MYTHOS SYSTEMS

PERMANENT LIBRARY PUBLICATION

ENGINEERING STANDARDS LIBRARY / CYBERSECURITY AND NETWORK SECURITY

Virtualization, Hypervisor, Container, and Workload Security Standard

Defines isolation, images, runtime, orchestration, patching, policy, and monitoring.

PERMANENT DOCUMENT ID
MYTHOS-SEC-0012

PUBLICATION
Candidate Standard

ASSURANCE SURFACE
6 Atomic Criteria / 6 Families

PERMANENT PUBLICATION IDENTITY

Virtualization, Hypervisor, Container, and Workload Security Standard

Defines isolation, images, runtime, orchestration, patching, policy, and monitoring.

DOCUMENT ID
MYTHOS-SEC-0012

VERSION
1.0.0-candidate

STATUS
Candidate Standard

PUBLISHER
Mythos Systems

PUBLICATION DATE
2026-07-23

SCHEDULED REVIEW
2027-01-23

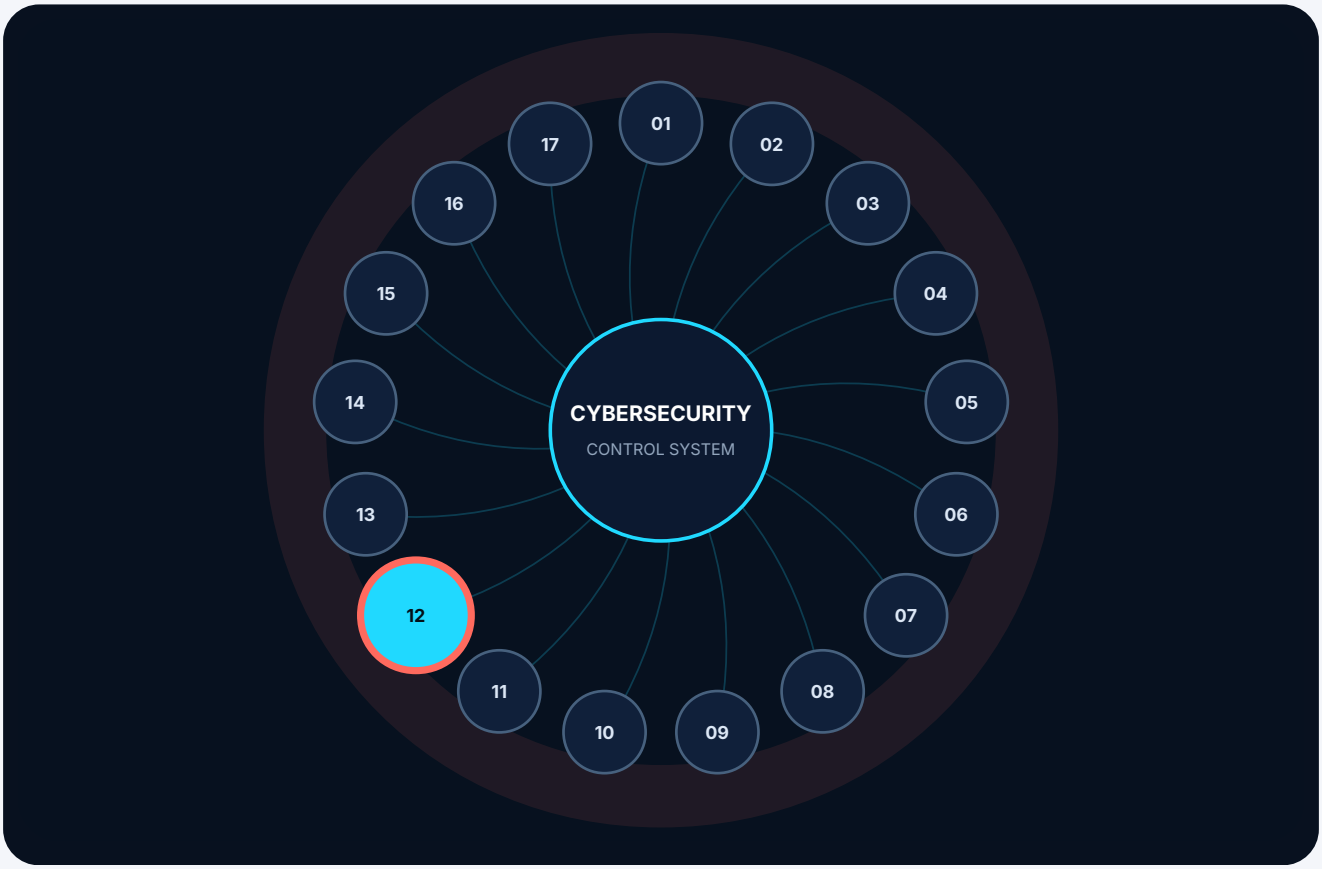
CLASSIFICATION
Public

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-0012 inside the cybersecurity and network security standard constellation



Connected assurance

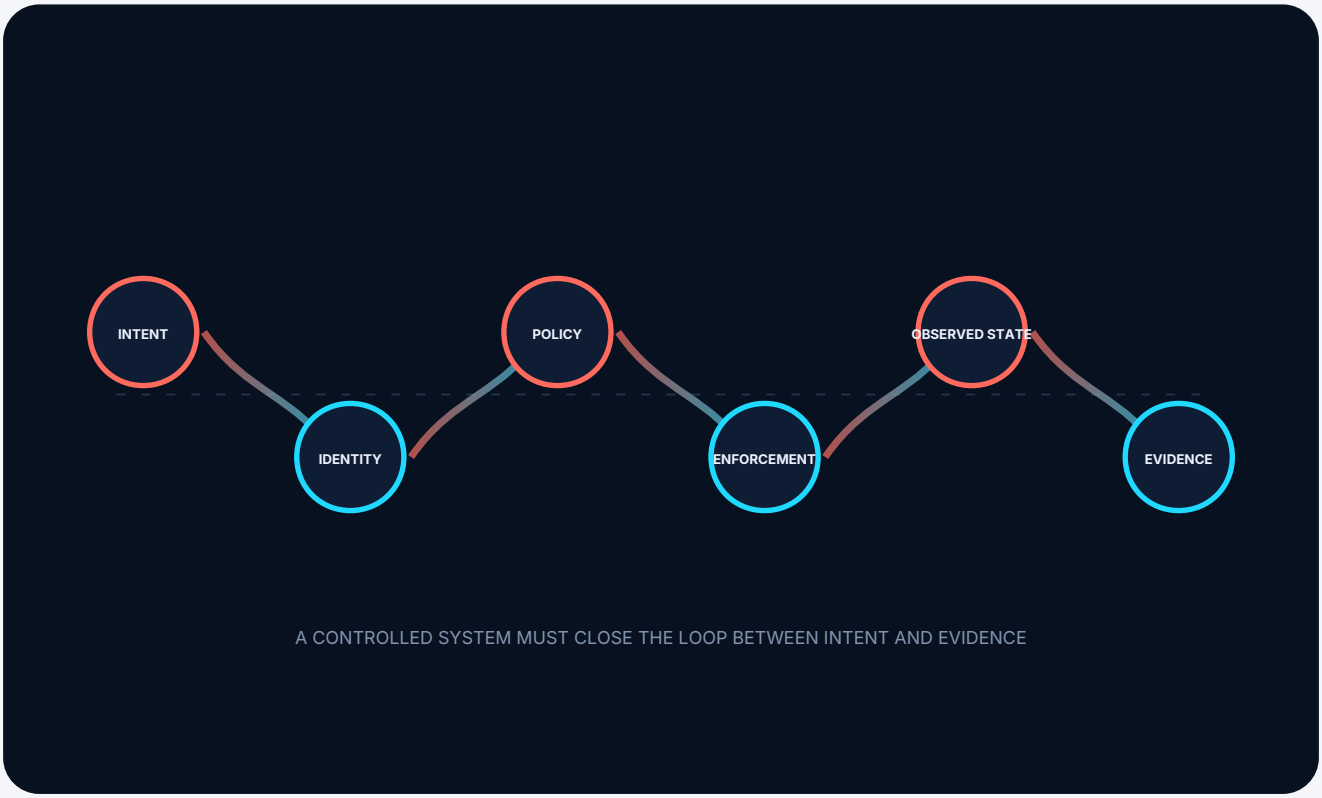
Architecture, policy, implementation, operations, and evidence are evaluated as one controlled system.

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



A CONTROLLED SYSTEM MUST CLOSE THE LOOP BETWEEN INTENT AND 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-0012

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 virtualization and container security has failed.

Organizations treat containers as lightweight, isolated Virtual Machines. They run the Docker daemon as root, execute application processes as root inside the container, and rely on default Linux namespaces for isolation. When an application vulnerability is exploited, the attacker instantly escapes the container, compromises the shared host kernel, and pivots to exfiltrating data from every other tenant on the node.

This standard exists because:

1. **Containers are Not VMs:** A container is simply a restricted process sharing the host's kernel. A kernel exploit (0-day) or a misconfigured capability grants instant host takeover. Systems MUST utilize microVMs (Firecracker), sandboxed runtimes (gVisor), or hardware-isolated VMs for multi-tenant or untrusted workloads.
2. **Root is an Attack Vector:** Running a container or pod as the `root` user is an unconditional security vulnerability. If the application is compromised, the attacker already has root privileges within the container, making kernel exploitation trivial. Systems MUST enforce rootless containers and drop all Linux capabilities by default.
3. **Bloated Images are Supply Chain Liabilities:** Using generic base images (e.g., `ubuntu:latest`) containing full OS utilities, package managers, and shells provides an attacker with a ready-made toolkit once they exploit the app. Systems MUST use minimal, immutable, distroless images that contain only the application binary and its immediate dependencies.
4. **The Hypervisor is the Last Line of Defense:** As AI agents execute untrusted, third-party code (e.g., MCP tools), the hypervisor must provide absolute hardware isolation. Systems MUST evaluate and implement Confidential VMs (CVMs) with encrypted memory (AMD SEV-SNP, Intel TDX) to protect data in-use from host administrators and compromised hypervisors.

This document establishes the **Mythos Virtualization & Container Standard (MVCS)**, a comprehensive, evidence-based methodology for evaluating virtualization and container architectures against kernel isolation, privilege escalation, and supply chain integrity.

ENGINEERING DOCTRINE

Part I. Scope and Applicability

1.1 In Scope

This standard covers the evaluation of:

- Container runtimes (containerd, CRI-O, Docker) and isolation boundaries
- MicroVMs and sandboxed runtimes (Firecracker, gVisor, Kata Containers)
- Hypervisor security (KVM, ESXi, AWS Nitro, Azure Hyper-V)
- Confidential VMs (CVMs) and encrypted memory (SEV-SNP, TDX)
- Kubernetes Pod Security Admission (PSA) and RBAC
- eBPF-based runtime threat detection (Falco, Tetragon)
- Image provenance, distroless images, and SBOM verification

1.2 Out of Scope

- General Kubernetes orchestration and cluster design (covered in MYTHOS-CLD-0007)
- WebAssembly (WASM) sandboxing (covered in MYTHOS-EMT-0001)
- General identity and access management (covered in MYTHOS-ID-0001)

1.3 Audience

- Platform engineers and cloud architects designing compute infrastructure
- DevSecOps teams building CI/CD pipelines for containerized applications
- Security engineers evaluating hypervisor and runtime isolation
- AI engineers executing untrusted agentic code in sandboxed environments
- Standards bodies seeking a reference virtualization security methodology

---

ENGINEERING DOCTRINE

Part II. Definitions and Taxonomy

2.1 Isolation Dimensions

Dimension	Definition
Container Escape	The act of exploiting a vulnerability in the shared host kernel or a misconfigured container runtime to break out of the container's namespace and execute code directly on the host.
MicroVM	A lightweight virtual machine with a minimal kernel and boot time (e.g., AWS Firecracker), providing hardware-level hypervisor isolation with container-like speed.
Rootless Container	A container that runs its processes as a non-root user and manages the container runtime itself via user namespaces, ensuring root inside the container maps to an unprivileged user on the host.
Confidential VM (CVM)	A virtual machine where the memory and CPU registers are encrypted by the hardware (e.g., AMD SEV-SNP, Intel TDX), preventing the hypervisor, host OS, or cloud administrator from reading the data in-use.
Distroless Image	A container image that contains only the application binary and its immediate runtime dependencies, stripping out the operating system, shell, and package managers.

2.2 The Virtualization Doctrine

> A container is a process, not a VM. Root is an attack vector. A shell in a container is a hacker's toolkit. If the hypervisor can read the memory, the tenant is compromised.

ENGINEERING DOCTRINE

Part III. Evaluation Methodology

3.1 Scoring Model

Each capability is scored from 0 to 5.

Score	Meaning
0	Fails basic task; root containers, bloated images, shared kernel, no runtime security
1	Basic RBAC, but runs as root and relies on default Docker isolation
2	Functional non-root containers, but lacks microVM isolation or distroless images
3	Solid production performance; rootless, distroless, eBPF monitoring, microVMs for untrusted code
4	Strong performance; Confidential VMs (CVMs), formally verified RBAC, zero standing privileges
5	Best-in-class; sets the standard for mathematically verified, hardware-isolated compute

Weighting

Category	Weight	Rationale
Runtime Isolation & MicroVMs	20%	Shared kernels guarantee container escapes eventually
Privilege & Rootless Enforcement	20%	Root containers provide instant privilege escalation
Image Provenance & Distroless	15%	Bloated images provide attackers with tools and vulnerabilities
Hypervisor & Confidential Computing	15%	The hypervisor is the ultimate isolation boundary
Runtime Detection (eBPF)	15%	Static defenses cannot stop zero-day kernel exploits
Network Microsegmentation	15%	Flat container networks allow lateral movement

Total: 100%

A system MUST score at least 3.0 in Runtime Isolation and Privilege Enforcement 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

Runtime Isolation and MicroVMs

SOURCE WEIGHT 20%

4.1.1
Boundary Integrity

MYTHOS-SEC-0012-EVAL-4.1.1

Boundary Integrity

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Runtime Isolation and MicroVMs	**4.1.1**	**20%**	**Evidence-led**

How is the workload isolated from the shared host kernel?

0
Standard runc/containerd sharing the host kernel (high risk of escape)

1
AppArmor/SELinux profiles applied, but still shares kernel

2
gVisor (userspace kernel intercept) used for untrusted workloads

3
MicroVMs (Firecracker/Kata) used for multi-tenant or untrusted workloads, providing hardware isolation

4
Hybrid architecture: standard containers for trusted internal services, microVMs for external/untrusted AI tools

5
Leading isolation: formally verified boundaries, zero shared kernel surface area, hardware-backed memory encryption for all untrusted code

ENGINEERING INTERPRETATION

This criterion evaluates whether boundary integrity 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.

4.2 / CAPABILITY FAMILY

Privilege and Rootless Enforcement

SOURCE WEIGHT 20%

4.2.1
Escalation Prevention

MYTHOS-SEC-0012-EVAL-4.2.1

Escalation Prevention

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Privilege and Rootless Enforcement	**4.2.1**	**20%**	**Evidence-led**

Are containers and the container runtime prevented from operating as root?

0
Docker daemon runs as root; containers run as root (critical failure)

1
Containers run as non-root, but daemon runs as root

2
Container runtime runs rootless via user namespaces

3
Kubernetes Pod Security Admission (PSA) set to `restricted`; all capabilities dropped (`ALL`)

4
Seccomp profiles (e.g., `RuntimeDefault`) strictly limit allowed syscalls; `no_new_privs` flag enforced

5
Leading enforcement: formally verified zero privilege escalation paths, absolute denial of root execution, cryptographic attestation of process privileges

ENGINEERING INTERPRETATION

This criterion evaluates whether escalation 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

- 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.

4.3 / CAPABILITY FAMILY

Image Provenance and Distroless

SOURCE WEIGHT 15%

4.3.1
Supply Chain Integrity

MYTHOS-SEC-0012-EVAL-4.3.1

Supply Chain Integrity

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Image Provenance and Distroless	**4.3.1**	**15%**	**Evidence-led**

Are container images minimal, verified, and free of attack tools?

0
Uses public base images (`ubuntu`, `node`) with shells and package managers

1
Custom base images, but still contains full OS utilities

2
Distroless images (e.g., `gcr.io/distroless/node`) used, removing shells and OS utilities

3
Images signed (Cosign/Sigstore) and verified by the runtime before execution

4
SBOM generation automated for all images; CI/CD blocks deployment if vulnerabilities found

5
Leading integrity: formally verified image provenance, zero unpatched dependencies, cryptographic attestation of image contents bound to hardware

ENGINEERING INTERPRETATION

This criterion evaluates whether supply chain integrity 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

Hypervisor and Confidential Computing

SOURCE WEIGHT 15%

4.4.1
Memory Isolation

MYTHOS-SEC-0012-EVAL-4.4.1

Memory Isolation

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Hypervisor and Confidential Computing	4.4.1	15%	Evidence-led

Can the hypervisor or host administrator read the memory of the running VM?

0
Standard VMs where host memory is plaintext to the hypervisor

1
Basic memory encryption enabled, but lacking attestation

2
Confidential VMs (SEV-SNP/TDX) used for highly sensitive workloads

3
Remote attestation verified by the guest before booting application logic

4
Data at rest, in transit, and in-use (memory) fully encrypted; zero host access to plaintext

5
Leading confidentiality: formally verified hardware boundaries, cryptographic proof of memory isolation, zero ability for cloud provider to inspect tenant state

ENGINEERING INTERPRETATION

This criterion evaluates whether memory 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

- 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

Runtime Detection (eBPF)

SOURCE WEIGHT 15%

4.5.1
Threat Visibility

MYTHOS-SEC-0012-EVAL-4.5.1

Threat Visibility

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Runtime Detection (eBPF)	**4.5.1**	**15%**	**Evidence-led**

How are container and kernel threats detected in real-time?

0
No runtime security; relies solely on static image scanning

1
Host-based agent monitoring file integrity and process lists (high overhead)

2
eBPF-based detection (Falco/Tetragon) monitoring system calls

3
eBPF detection with automated response (e.g., instantly killing pods that attempt kernel exploits)

4
Behavioral profiling: eBPF rules tuned to detect container escape attempts and lateral movement

5
Leading visibility: formally verified syscall tracing, zero runtime overhead, mathematical proof of anomaly detection coverage

ENGINEERING INTERPRETATION

This criterion evaluates whether threat visibility 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

- telemetry coverage, detection source, test fixtures, version history, alerts, and case outcomes
- approved architecture or policy record
- current configuration or platform export

ASSESSMENT PROCEDURE

1. replay benign and malicious fixtures; measure fidelity and operational response
2. Examine authoritative design, policy, configuration, and lifecycle records.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- untested rules, missing telemetry, alert-only metrics, and undocumented suppression
- The control exists only in diagrams or policy text.

OPERATIONAL MEASURES

- true-positive rate
- false-positive burden
- coverage by technique
- mean time to contain
- 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

Network Microsegmentation

SOURCE WEIGHT 15%

4.6.1
Lateral Movement Prevention

MYTHOS-SEC-0012-EVAL-4.6.1

Lateral Movement Prevention

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Network Microsegmentation	**4.6.1**	**15%**	**Evidence-led**

Can a compromised container access the network of another container?

0
Flat overlay network; any container can talk to any container

1
Basic NetworkPolicies restricting ingress/egress by namespace

2
Strict default-deny NetworkPolicies applied cluster-wide

3
Service mesh (Istio/Linkerd) enforcing mTLS and Layer 7 authorization policies between pods

4
eBPF-based network security (Cilium) replacing standard kube-proxy with identity-aware enforcement

5
Leading segmentation: formally verified zero-trust east-west traffic, absolute denial of lateral movement, cryptographic proof of service identity

ENGINEERING INTERPRETATION

This criterion evaluates whether lateral movement 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

- zone and identity model, policy graph, enforcement exports, and east-west telemetry
- approved architecture or policy record
- current configuration or platform export

ASSESSMENT PROCEDURE

1. test intended and prohibited flows across normal and degraded conditions
2. Examine authoritative design, policy, configuration, and lifecycle records.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- VLAN-only assumptions, transitive access, broad service accounts, and undocumented bypass
- The control exists only in diagrams or policy text.

OPERATIONAL MEASURES

- policy coverage
- prohibited-flow success rate
- exception age
- 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 Virtualization & Container Suite (MVCS)

Traditional compute benchmarks measure startup time and density. The MVCS evaluates kernel isolation, privilege boundaries, and runtime threat response.

5.1 Suite Composition

5.1.1 MVCS-Escape

- **Kernel Exploit Injection:** 100+ tests executing known container escape payloads (e.g., dirty COW, CVE exploits) inside the container, verifying the host kernel and other tenants remain uncompromised.
- **Privilege Escalation:** 50+ tests attempting `sudo`, `setuid`, and capability exploitation, verifying the system blocks the escalation and kills the process.

5.1.2 MVCS-SupplyChain

- **Tampered Image Test:** 100+ tests attempting to deploy unsigned or modified images, verifying the runtime refuses to execute them.
- **Shell Injection:** 50+ tests attempting to spawn a shell (`/bin/sh`) inside a running container, verifying the distroless image lacks a shell and blocks the action.

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.