success
- unowned object count
- coverage of in-scope assets or flows

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

4.2 / CAPABILITY FAMILY

Token Binding and Non-Repudiation

SOURCE WEIGHT 20%

4.2.1
Replay Prevention

MYTHOS-SEC-0011-EVAL-4.2.1

Replay Prevention

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Token Binding and Non-Repudiation	4.2.1	20%	Evidence-led

Are access tokens cryptographically bound to the client?

0
Opaque Bearer tokens sent in URL parameters or headers (highly replayable)

1
JWT Bearer tokens in `Authorization` header

2
Short-lived JWT Bearer tokens (< 5 minutes)

3
OAuth 2.1 DPoP: tokens bound to the client's asymmetric public key; proof required per request

4
mTLS-bound tokens: tokens bound to the client's TLS certificate (highest transport security)

5
Leading binding: formally verified cryptographic proof-of-possession, zero replay capability, transaction-bound non-repudiation

ENGINEERING INTERPRETATION

This criterion evaluates whether replay prevention is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- cryptographic inventory, algorithm registry, key records, certificate data, and compatibility tests
- approved architecture or policy record
- current configuration or platform export

ASSESSMENT PROCEDURE

1. test rotation, hybrid negotiation, rollback, revocation, and algorithm-policy enforcement
2. Examine authoritative design, policy, configuration, and lifecycle records.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- hardcoded algorithms, unknown dependencies, missing rollback, and unverifiable keys
- The control exists only in diagrams or policy text.

OPERATIONAL MEASURES

- inventory coverage
- deprecated algorithm count
- rotation success rate
- coverage of in-scope assets or flows

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

4.3 / CAPABILITY FAMILY

Schema Validation and Payload Integrity

SOURCE WEIGHT 15%

4.3.1
Input Enforcement

MYTHOS-SEC-0011-EVAL-4.3.1

Input Enforcement

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Schema Validation and Payload Integrity	**4.3.1**	**15%**	**Evidence-led**

Does the system reject malformed or unauthorized payloads at the edge?

0
API accepts arbitrary JSON and attempts to parse it downstream

1
Basic type checking (e.g., "is this an integer"), but allows extra fields

2
Schema validation exists, but not enforced at the gateway

3
Strict, gateway-level schema validation (OpenAPI/Protobuf); extra fields are rejected, not ignored

4
Automated mass assignment prevention: schema explicitly defines writable vs. readable fields

5
Leading integrity: formally verified payload bounds, zero unvalidated fields reach backend logic, cryptographic validation of request signatures

ENGINEERING INTERPRETATION

This criterion evaluates whether input enforcement is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- approved architecture or policy record
- current configuration or platform export
- change and exception records
- monitoring, test, or assessment evidence

ASSESSMENT PROCEDURE

1. Examine authoritative design, policy, configuration, and lifecycle records.
2. Interview the accountable owner and operators responsible for the control.
3. Test a representative positive, negative, failure, and recovery scenario.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- The control exists only in diagrams or policy text.
- Observed state differs from approved intent without a governed exception.
- Evidence cannot establish scope, time, owner, or operating effectiveness.

OPERATIONAL MEASURES

- coverage of in-scope assets or flows
- age and closure rate of unresolved findings
- percentage of changes passing pre-deployment validation
- time to detect and contain control failure

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

4.4 / CAPABILITY FAMILY

Token Revocation and Lifecycle

SOURCE WEIGHT 15%

4.4.1
Session Termination

MYTHOS-SEC-0011-EVAL-4.4.1

Session Termination

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Token Revocation and Lifecycle	**4.4.1**	**15%**	**Evidence-led**

Can an active token be instantly invalidated?

0
Long-lived tokens (days/weeks); cannot be revoked, must wait for expiry

1
Refresh token rotation, but access tokens remain valid until expiry

2
Centralized token introspection (OAuth 2.0 RFC 7662) checked on every request

3
Real-time revocation lists (Redis-backed blacklist) with < 1ms latency penalty

4
Push-based revocation: gateway receives webhooks from IdP to instantly drop active tokens

5
Leading lifecycle: formally verified revocation propagation, zero delay between revocation and enforcement, absolute denial of revoked sessions

ENGINEERING INTERPRETATION

This criterion evaluates whether session termination is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- approved architecture or policy record
- current configuration or platform export
- change and exception records
- monitoring, test, or assessment evidence

ASSESSMENT PROCEDURE

1. Examine authoritative design, policy, configuration, and lifecycle records.
2. Interview the accountable owner and operators responsible for the control.
3. Test a representative positive, negative, failure, and recovery scenario.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- The control exists only in diagrams or policy text.
- Observed state differs from approved intent without a governed exception.
- Evidence cannot establish scope, time, owner, or operating effectiveness.

OPERATIONAL MEASURES

- coverage of in-scope assets or flows
- age and closure rate of unresolved findings
- percentage of changes passing pre-deployment validation
- time to detect and contain control failure

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

4.5 / CAPABILITY FAMILY

DoS Protection and Rate Limiting

SOURCE WEIGHT 15%

4.5.1
Resource Exhaustion Prevention

MYTHOS-SEC-0011-EVAL-4.5.1

Resource Exhaustion Prevention

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
DoS Protection and Rate Limiting	4.5.1	15%	Evidence-led

How does the API defend against volumetric and complex-query attacks?

0
No rate limiting; single requests can exhaust the database

1
Global IP-based rate limiting (easily bypassed by botnets)

2
Per-user/per-token rate limiting on REST endpoints

3
GraphQL query complexity analysis and depth limiting

4
Adaptive rate limiting: limits adjust dynamically based on backend latency and error rates

5
Leading protection: formally verified compute bounds, zero ability to exhaust resources, AI-driven anomaly detection for traffic spikes

ENGINEERING INTERPRETATION

This criterion evaluates whether resource exhaustion prevention is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- source-of-truth objects, Git history, observed-state snapshots, and reconciliation evidence
- approved architecture or policy record
- current configuration or platform export

ASSESSMENT PROCEDURE

1. introduce controlled drift and verify detection, adjudication, and restoration
2. Examine authoritative design, policy, configuration, and lifecycle records.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- multiple authorities, silent manual changes, stale inventory, and destructive auto-remediation
- The control exists only in diagrams or policy text.

OPERATIONAL MEASURES

- drift detection time
- reconciliation success
- unowned object count
- coverage of in-scope assets or flows

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

4.6 / CAPABILITY FAMILY

Transport and Edge Security

SOURCE WEIGHT 15%

4.6.1
Traffic Isolation

MYTHOS-SEC-0011-EVAL-4.6.1

Traffic Isolation

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Transport and Edge Security	4.6.1	15%	Evidence-led

How is API traffic secured at the network and proxy layer?

0
HTTP plaintext internally, TLS terminated only at the edge

1
TLS 1.3 to the edge, mTLS internally but self-signed certificates

2
Automated certificate rotation (Let's Encrypt / internal CA) for all services

3
Zero-trust API gateway: all traffic inspected, mTLS enforced, egress controlled

4
Service mesh (Istio/Linkerd) enforcing mTLS and authorization policies between every microservice

5
Leading isolation: formally verified mTLS everywhere, zero plaintext API traffic, cryptographic attestation of service identity

ENGINEERING INTERPRETATION

This criterion evaluates whether traffic isolation is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- API inventory, authorization policy, token metadata, gateway logs, and abuse controls
- approved architecture or policy record
- current configuration or platform export

ASSESSMENT PROCEDURE

1. test object and function authorization, token misuse, replay, expiration, and rate limits
2. Examine authoritative design, policy, configuration, and lifecycle records.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- bearer tokens without audience limits, hidden APIs, and inconsistent authorization
- The control exists only in diagrams or policy text.

OPERATIONAL MEASURES

- authorized API coverage
- token age
- replay rejection
- abuse-control efficacy
- coverage of in-scope assets or flows

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

LIFECYCLE AND ASSURANCE

Part V. The Mythos API Security Suite (MAPIS)

Traditional security benchmarks measure WAF rule coverage. The MAPIS evaluates BOLA resistance, token replay resistance, and payload integrity.

5.1 Suite Composition

5.1.1 MAPIS-BOLA

- **IDOR Injection:** 100+ tests iterating object IDs (e.g., `/users/1` to `/users/1000`) using User A's token, verifying the system denies access to User B's resources.
- **Mass Assignment:** 50+ tests appending `{"role":"admin"}` or `{"is_verified":true}` to standard PUT/PATCH requests, verifying the schema rejects the fields.

5.1.2 MAPIS-Token

- **Token Replay:** 100+ tests intercepting a DPoP token and attempting to replay it from a different client (without the private key), verifying the gateway rejects it.
- **Revocation Latency:** 50+ tests revoking a user's session and immediately sending requests, verifying the system blocks them within < 100ms.

5.2 Execution Protocol

PUBLICATION BOUNDARY

Candidate standard. Permanent identity. Evidence before authority.

Approval requires designated technical, security, legal, accessibility, and governance review with recorded findings and dispositions.