MRIS IMPLEMENTATION GUIDE

Implementation guideline for the thirteen Mythos Hardening Controls

Companion document to MRIS, Chapter 9 (Version 1.6) — Mythos-Resistant Information Security

ISO 27002 depth · Evidence-based · Practice-oriented

Version 1.2 | June 2026
Author: Richard Peddi

TARGET AUDIENCE

CISOs and ISMS officers who want to practically implement the thirteen "Mythos Hardening Controls" from MRIS, Chapter 9.

Purpose, Independence and Conventions

This document is the implementation layer for MRIS, Chapter 9. MRIS 1.6 lists the thirteen Mythos Hardening Controls (MHC) concisely in Annex A logic; this companion document provides for each MHC the implementation guideline in ISO 27002 logic — such that a CISO function can follow purpose, organisational and technical implementation, maturity level and evidence provision without additional specialist literature. It does not replace MRIS 1.6, but carries its catalogue into practice. The source references of this guide have already been brought to the verified current status (June 2026).

Fixed structure per MHC

- Header »At a glance«
- 1. Control statement
- 2. Purpose and threat linkage
- 3. Organisational implementation (CISO)
- 4. Technical implementation (IT, concludes with an effectiveness test)
- 5. Implementation examples (Example A / Example B)
- 6. Maturity path
- 7. Measurement and audit evidence
- 8. Typical mistakes
- 9. Delineation, residual risk and references.

The "Measurement and audit evidence" sections contained per MHC in this guide are deliberately kept lean. They are intended to enable practicable evidence provision without hampering implementation through excessive documentation requirements.

Organisations with an elevated level of audit, regulatory or customer requirements can voluntarily extend the evidence provision. For this, a supplementary audit grid can be used per applicable MHC:

- Control owner: responsible role or organisational unit
- Scope: affected systems, services, sites, platforms or data classes
- Applicability: rationale for why the MHC is applicable or not applicable
- Implementation status: planned, partially implemented, implemented, effectiveness-tested
- Evidence type: policy, configuration, log, report, test protocol, ticket, risk acceptance
- Test frequency: occasion and frequency of the effectiveness testing
- Sampling logic: sampling logic for large system landscapes
- Exceptions: documented deviations, exceptions and compensating measures
- Risk acceptance: accepted residual risks including the approving body
- KPI/KRI: metrics used and target values
- Last effectiveness review: date and result of the last effectiveness review

This extended grid is not a mandatory component of the MRIS Implementation Guide. It is intended as an optional supplement for organisations that need greater audit depth, for instance owing to regulatory requirements, customer audits, certification preparation or a higher ISMS maturity level. For organisations with a lower maturity level, the minimum evidence provision described per MHC suffices initially.

Licence and Disclaimer

Disclaimer

This guide constitutes a technical and organisational reference. It does not replace an individual risk analysis, legal advice or an audit-specific review. The assessment of individual controls presented here refers to the publicly documented state of agentic AI threats at the time of writing. The assessment may shift as new evidence emerges.

Recommended Citation

Peddi, Richard (2026): MRIS IMPLEMENTATION GUIDE, Version 1.2.

Contents

Prioritisation of the MHC — Threat Correlation and Roadmap

The thirteen MHC are not to be understood as a linear sequence from MHC-01 to MHC-13. Without prioritisation, every control appears equally urgent — and exactly that hampers the implementation decision. What is prioritised is not the control as such, but its contribution to reducing the threats most relevant to the organisation. For this, this chapter combines two complementary views: the measurable breadth of effect of each MHC (how many degraded controls it carries) and the threat- and effort-related ordering (threat correlation). The prioritisation remains compatible with the overarching MRIS principle: structural controls before new detection or automation capabilities.

At a glance

- Purpose: a traceable, threat-oriented implementation roadmap instead of a linear MHC sequence.
- Basis: breadth of effect (coverage) per MHC plus the Threat Priority Score.
- Result: standard tiering P0/P1/P2 as an adaptable template, with dependency rules.
- Important: MHC-01 and MHC-13 have coverage 0, but can be P0 depending on architecture and data situation.

1. Breadth of effect: how many degraded controls each MHC carries

The following overview shows how many of the controls degraded under Mythos (categories "partially degraded" and "friction-only", together 41 controls) each MHC flanks. The figures have been checked against the assessment matrix in MRIS, Annex A (overall distribution 29 robust / 37 partially degraded / 4 friction-only / 23 not affected). Since a degraded control can be flanked by several MHC, the column total (44) is greater than 41. The column "also robust" additionally names flanked, already robust controls — so several MHC also further harden robust controls beyond the degraded ones.

MHC	Degraded controls	Also flanking robust	Character
MHC-05 Behaviour-based detection	10	A.8.15	broadly cross-cutting
MHC-02 SBOM/build provenance	6	—	supply chain
MHC-03 Phishing-resistant MFA	5	—	identity/auth
MHC-04 Workload identity/Zero Trust	5	A.5.16, A.8.2	identity/access
MHC-09 AI security testing	5	A.8.29	secure development
MHC-11 SOAR/tier-1 automation	4	—	incident response
MHC-10 Continuous control monitoring	3	A.8.9	assurance
MHC-07 Multi-tenancy isolation	2	—	cloud
MHC-08 Immutable backups	2	A.5.30, A.8.13, A.8.14	resilience
MHC-06 Containers/confidential computing	1	A.8.31	narrow
MHC-12 Threat-led pentest	1	A.8.29	narrow, validating
MHC-01 Post-quantum	0	A.8.24	flanks robust only
MHC-13 AI agent governance	0	A.5.9, A.5.16, A.8.27	flanks robust only

MHC-05 is by far the broadest cross-cutting control. Effect is measurably unevenly distributed — that is the strongest argument against the perception that every MHC is equally important.

2. Two priority views and their relationship

- Breadth of effect (coverage): unconditional leverage — how many degraded controls an MHC carries. High coverage means high leverage across many risks at once.
- Threat and effort relation: the ordering — dependent on threat situation, impact, exposure, control gap, dependencies and effort.

The two views do not contradict each other; they complement each other. Coverage alone would place MHC-01 and MHC-13 at the end because they flank only robust controls. That would be wrong: MHC-13 is potentially P0 for every organisation with AI agents (it addresses the core of the Mythos threat landscape), MHC-01 for data with long-lived confidentiality. Hence: order by leverage, but with a threat and applicability gate — never by coverage alone. "Low hanging fruits" are then controls with high leverage, a large control gap and low effort, subject to the dependencies.

3. Assessment logic: Threat Priority Score

For the ordering, a simple, transparent score is formed per threat scenario. It makes visible whether an MHC is prioritised because of a high threat situation, high impact, strong exposure or a large control gap.

Threat Priority Score = Threat Relevance × Business Impact × Exposure × Control Gap

Factor	Guiding question	Scale
Threat relevance	How realistic, current and relevant is the attack scenario for the organisation?	1 = low … 5 = very high
Business impact	How severe would the consequences be for DORA/ BIA-critical services, customer data, availability, reputation or finances?	1 = low … 5 = existential
Exposure	How strongly is the organisation exposed (internet-facing, cloud, suppliers, admin access, external users)?	1 = low … 5 = strongly exposed
Control gap	How large is the gap relative to the MHC target state?	1 = well covered … 5 = barely covered

Example: AI-supported phishing with threat relevance 5, business impact 4, exposure 5 and control gap 4 yields a score of 400 — very high priority, primarily for MHC-03, flanked by MHC-05 and MHC-11.

The Threat Priority Score is a qualitative prioritisation index, not a replacement for a formal risk analysis (ISO/IEC 27005) or an organisation-specific risk matrix. Multiplying the factors makes relative action priorities visible; the value is ordinal, not metric — a score of 400 is not "twice as urgent" as 200, but serves ranking and triage.

4. Threat-to-MHC matrix

The matrix shows which MHC are particularly effective against which threat. It is suitable as a basis for the Statement of Applicability, the risk treatment plan and the implementation roadmap.

Threat / attack scenario	Primary MHC	Rationale	Priority
AI-supported phishing / credential theft	MHC-03, MHC-05,	MHC-03 prevents credential misuse, MHC-05 detects conspicuous use,	very high

Threat / attack scenario	Primary MHC	Rationale	Priority
	MHC-11	MHC-11 automates the first response.	
Lateral movement after initial compromise	MHC-04, MHC-05, MHC-11, MHC-12	MHC-04 limits lateral movement via workload identity, MHC-05 detects attack chains, MHC-12 validates effectiveness.	very high
Software supply chain attack	MHC-02, MHC-09, MHC-06	SBOM, build provenance, pipeline testing and image signing interlock.	high
Ransomware / destructive attack	MHC-08, MHC-05, MHC-11, MHC-12	Immutable backups secure recovery; detection and SOAR shorten the response time; TLPT/purple team tests resilience.	very high
Manipulated container image	MHC-06, MHC-02, MHC-09	Signed images and admission control require provenance and vulnerability information from the supply chain controls.	medium to high
Tenant breakout / cross-tenant access	MHC-07, MHC-04, MHC-06	Relevant for SaaS/multi-tenant platforms; the separation must be technically enforced and tested.	high, if applicable
AI agent misuse / prompt injection / tool abuse	MHC-13, MHC-04, MHC-10, MHC-09	Agents need their own identities, limited tool rights, policy control and a secure pipeline.	high, if AI agents are in use
Harvest now, decrypt later / quantum risk	MHC-01	Particularly relevant for data with long-lived confidentiality and crypto-agility.	medium to high
Undetected control deviations	MHC-10	Continuous control monitoring detects deviations continuously rather than only at points in an audit.	medium
False-positive security assumptions	MHC-12	Threat-led tests show whether documented controls actually work.	high after baseline implementation

5. Dependencies as roadmap rules

The following logic prevents controls from being implemented in isolation even though they only become effective, or meaningfully testable, through other MHC.

Dependency	Meaning
MHC-02 → MHC-09	AI-supported security testing in the pipeline needs a build/CI foundation and benefits directly from SBOM and component information.
MHC-04 → MHC-13	AI agent governance needs technical identities, capability-scoped access and a clear separation from personal user accounts.
Logging (A.8.15) → MHC-05 → MHC-11	SOAR automation is only resilient when detection is based on reliable, central and tamper-proof logs.
MHC-05 + MHC-11 → MHC-12	Threat-led tests validate whether detection and response work in realistic attack chains.

Dependency	Meaning
MHC-02 + MHC-04 → MHC-06	Container security becomes stronger when the provenance of the images and the workload identities are cleanly controlled.
MHC-04 + MHC-06 → MHC-07	Multi-tenancy isolation makes use of identity, network, resource and runtime separation.

6. Standard prioritisation P0 / P1 / P2

The following tiering is an example for typical enterprise environments. It must be reconciled with the actual threat situation, asset criticality and applicability. MHC-01 and MHC-13 formally sit in P2, but move up for data with long-lived confidentiality (MHC-01) or productive AI agent use (MHC-13). The standard tiering is not a universal implementation plan, but a starting point; it must be adapted per organisation to exposure, architecture, asset criticality, existing controls, resource situation and regulatory context.

Priority	MHC	Control	Rationale
P0 — immediately	MHC-03	Phishing-resistant MFA	High likelihood, fast risk reduction for admins and external access.
P0 — immediately	MHC-08	Immutable backups and recovery validation	Baseline capability against ransomware and destructive attacks.
P0 — immediately	MHC-05	Behaviour-based detection and kill chain correlation	Without detection there is no resilient response and no meaningful SOAR automation.
P0 — immediately	MHC-04	Workload identity and Zero Trust	Reduces lateral movement and is an enabler for AI agents.
P1 — afterwards	MHC-02	SBOM and build provenance	Foundation for supply chain transparency and pipeline testing.
P1 — afterwards	MHC-09	AI-supported security testing	Effective where a pipeline and an SBOM base exist.
P1 — afterwards	MHC-11	SOAR tier-1 automation	Automate only once detection is stable.
P1 — afterwards	MHC-10	Continuous control monitoring	Cross-cutting control for ongoing deviation detection.
P2 — risk-based	MHC-06	Container security and confidential computing	Highly relevant for container/cloud platforms.
P2 — risk-based	MHC-07	Multi-tenancy isolation	Directly applicable only for multi-tenant platforms or SaaS operations.
P2 — risk-based	MHC-13	AI agent governance	Highly relevant where AI agents, tool calling or automation are used; then promote above P0.
P2 — risk-based	MHC-01	Post-quantum strategy	Prioritise higher for data with long-lived confidentiality.
P2 — risk-based	MHC-12	Threat-led penetration testing	Validation after baseline implementation, thereafter regularly.

7. Procedure in four steps

- Step 1 — Create the threat landscape: identify relevant attacker types, TTPs and attack scenarios; consider industry-specific threats (financial services, critical services, supply chain, cloud, AI); rate threats by currency, likelihood and proximity to the organisation.
- Step 2 — Map threats to MHC: per threat, determine which MHC act preventively, detectively, reactively or in a validating role; weight controls with multiple threat linkages higher; cleanly justify non-applicable controls in the Statement of Applicability.
- Step 3 — Calculate the score: rate threat relevance, business impact, exposure and control gap each from 1 to 5; form the score and factor in effort and dependencies; for equal risk, start with quick wins offering high risk reduction and low dependency.
- Step 4 — Derive the roadmap: P0 (immediate measures against high threats and large gaps), P1 (controls with dependencies or foundational technical work), P2 (context-dependent controls and validation); reassess regularly as soon as the threat situation, architecture or regulation changes.

8. Wording for governance documents

For the risk-based prioritisation of the Mythos Hardening Controls, the MHC are correlated with relevant threat scenarios. For each threat, it is assessed which controls act preventively, detectively, reactively or in a validating role. The prioritisation results from threat relevance, business impact, exposure, the existing control gap, dependencies and implementation effort. This produces not a linear MHC sequence, but a threat-oriented roadmap that justifies for each control why it is implemented at its point in time.

MHC-01 — Post-Quantum Strategy and Cryptographic Inventory

At a glance

- Operational benefit: Protects data with long-lived confidentiality against the "harvest now, decrypt later" pattern and creates the ability to swap cryptographic methods swiftly when needed.
- Affected policy: Cryptography policy (encryption and key management).
- Dependencies: no precondition; presupposes an understanding of the encryption in use (inventory).
- Effort drivers: depends on the number of systems and data flows with encryption and on the replaceability of the libraries used. The concrete effort depends on the organisation's resources.
- Primary metric: share of the cryptographic methods in use that are captured in the central inventory and classified by migration priority.
- Standards touched (without claim of fulfilment): ISO/IEC 27002 A.8.24; NIST FIPS 203/204/205; BSI TR-02102; C5:2026 CRY-01; EU recommendation on PQC migration.

1. Control statement

The organisation maintains an inventory of all cryptographic methods in use and pursues a documented strategy to migrate to quantum-safe methods in good time. Hybrid methods are used during the transition.

2. Purpose and threat linkage

Future quantum computers will be able to break encryption that is widespread today (such as RSA and elliptic curves). Attackers therefore already intercept encrypted data today in order to decrypt

it later — the "harvest now, decrypt later" pattern. Particularly affected is data that must remain confidential over many years. AI additionally accelerates the discovery and exploitation of weak or outdated crypto implementations. The protection objective is to migrate data with long-lived confidentiality to quantum-safe methods in good time and to create the ability to switch methods swiftly when needed.

3. Organisational implementation (CISO perspective)

- Policy: The cryptography policy is supplemented with the obligation of a central crypto inventory and a migration strategy towards quantum-safe methods.
- Responsibilities: A role is named for maintaining the inventory and updating the strategy, and assigned in the Statement of Applicability (template MRIS Annex H).
- Risk analysis and SoA: The migration order is derived from the risk assessment — data with a long confidentiality duration first. Risk treated: later decryption of data exfiltrated today (linkage to the risk assessment bridge, MRIS Annex F).
- Processes: regular (annual or event-driven) review of the inventory and strategy; observation of the relevant standards; defined triggers, means and transition paths for the migration.
- Procurement: for new systems and contracts, support for quantum-safe or replaceable methods is included as a requirement.

4. Technical implementation (for IT specialists)

- Crypto inventory: capture of all algorithms, key lengths, protocols, certificates and the underlying libraries in use; prioritisation by protection needs and confidentiality duration.
- Quantum-safe methods: migration to the finalised NIST standards — FIPS 203 (ML-KEM) for key exchange and FIPS 204 (ML-DSA) and FIPS 205 (SLH-DSA) for signatures. HQC is in standardisation as a supplementary key exchange (backup), FN-DSA (FALCON) as a further signature in preparation.
- Hybrid methods: during the transition, classical and quantum-safe methods are combined (e.g. in the TLS key exchange) so that the connection remains secure as long as at least one of the two methods holds.
- Crypto-agility: encryption is encapsulated so that methods can be swapped without deep interventions in the applications (central crypto libraries, configurable algorithms).
- Deep dive: migration approach in NIST SP 1800-38 (Migration to Post-Quantum Cryptography, NCCoE); method and key length recommendations in BSI TR-02102; algorithm specification in FIPS 203/204/205.
- Effectiveness test: For a prioritised system it is verified that the method actually in use matches the inventory and the target specification (e.g. the negotiated key exchange in the TLS handshake) and that a method can be switched without code changes.

5. Implementation examples

Example A — development/platform context. A software vendor initially does not know precisely which encryption is actually in use in its services and libraries. It therefore first creates a central register of all methods, keys and certificates in use, and records for each data set how long it must remain confidential. For the connections carrying particularly long-lived data, it switches the key exchange to a combined method that uses classical and quantum-safe cryptography at the same time. That way, this data remains protected even if the classical method is later broken by quantum computers; at the same time, the method can be switched centrally when needed.

Example B — general department. An organisation without in-house development mainly uses off-the-shelf software and cloud services. It cannot switch the encryption itself, but gains an overview

of where data with particularly long-lived confidentiality resides (such as personnel or contract documents) and asks its providers about their roadmap for the quantum-safe migration. In new contracts, it includes support for quantum-safe methods as a requirement. It thus manages the topic via oversight and procurement, even without its own technical implementation.

6. Maturity path (cumulative)

- Initial: cryptographic inventory documented.
- Defined: migration strategy adopted; hybrid methods trialled in a pilot area.
- Managed: hybrid or quantum-safe methods in productive use; annual review automated.

7. Measurement and audit evidence

- Metric: share of methods in use that are captured in the inventory and classified by migration priority (target value: complete capture of the areas worth protecting).
- Evidence: crypto inventory, migration strategy with triggers and timeline, pilot and production evidence of hybrid methods.
- Audit logic per ISO/IEC 27005: documented? — inventory and strategy; responsible? — named role; frequency? — annual or event-driven review; tolerance? — no data with long-lived confidentiality without a migration plan.

8. Typical mistakes

- A strategy is formulated without a complete inventory in place — the most important spots remain unknown.
- Hybrid methods are introduced across the board without prioritising by confidentiality duration — a lot of effort without a recognisable focus of benefit.
- Encryption is hard-wired in the code, so that a later switch of the method requires interventions in the application each time.
- The migration is treated purely as a future topic, even though today's data leakage already determines tomorrow's risk.

9. Delineation, residual risk and references

- Delineation: concerns the cryptography in use; the management of the keys themselves remains part of general cryptography practice (A.8.24).
- Residual risk: as long as classical methods run in parallel, the risk remains that data already exfiltrated will be decrypted later; quantum-safe methods are moreover comparatively young and are kept under observation.
- ISO/IEC 27002:2022: A.8.24 Use of cryptography.
- Framework: NIST FIPS 203 (ML-KEM), FIPS 204 (ML-DSA), FIPS 205 (SLH-DSA); NIST SP 1800-38 (migration guide, NCCoE); BSI TR-02102; C5:2026 CRY-01; EU recommendation on the coordinated PQC migration.

MHC-02 — SBOM and Build Provenance

At a glance

- Operational benefit: Makes immediately visible which third-party components are contained in one's own software, so that when a new vulnerability emerges it is clear in minutes rather than days whether and where one is affected.
- Affected policy: Policy on secure software development and supplier management.
- Dependencies: no precondition; presupposes an automated build/CI pipeline to realise the full benefit.

- Effort drivers: depends on the number and structure of the build pipelines and the depth of the supply chain. The concrete effort depends on the organisation's resources.
- Primary metric: share of shipped artefacts with a current, automatically generated bill of materials (SBOM coverage).
- Standards touched (without claim of fulfilment): ISO/IEC 27002 A.5.21/A.8.30; CRA Annex I; BSI TR-03183-2; C5:2026 DEV-13; SLSA.

1. Control statement

For every piece of software created and distributed in-house, a bill of the contained components (SBOM) is automatically generated and reconciled against vulnerability databases. For critical artefacts, it is additionally demonstrably recorded how and from what they were built (build provenance).

2. Purpose and threat linkage

Modern software largely consists of third-party building blocks (open-source libraries, frameworks). If a vulnerability becomes known in one of these building blocks — or an attacker deliberately injects malicious code into the supply chain —, without an overview it often remains unclear for days whether and where one is affected. AI drastically shortens the time between the disclosure of a vulnerability and a working attack. A current bill of materials makes one's own exposure immediately verifiable; evidence of origin and build method makes it harder to slip in manipulated building blocks unnoticed. The protection objective is to detect and contain attacks via the software supply chain quickly.

3. Organisational implementation (CISO perspective)

- Policy: The development and supplier policy prescribes automatic SBOM generation for own and distributed artefacts, as well as the contractual SBOM requirement for critical software suppliers.
- Responsibilities: In the Statement of Applicability, the measure is assigned to a role (template MRIS Annex H); development and procurement work together.
- Risk analysis and SoA: critical artefacts and suppliers are prioritised from the risk assessment. Risk treated: undetected vulnerabilities or manipulations in third-party components (linkage to the risk assessment bridge, MRIS Annex F).
- Processes: a fixed procedure for determining and remedying exposure via the bills of materials when a new vulnerability becomes known; retention of the bills of materials per software version.
- Procurement: SBOM and, where possible, provenance evidence are included as requirements in contracts with critical suppliers.

4. Technical implementation (for IT specialists)

- SBOM generation: automatically in the build/CI pipeline, in an established, machine-readable format — CycloneDX (standardised as ECMA-424) or SPDX (standardised as ISO/IEC 5962). The indirect dependencies (transitive) are captured too, not only the top-level ones.
- One SBOM per version: whenever a component changes, a new version with its own bill of materials is generated; bills of materials are retained in versioned form.
- Vulnerability reconciliation: automated correlation of the components against vulnerability databases (CVE/NVD, OSV, ENISA's EUVD). Vulnerability data does not belong in the SBOM itself, but is communicated separately via VEX or CSAF.

- Build provenance: for critical artefacts, it is recorded traceably and tamper-proof from which sources and with which build process they were created (SLSA build provenance, higher level for critical artefacts).
- Tools (non-binding examples): generation with Syft, cdxgen or Trivy; central management and vulnerability correlation e.g. with Dependency-Track.
- Deep dive: build provenance levels in the SLSA framework; SBOM quality requirements in BSI TR-03183-2; legal framework in CRA Annex I.
- Effectiveness test: For a real, known vulnerability in a widespread library (e.g. a Log4j-style flaw), it is checked solely via the bills of materials whether and which own artefacts contain the affected component — the result must be available in minutes.

5. Implementation examples

Example A — development/platform context. A software vendor learns from the news of a severe vulnerability in a widely used library. Until now it has had to laboriously search all projects individually, which takes days. It therefore extends its build pipeline so that every build automatically produces a complete bill of all contained building blocks, continuously reconciled against vulnerability databases. At the next such incident, a single query across the bills of materials suffices to see in minutes which products contain the affected component. For its most important artefacts it additionally deposits tamper-proof evidence of what they were built from and how.

Example B — general department. An organisation without in-house development cannot generate SBOMs itself, but depends on purchased software. It therefore includes in its contracts with the most important software suppliers the requirement to provide a current bill of materials with every delivery. When a new vulnerability becomes known, it can then follow up with the manufacturer specifically on whether the deployed product is affected, instead of waiting for vague blanket notices. It thus uses the principle via procurement, even without its own pipeline.

6. Maturity path (cumulative)

- Initial: SBOM generation in individual pipelines.
- Defined: SBOM for all own artefacts; automated vulnerability reconciliation.
- Managed: SBOM also from critical suppliers; build provenance (SLSA) for critical artefacts; automated verification of the provenance evidence.

7. Measurement and audit evidence

- Metric: SBOM coverage — share of shipped artefacts with a current, automatically generated bill of materials (target value: complete coverage of own artefacts).
- Evidence: SBOM files per version, pipeline configuration, vulnerability reconciliation reports, provenance evidence of critical artefacts.
- Audit logic per ISO/IEC 27005: documented? — pipeline configuration and bills of materials; responsible? — development/product owners; frequency? — with every build, reconciliation ongoing; tolerance? — no shipped artefact without a current bill of materials.

8. Typical mistakes

- Bills of materials are generated but never evaluated — in a vulnerability case an SBOM exists, but nobody reconciles it.
- Only the top-level dependencies are captured; precisely the deeply nested building blocks in which vulnerabilities reside are missing.
- The SBOM is created once and not updated per version, so that it does not match the state actually shipped.

- Vulnerability data is mixed into the SBOM; this blends the static bill of materials with dynamic information and quickly makes it unusable.

9. Delineation, residual risk and references

- Delineation: concerns transparency and provenance of the software building blocks; the actual closing of the vulnerabilities is part of vulnerability and patch management, automated testing is covered by MHC-09.
- Residual risk: a bill of materials detects known but not unknown vulnerabilities; correctly declared but manipulated components are not exposed by the SBOM alone — provenance serves that purpose.
- ISO/IEC 27002:2022: A.5.19 Information security in supplier relationships; A.5.20 Addressing information security within supplier agreements; A.5.21 Managing information security in the ICT supply chain; A.8.4 Access to source code; A.8.30 Outsourced development.
- Framework: C5:2026 DEV-13; CRA Annex I (SBOM obligation; vulnerability reporting obligations from 2026, main obligations for products placed on the market after 11 December 2027); BSI TR-03183-2; DORA Art. 28; NIS2 IR No. 5; SLSA framework; formats CycloneDX (ECMA-424) and SPDX (ISO/IEC 5962).

MHC-03 — Phishing-Resistant Multi-Factor Authentication

At a glance

- Operational benefit: Renders intercepted or captured credentials worthless on fake pages and thereby removes the basis of AI-supported, deceptively genuine phishing.
- Affected policy: Policy on access control and authentication.
- Dependencies: no precondition; works well together with MHC-04 (privileged access).
- Effort drivers: depends on the number of users, services and legacy systems and on the existing identity platform. The concrete effort depends on the organisation's resources.
- Primary metric: share of logins via phishing-resistant methods, particularly for privileged and externally reachable access.
- Standards touched (without claim of fulfilment): ISO/IEC 27002 A.5.17/A.8.5; NIST SP 800-63B Revision 4; FIDO2/WebAuthn; C5:2026 IAM-08; NIS2 IR No. 11.2.

1. Control statement

Privileged access, externally reachable services and access to particularly sensitive systems are secured with phishing-resistant methods (FIDO2/WebAuthn, passkeys or hardware tokens). SMS and simple confirmation methods are excluded for new access.

2. Purpose and threat linkage

Phishing has risen sharply in quality and volume through AI: fake login pages, deceptively genuine emails and codes relayed in real time bypass classic two-factor methods such as SMS or app confirmation. Phishing-resistant methods cryptographically bind the authentication credential to the genuine address of the service; on a fake page the credential is therefore worthless. The protection objective is that intercepted or captured credentials are of no use to the attacker.

3. Organisational implementation (CISO perspective)

- Policy: The authentication policy prescribes phishing-resistant methods for privileged, externally reachable and particularly sensitive access; SMS and simple push methods are excluded for new access, legacy methods are replaced with a transition plan.

- Responsibilities: The migration is assigned to a role in the Statement of Applicability (template MRIS Annex H).
- Risk analysis and SoA: The target maturity level is derived from the risk assessment; privileged and externally reachable access first.
- Processes: issuance and withdrawal of the tokens or passkeys as a regulated process; replacement procedures for loss, without undermining the protection level; decommissioning of the legacy methods after the transition.
- Training: users are supported during setup and use of the new methods to ensure acceptance and a smooth transition.

4. Technical implementation (for IT specialists)

- Phishing-resistant methods: FIDO2/WebAuthn with passkeys or hardware security keys; the cryptographic binding to the service's domain (origin binding) prevents relaying to fake pages.
- Protection levels per NIST SP 800-63B Revision 4: at level AAL2, synchronisable passkeys (across multiple devices) are also permitted; the highest level AAL3 requires a device-bound, non-exportable key (hardware token or smartcard), and synchronisable passkeys are not permitted there.
- Switch-off of weak methods: SMS codes and simple app confirmations without number matching are no longer permitted for new access; existing methods are replaced.
- Integration: enforcement via the central identity platform and conditional access; applications are connected via the identity platform rather than via their own login masks.
- Deep dive: requirements per protection level (AAL2/AAL3) in NIST SP 800-63B Revision 4; method details in the FIDO2/WebAuthn specifications (FIDO Alliance, W3C WebAuthn).
- Effectiveness test: A controlled phishing attempt with a replicated login page is conducted; the phishing-resistant credential must not be usable on the fake page (the login fails).

5. Implementation examples

Example A — development/platform context. A company has so far secured its administrators' and developers' access with a password and a second factor confirmed via app. A well-crafted phishing attack intercepts both password and confirmation in real time. The company first migrates all privileged and externally reachable access to hardware security keys whose credential is cryptographically bound to the genuine address of the service. On a fake login page this credential is useless. In a further expansion stage, login becomes passwordless organisation-wide, and the old SMS and app methods are switched off.

Example B — general department. An organisation without in-house development uses Microsoft 365 and a few other cloud services. Employees have so far logged in with a password and SMS code — susceptible to phishing. The organisation activates passkey login in its identity platform and issues hardware security keys for particularly sensitive roles. Login henceforth takes place without a password and without SMS; a central access rule explicitly requires a phishing-resistant method for sensitive applications. Protection can thus be enforced via the settings of the platform in use, even without in-house development.

6. Maturity path (cumulative)

- Initial: FIDO2/passkeys for privileged access.
- Defined: phishing-resistant methods mandatory for all externally reachable services.
- Managed: passwordless login organisation-wide; SMS methods switched off.

7. Measurement and audit evidence

- Metric: share of logins via phishing-resistant methods, separated by privileged, externally reachable and other access (target value: complete for privileged and externally reachable access).
- Evidence: identity platform configuration, list of permitted methods per access class, evidence of the switch-off of SMS methods.
- Audit logic per ISO/IEC 27005: documented? — authentication policy and platform configuration; responsible? — identity/IT management; frequency? — ongoing; tolerance? — no privileged or externally reachable access with SMS or simple push confirmation only.

8. Typical mistakes

- Phishing-resistant methods are introduced, but weaker methods remain active as a fallback path — the attacker uses the weakest path.
- Synchronisable passkeys are used for the highest protection level, even though a device-bound, non-exportable key is required there.
- The replacement procedure for loss (such as reset via phone call) undermines the protection level.
- Only individual applications are migrated, while privileged access remains more weakly secured.

9. Delineation, residual risk and references

- Delineation: concerns the users' authentication credential; the identity of services and workloads is covered by MHC-04.
- Residual risk: attacks can shift to the recovery and fallback procedures; session takeovers after successful login also remain a separate topic.
- Effectiveness limit: phishing-resistant does not mean phishing-proof. FIDO2/passkeys devalue classic credential phishing and MFA fatigue, but not session or token theft via malware, adversary-in-the-middle after login, or social engineering at the helpdesk.
- ISO/IEC 27002:2022: A.5.17 Authentication information; A.8.5 Secure authentication; A.6.7 Remote working; A.8.1 User endpoint devices; A.8.4 Access to source code; A.8.23 Web filtering.
- Framework: NIST SP 800-63B Revision 4 (protection levels AAL2/AAL3); FIDO2/WebAuthn (FIDO Alliance, W3C); C5:2026 IAM-08; NIS2 IR No. 11.2.

MHC-04 — Workload Identity and Zero Trust Network Architecture

At a glance

- Operational benefit: Prevents an attacker's lateral spread after an initial compromise; a taken-over service can no longer move on to neighbouring services.
- Affected policy: Access control and network security policy.
- Dependencies: forms the identity foundation for MHC-13; itself without precondition.
- Effort drivers: migrating existing services binds more resources than a greenfield build; the concrete effort depends on the organisation's available resources.
- Primary metric: share of productive service-to-service connections with workload identity and mTLS.
- Standards touched (without claim of fulfilment): inter alia ISO/IEC 27002 A.8.20/A.8.21/A.8.22, NIST SP 800-207 and 800-207A, NIS2 IR No. 6.7/8/11.

1. Control statement

Every workload (service, process, container, AI agent) receives a cryptographically verifiable, short-lived and non-reusable identity. Communication between services is authorised on the basis of this identity, not on the basis of network position (IP address, subnet, perimeter).

2. Purpose and threat linkage

AI-accelerated attackers move from initial compromise to lateral movement within minutes. Classic perimeter, VPN and jump host models trust a service solely because of its network position; a taken-over host thereby effectively gains access to the neighbouring services. A dedicated identity per workload removes the basis of this pattern: without a valid, short-lived identity, no connection is established. The objective is to structurally close the classic paths for lateral spread. The protection rests on a cryptographic barrier, not on an attack merely being delayed.

3. Organisational implementation (CISO perspective)

- Policy: The access control and network security policy is supplemented with the principle that productive services identify themselves via their own, technically managed identity; trust based purely on network addresses is permitted only temporarily and with documentation.
- Responsibilities: The measure is assigned to a named role in the Statement of Applicability (template MRIS Annex H: IT management responsible for the build-up, development responsible for migrating the services, CISO accountable).
- Risk analysis and SoA: The target maturity level (Initial/Defined/Managed) is derived from the risk assessment; services with access to sensitive data first. Risk treated: lateral spread after an initial compromise (linkage to the risk assessment bridge, MRIS Annex F).
- Processes: A documented process is needed for issuing, regularly renewing and revoking the service identities; the time to revocation of an identity is tracked as a metric.
- Training: development teams are bound by the changed requirement (no static access keys in code) and instructed accordingly.
- AI agents: It is stipulated that AI agents operate under their own, technically managed identity and not under personal user accounts; the detailed regulation follows in MHC-13.

4. Technical implementation (for IT specialists)

- Workload identity with SPIFFE/SPIRE: Every workload receives a SPIFFE ID, issued as a short-lived X.509 SVID or JWT SVID by a SPIRE control plane (server) with one SPIRE agent per node. The workload is attested before issuance via platform selectors (e.g. Kubernetes service account, node attestation). SPIFFE/SPIRE are a CNCF-graduated (production-ready) project.
- mTLS: Both endpoints mutually authenticate via their SVIDs on the basis of TLS 1.3. Short certificate lifetimes (minutes to hours) with automatic rotation; long-lived shared secrets are eliminated.
- Service mesh (e.g. Istio, Linkerd): One sidecar proxy per service handles mTLS, identity verification and policy enforcement transparently to the application code. Authorisation occurs via identity policies (which service identity may call which service) instead of via IP allowlists.
- Segmentation: Identity-based authorisation with default deny between services supplements or replaces network-based micro-segmentation.
- External and privileged access: ZTNA or identity-aware proxy instead of classic VPN; access is bound to user identity, device identity and context.

- Migration: Permissive mode (plaintext and mTLS in parallel) serves only as a short measurement period; thereafter strict mode is enforced. Order: the most critical service-to-service paths first.
- AI agents: time-limited identities limited to the necessary capabilities (capability-scoped) per tool call instead of developer credentials.
- Deep dive: architecture model (identity tier vs. network tier, ingress/egress gateways, multi-cloud) in NIST SP 800-207A, Sections 3–4; Zero Trust core principles in NIST SP 800-207, Section 2; SPIFFE specification (SPIFFE ID, X.509 SVID, JWT SVID) at spiffe.io.
- Effectiveness test: From a service or host without a valid identity, a connection to a protected service is attempted — it must be rejected. Additionally: an expired credential must no longer enable a connection.

5. Implementation examples

Example A — development/platform context. A mid-sized software vendor operates its application from many individual, interconnected services. Until now, these services trusted each other solely because they ran in the same internal network; if an attacker took over one service, it could reach all others unhindered from there. The company changes this foundation: every service receives its own, constantly changing technical credential, and without a valid credential no other service accepts a connection. It begins with the few services that process customer data; later the migration is extended to all. The result: a taken-over service stands alone and can no longer move sideways to the neighbouring services.

Example B — general department. An organisation without in-house software development operates a few internal applications, such as a file server, and has so far granted access via the internal network and a VPN: whoever is on the network is considered trustworthy. That harbours the same fundamental problem — a taken-over device on the network gains far-reaching access. The organisation henceforth binds access to internal applications to the verified identity of user and device, no longer to mere network membership; an upstream access service checks login, device and situation on every access. It begins with the most sensitive applications and the administrative access paths. *Applicability note:* An organisation that exclusively uses off-the-shelf cloud services and operates no infrastructure of its own is barely affected by MHC-04; in the Statement of Applicability the control is recorded as not applicable. Here the provider is responsible for the protection, which is demanded contractually via supplier management.

6. Maturity path (cumulative; critical paths first, no big bang)

- Initial: SPIFFE/SPIRE for privileged workloads; mTLS on the 3–5 most critical service-to-service paths.
- Defined: workload identity for all productive microservices; ZTNA for privileged remote access.
- Managed: across the board including dev/test; identity-based micro-segmentation; AI agents with capability-scoped identities.

7. Measurement and audit evidence

- Metrics: share of productive service-to-service connections with mTLS/workload identity (target value Defined: 100 % productive); time to revocation of a workload identity.
- Evidence: SPIRE registry (inventory of the issued identities), service mesh policy configuration, logs of the mTLS enforcement.
- Audit logic per ISO/IEC 27005: documented? — via the SPIRE registry; responsible? — platform team; frequency? — ongoing (renewal at minute-to-hour cadence); tolerance? — no productive shared secrets stored in plaintext on critical paths.

8. Typical mistakes

- mTLS remains in permissive mode (plaintext is still accepted) — no protective effect.
- Identities with too long a validity (days instead of minutes) — the benefit of the short lifetime is lost.
- AI agents continue to operate under personal developer accounts — traceability and limitation of rights are missing.
- IP-based approvals as a permanent parallel path instead of a temporary, documented exception.

9. Delineation, residual risk and references

- Delineation: protects the service-to-service level; end-user authentication is covered by MHC-03; the governance of the AI agents is deepened by MHC-13.
- Residual risk: a taken-over issuing infrastructure (SPIRE) or a legitimately credentialed but taken-over service remains effective; an identity does not replace the minimisation of rights (least privilege remains required).
- Effectiveness limit: Zero Trust is an architectural path, not a product. Only partially implemented segmentation or workload identity can create false security; effectiveness arises only when authentication, authorisation and least privilege apply consistently and verifiably.
- ISO/IEC 27002:2022: A.5.15 Access control; A.5.16 Identity management; A.6.7 Remote working; A.8.2 Privileged access rights; A.8.20 Networks security; A.8.21 Security of network services; A.8.22 Segregation of networks.
- Framework: NIST SP 800-207 (Zero Trust principles); NIST SP 800-207A (cloud-native Zero Trust, workload identity, service mesh); SPIFFE/SPIRE (CNCF).

MHC-05 — Behaviour-Based Detection and Kill Chain Correlation

At a glance

- Operational benefit: Detects attackers by their behaviour and by the chain of connected steps, rather than only by known signatures — and thus also catches camouflaged attacks conducted with legitimate means.
- Affected policy: Security monitoring policy (logging, monitoring, detection).
- Dependencies: presupposes central, tamper-proof logging (A.8.15); supplies the signals for the automation in MHC-11.
- Effort drivers: depends on the scope of the log sources, the existing SIEM/detection platform and the available analysis capacity. The concrete effort depends on the organisation's resources.
- Primary metric: coverage of the relevant attack techniques per MITRE ATT&CK (share of the most important techniques with effective detection).
- Standards touched (without claim of fulfilment): ISO/IEC 27002 A.8.16/A.5.7; MITRE ATT&CK; C5:2026 OPS-13; NIS2 IR No. 3.2; DORA Art. 10.

1. Control statement

Attacks are detected by behavioural anomalies and the linkage of connected events, not solely by individual known signatures. Detection is oriented towards a recognised catalogue of attack techniques (MITRE ATT&CK) with measurable coverage; the targeted search for as-yet-undetected attacks (threat hunting) is an established function.

2. Purpose and threat linkage

AI-supported attackers proceed inconspicuously: they use on-board system tools instead of imported malware and camouflage their command channels, so that every single step appears harmless in itself. Classic, signature-based detection does not trigger. Moreover, agentic attackers decompose their approach into many small actions, each unsuspicious on its own — the danger only shows itself in context. The protection objective is to detect attacks by their behaviour and by the sequence of connected steps, even when no known signature exists.

3. Organisational implementation (CISO perspective)

- Policy: The monitoring policy stipulates that detection is behaviour- and technique-based (with reference to MITRE ATT&CK), with measurable coverage and regular review.
- Responsibilities: Detection operations and threat hunting are assigned in the Statement of Applicability (template MRIS Annex H); threat hunting is anchored as a dedicated function with minimum capacity.
- Risk analysis and SoA: The most important techniques to detect are derived from the threat situation and the risk assessment. Risk treated: camouflaged attacks conducted with legitimate means that remain below the threshold of individual alerts.
- Processes: documented threat hunting methodology (e.g. PEAK or TaHiTI) with regular, hypothesis-driven runs; regular review and extension of the detection coverage; regulated handover of detected incidents into the response.
- Capacity: sufficient analysis capacity for hunting and evaluation; orientation of security operations towards analysis rather than pure alert review.

4. Technical implementation (for IT specialists)

- Behaviour-based detection: evaluation of user and system behaviour against a learned baseline (UEBA), with alerting on deviations.
- Technique coverage per MITRE ATT&CK: detection rules are aligned with the techniques of the ATT&CK catalogue (continuously updated, currently v19) instead of isolated individual signatures; coverage is measured (e.g. with DeTT&CT) and confirmed by controlled attack tests (e.g. Atomic Red Team). Guideline: the most important widespread techniques reliably covered.
- Kill chain correlation: individual events are linked across time and systems into an attack chain, so that a sequence of steps each inconspicuous on its own stands out as a whole.
- Detection focuses: use of on-board tools (e.g. conspicuous script or task scheduler activity, unusual system processes) and camouflaged command channels (e.g. unusual DNS or encrypted traffic with long dormant phases).
- Robustness of one's own detection AI: if detection itself relies on machine learning, the models are tested against deception (adversarial examples) and data poisoning.
- Deep dive: technique catalogue and detection guidance in MITRE ATT&CK (Enterprise); background on detection systems in NIST SP 800-94 (2007 edition; no revised edition exists).
- Effectiveness test: A multi-stage, controlled attack composed of steps each inconspicuous on its own (e.g. via Atomic Red Team) is conducted; the chain must be detected as a connected incident, not merely as individual, unconnected events.

5. Implementation examples

Example A — development/platform context. A company with its own security operations discovers that its previous detection only triggers on known malware. An attacker who exclusively uses on-board system tools remains undetected. The company therefore aligns its detection rules with

a recognised catalogue of attack techniques and measures what share of the most important techniques is actually covered. Suspicious sequences of steps are linked across time and systems into a chain. In addition, a small team regularly and deliberately searches for traces of attacks not yet detected. Now even an attacker moving quietly with legitimate means stands out.

Example B — general department. An organisation without its own 24/7 operations cannot build such detection itself. It therefore procures security monitoring as a service (managed detection & response) and, when selecting, ensures that the provider detects behaviour- and technique-based, evaluates connected attack chains and contractually assures a fixed response time. Internally it names a contact person who receives the reported incidents and steers the remediation. It thus achieves a comparable protection level via a service provider.

6. Maturity path (cumulative)

- Initial: MITRE ATT&CK introduced as the detection framework; low coverage.
- Defined: the most important widespread techniques reliably covered; documented threat hunting.
- Managed: deception-robust, learning detection; ongoing review and confirmation of coverage.

7. Measurement and audit evidence

- Metric: coverage of the most important attack techniques per MITRE ATT&CK (measured and confirmed by attack tests); additionally the number and results of the threat hunts conducted.
- Evidence: coverage overview (e.g. DeTT&CT), results of the attack tests, documented hunting runs with technique mapping.
- Audit logic per ISO/IEC 27005: documented? — monitoring policy, coverage overview, hunting protocols; responsible? — detection/SOC management; frequency? — ongoing operations, regular hunts and coverage checks; tolerance? — no permanently unobserved important techniques.

8. Typical mistakes

- Detection remains signature-based; attacks with on-board tools trigger no alert.
- The ATT&CK coverage is asserted but never verified through controlled attack tests — detection fails when it matters.
- Individual alerts are worked through but never linked into chains; the connections remain invisible.
- Threat hunting is treated as a side task and, for lack of capacity, is de facto not carried out.

9. Delineation, residual risk and references

- Delineation: concerns the detection of attacks; the subsequent automated response is covered by MHC-11, the automated testing of one's own defences by MHC-09.
- Residual risk: detection is not prevention — very slow, heavily camouflaged or novel attacks can remain undetected; where detection relies on machine learning, the targeted deception of the models remains a residual risk.
- ISO/IEC 27002:2022: A.5.3 Segregation of duties; A.5.7 Threat intelligence; A.5.14 Information transfer; A.5.15 Access control; A.8.1 User endpoint devices; A.8.3 Information access restriction; A.8.4 Access to source code; A.8.7 Protection against malware; A.8.12 Data leakage prevention; A.8.16 Monitoring activities.

- Framework: MITRE ATT&CK (Enterprise, continuously updated); C5:2026 OPS-13; NIS2 IR No. 3.2; DORA Art. 10; NIST SP 800-94 (2007 edition); methodology examples PEAK and TaHiTI, coverage measurement with DeTT&CT, attack tests with Atomic Red Team.

MHC-06 — Container Security and Confidential Computing

At a glance

- Operational benefit: Ensures that only unaltered, verified container images run, and protects particularly sensitive data even during processing — including against an attacker with access to the underlying infrastructure.
- Affected policy: Policy on secure development and secure operations (container and cloud operations).
- Dependencies: no precondition; presupposes a container/cloud platform; works together with MHC-02 (provenance of the images) and MHC-04 (identities).
- Effort drivers: depends on the number of container workloads and on whether confidential computing is available in the platform in use. The concrete effort depends on the organisation's resources.
- Primary metric: share of productively operated containers from signed images verified before start; additionally, coverage of confidential computing for highly sensitive workloads.
- Standards touched (without claim of fulfilment): ISO/IEC 27002 A.8.27/A.8.31; C5:2026 OPS-32 to OPS-35; NIST SP 800-190; Confidential Computing Consortium.

1. Control statement

Containers run only from signed images from controlled sources, verified before start, with enforcement via the platform. For workloads with particularly high protection needs, protected execution environments (confidential computing) with proof of authenticity (remote attestation) are used.

MHC-06 comprises two protection layers of differing maturity and differing breadth of applicability: container security as a broad baseline requirement for containerised workloads, and confidential computing as an advanced hardening for particularly protection-needing workloads, tenant separation, regulated data processing or infrastructure with elevated trust risk. Both are treated together here, but should be planned, assessed and evidenced separately in implementation.

2. Purpose and threat linkage

Containers are often assembled from publicly available base images. If such an image is manipulated, the malicious code runs along unnoticed. AI makes it easier for attackers to find manipulated images and matching vulnerabilities. Moreover, in shared cloud environments, data is encrypted in transit and at rest, but during processing in memory it is in principle visible — an attacker with access to the infrastructure could harvest it there. The protection objective is to execute only unaltered, verified containers and to protect particularly sensitive data even during processing.

3. Organisational implementation (CISO perspective)

- Policy: The operations policy prescribes signed, verified images from controlled sources, as well as confidential computing for clearly designated, particularly sensitive workloads.
- Responsibilities: Assigned in the Statement of Applicability (template MRIS Annex H); the platform/operations team and security work together.
- Risk analysis and SoA: Which workloads need confidential computing is derived from the protection needs (not everything needs it). Risk treated: manipulated images and access to data in processing (linkage to the risk assessment bridge, MRIS Annex F).

- Processes: controlled image sources and approvals; regulated handling of vulnerability findings before deployment; where procured from the cloud, evidence of the provider capabilities (confidential computing, attestation).

4. Technical implementation (for IT specialists)

- Signed images: base and own images are signed (e.g. with Sigstore/cosign) and pulled from controlled registries; the signature is verified before start.
- Enforcement at deployment: a platform admission controller (e.g. OPA Gatekeeper or Kyverno in Kubernetes) admits only signed, verified images and blocks the rest.
- Image verification: automated vulnerability scans of the images before deployment; regular re-scanning of stored images when new vulnerabilities become known.
- Confidential computing: for highly sensitive workloads, protected execution environments (TEE/secure enclaves, e.g. Intel TDX, AMD SEV-SNP, ARM CCA) that encrypt data even in memory; before use, remote attestation proves that the environment is genuine and unaltered.
- Deep dive: container risks and protective measures in NIST SP 800-190 (2017 edition, still the authoritative reference); vendor-neutral foundations at the Confidential Computing Consortium; requirements in C5:2026 OPS-32 to OPS-35.
- Effectiveness test: An attempt is made to start an unsigned or unverified image in the productive environment — the start must be prevented by the platform.

5. Implementation examples

Example A — development/platform context. A vendor operates its software in containers largely composed of public base images. Until now, nobody has checked whether these images are unaltered. The vendor therefore introduces end-to-end image signing: only signed images, previously scanned for vulnerabilities, from its own controlled registries may start, enforced via a platform admission controller. For the few services processing particularly sensitive customer data, it additionally uses a protected execution environment in which the data remains encrypted even in memory, and proves its authenticity before use. Manipulated images are thus rejected, and even an attacker with infrastructure access cannot reach the most sensitive data.

Applicability note (instead of Example B). Container security and confidential computing presuppose that the organisation builds or operates containers itself. An organisation that exclusively uses off-the-shelf SaaS services cannot implement this measure itself; for it, the measure becomes a procurement and evidence question: it follows up with the provider on whether it signs and verifies images, prevents the start of unapproved images and offers protected execution environments for particularly sensitive data — substantiated, for instance, by certificates or audit reports (e.g. C5).

6. Maturity path (cumulative)

- Initial: container images signed; vulnerability scan before deployment.
- Defined: enforcement via admission controllers in operations; use cases for confidential computing designated and documented.
- Managed: confidential computing in production for highly sensitive workloads; remote attestation automated.

7. Measurement and audit evidence

- Metric: share of productive containers from signed images verified before start (target value: complete); additionally, coverage of confidential computing for the workloads classified as highly sensitive.

- Evidence: admission controller configuration, signing and scan logs, attestation evidence of the protected environments.
- Audit logic per ISO/IEC 27005: documented? — operations policy, platform configuration; responsible? — platform/operations management; frequency? — at every deployment, scans ongoing; tolerance? — no productive container without a valid signature and verification.

8. Typical mistakes

- Images are signed, but the signature is not verified at start — the signature is then ineffective.
- The admission controller exists but runs only in warn mode; non-permitted images are reported but not blocked.
- Confidential computing is demanded across the board, even though only a few highly sensitive workloads need it — unnecessary effort.
- Stored images are never re-scanned after the first scan; newly disclosed vulnerabilities remain undetected.

9. Delineation, residual risk and references

- Delineation: concerns the integrity of the containers and the protection of data in processing; the provenance of the contained components is covered by MHC-02, the separation of multiple tenants by MHC-07.
- Residual risk: confidential computing protects the processing, but not defects in the application itself; protected execution environments are not available everywhere and can affect performance.
- Applicability: container security is a baseline requirement for containerised workloads; confidential computing is context-dependent and frequently to be justified as N/A in the Statement of Applicability. Both protection layers must be planned, assessed and evidenced separately.
- ISO/IEC 27002:2022 (indirect): A.8.27 Secure system architecture and engineering principles; A.8.31 Separation of development, test and production environments; A.5.23 Information security for use of cloud services.
- Framework: C5:2026 OPS-34/35 (Container Management), OPS-32/33 (Confidential Computing), PSS-11; NIST SP 800-190; Confidential Computing Consortium; image signing via Sigstore/cosign.

MHC-07 — Multi-Tenancy Isolation with Demonstrable Separation

At a glance

- Operational benefit: Demonstrably prevents an attacker who gains a foothold with one tenant from reaching over to the data of other tenants — the damage remains limited to one tenant.
- Affected policy: Policy on secure architecture and tenant separation (for multi-tenant services).
- Dependencies: no precondition; concerns organisations operating multiple customers on a shared platform; makes use of key separation (linkage to A.8.24) and network separation.
- Effort drivers: depends on the architecture model (shared vs. separate resources) and the number of tenants. The concrete effort depends on the organisation's resources.

- Primary metric: result of regular, ideally automated separation tests (no cross-tenant access demonstrable).
- Standards touched (without claim of fulfilment): ISO/IEC 27002 A.8.22/A.5.23; C5:2026 OPS-30/31, PSS-10; CSA Cloud Controls Matrix.

1. Control statement

In multi-tenant services, data, networks and compute resources are separated per tenant, and the separation is demonstrated through regular, ideally automated tests — not merely asserted conceptually.

2. Purpose and threat linkage

In multi-tenant services, many customers share a common platform. If an attacker succeeds in gaining a foothold with one tenant, the central question is whether it can reach over from there to the data of other tenants. AI-supported attackers systematically search for precisely such gaps in the separation. A merely conceptual separation is not enough; what matters is that the separation actually takes effect and that this is substantiated. The protection objective is to demonstrably limit the damage of an attack to a single tenant.

3. Organisational implementation (CISO perspective)

- Policy: The architecture policy prescribes, for multi-tenant services, documented separation concepts for data, networks and compute resources, as well as their regular demonstration.
- Responsibilities: Assigned in the Statement of Applicability (template MRIS Annex H); architecture and operations work together.
- Risk analysis and SoA: The required separation level (logical or physical) is derived from the protection needs of the customer data. Risk treated: cross-tenant access (linkage to the risk assessment bridge, MRIS Annex F).
- Processes: regular separation tests with protocol; treatment of identified weaknesses; evidence provision to customers and auditors.

4. Technical implementation (for IT specialists)

- Data separation: tenant-specific keys and logical or — for the highest protection needs — physical separation of the data sets.
- Network separation: separate network areas per tenant (e.g. dedicated virtual networks/ VPCs, namespaces, rule-based separation via software-defined networking).
- Separation of compute resources: tenant-related allocation (scheduling), so that workloads of different tenants do not share resources in an uncontrolled way.
- Demonstration: regular, ideally automated tests that deliberately attempt to access one tenant from another; the result (no access possible) is documented.
- Deep dive: separation requirements in C5:2026 OPS-30/31 (data separation) and PSS-10 (network); control catalogue of the CSA Cloud Controls Matrix; network separation as a principle in A.8.22.
- Effectiveness test: From a test tenant, a deliberate attempt is made to access data or resources of another tenant — the access must fail, and the result is recorded.

5. Implementation examples

Example A — development/platform context. A SaaS provider operates many customers on a shared platform. The separation has so far been described only in architecture documents but never verified. The provider introduces tenant-specific keys, separates the network areas per customer and ensures that compute loads of different customers do not share resources in an uncontrolled way. Above all, it establishes regular, automated tests that attempt to force access to

other tenants from within one tenant. If such an attempt fails, the separation is substantiated; if it succeeds, there is a clear finding to remedy. An asserted separation thus becomes a demonstrated one.

Applicability note (instead of Example B). This measure is aimed at organisations that themselves operate multi-tenant services. An organisation that only uses such services does not implement the separation itself; for it, the measure becomes a procurement and evidence question: it follows up with the provider on how it implements and demonstrates tenant separation (for instance through audit reports or certificates such as C5) and whether the separation is tested regularly.

6. Maturity path (cumulative)

- Initial: logical tenant separation documented.
- Defined: tenant-specific keys; network and resource separation implemented.
- Managed: demonstrable separation through automated tests; regular checks.

7. Measurement and audit evidence

- Metric: result of the regular separation tests (no cross-tenant access demonstrable); frequency and coverage of the tests.
- Evidence: separation concepts, test protocols, remediation evidence of identified weaknesses.
- Audit logic per ISO/IEC 27005: documented? — separation concepts and test protocols; responsible? — architecture/operations management; frequency? — regular tests; tolerance? — no demonstrated cross-tenant access.

8. Typical mistakes

- The separation is described only conceptually but never substantiated through tests.
- Data separation is implemented, but network or resource separation is not — the breach occurs via the unprotected path.
- Tests only check the normal case, not the deliberate circumvention attempt from within a tenant.
- Upon extensions (new features, new tenants), the separation is not re-verified.

9. Delineation, residual risk and references

- Delineation: concerns the separation between tenants; the protection of the processing itself is covered by MHC-06, the identity of the workloads by MHC-04.
- Residual risk: a shared platform retains a shared base layer; defects in this shared layer can affect several tenants despite the separation.
- ISO/IEC 27002:2022 (indirect): A.8.22 Segregation of networks; A.5.23 Information security for use of cloud services.
- Framework: C5:2026 OPS-30/31 (data separation), PSS-10 (Software Defined Networking); CSA Cloud Controls Matrix; additionally ISO/IEC 27017 (cloud services).

MHC-08 — Immutable Backups and Recovery Validation

At a glance

- Operational benefit: Ensures that after ransomware or data manipulation a clean, unaltered copy exists and that recovery demonstrably works.
- Affected policy: Backup and recovery policy (part of contingency and continuity management).

- Dependencies: no precondition; separate access paths presuppose orderly rights management.
- Effort drivers: depends on data volume, recovery objectives and the existing backup infrastructure. The concrete effort depends on the organisation's resources.
- Primary metric: success rate of the regular recovery tests from immutable backups.
- Standards touched (without claim of fulfilment): ISO/IEC 27002 A.8.13/A.8.14; C5:2026 OPS-06 to OPS-09; DORA Art. 12; NIS2 IR No. 4.1; NIST SP 800-34, SP 1800-11/-25.

1. Control statement

The organisation maintains at least one immutable or network-separated backup copy and regularly verifies, through documented recovery tests, that the data can be restored from it without errors. The backup infrastructure is protected via its own, separate access paths.

2. Purpose and threat linkage

Modern, AI-supported attacks encrypt or falsify data and deliberately also target the backups to prevent recovery. An attacker with far-reaching access can encrypt or delete backups reachable online along with everything else; an immutable or network-separated copy remains untouched by this. The protection objective is to be able to fall back at any time after an incident on a clean, unaltered data basis — and to be certain that the recovery actually succeeds.

3. Organisational implementation (CISO perspective)

- Policy: The backup and recovery policy prescribes the 3-2-1-1-0 principle, separate access paths for the backup infrastructure and regular, documented recovery tests.
- Responsibilities: The measure is assigned to a role in the Statement of Applicability (template MRIS Annex H); the results of the recovery tests are reported.
- Risk analysis and SoA: Recovery objectives (how fast, with what data state) are derived from the risk and impact analysis; the most business-critical data first.
- Processes: a fixed plan for regular recovery tests with protocol; separate management of the access rights to the backup infrastructure; retention and protection of the backup copies.

4. Technical implementation (for IT specialists)

- 3-2-1-1-0 principle: three copies of the data, on two different media types, at least one offsite, at least one immutable or network-separated (immutable or air-gapped), zero errors in the regular recovery test.
- Immutability: backups are stored so that they cannot be altered or deleted for a defined period (WORM storage, object locks), or are kept physically separated from the network.
- Separate access paths: The backup infrastructure uses its own accounts and rights, separate from the productive administration access, so that a taken-over production account cannot reach the backups.
- Integrity checking: regular, ideally automated verification that the backups are complete and unaltered.
- Deep dive: approach and building blocks in NIST SP 1800-11 (recovery from ransomware) and SP 1800-25 (protection of data assets); contingency planning in NIST SP 800-34 Rev. 1; requirements for backup and testing in C5:2026 OPS-06 to OPS-09.
- Effectiveness test: A complete recovery test is performed from an immutable copy; additionally, an attempt is made with a productive administration account to alter or delete a backup — this must be denied.

5. Implementation examples

Example A — development/platform context. A vendor has so far backed up its systems regularly, but all backups are reachable via the same administration access as the productive systems. An attacker who takes over such an access could also encrypt the backups. The vendor therefore stores at least one copy immutably, so that it cannot be deleted or altered for a fixed period, and separates the administration of the backup infrastructure from the productive access. Quarterly it restores the data in full as a test and documents the result. Even in the event of a far-reaching compromise, a clean data basis is thus preserved.

Example B — general department. An organisation without in-house IT development backs up its files and mailboxes via a cloud service. It ensures that at least one copy is kept immutably (many services offer a retention or lock function for this) and that the administration of this backup does not hang on the same accounts as the daily administration. Once a quarter it restores files from the backup on a sample basis and records that it worked. It thus achieves the core of the protection even without its own backup infrastructure.

6. Maturity path (cumulative)

- Initial: backups in place; recovery tests annually.
- Defined: 3-2-1-1-0 principle implemented; recovery tests quarterly.
- Managed: immutable backups with separate access paths; automated integrity checking.

7. Measurement and audit evidence

- Metric: success rate of the quarterly recovery tests from immutable backups (target value: 100 %).
- Evidence: protocols of the recovery tests, evidence of immutability (retention and lock settings), separate rights assignment of the backup infrastructure, integrity check reports.
- Audit logic per ISO/IEC 27005: documented? — policy and test protocols; responsible? — IT/backup owners; frequency? — quarterly tests, ongoing integrity checking; tolerance? — no business-critical data set without an immutable or separated copy and without a passed recovery test.

8. Typical mistakes

- Backups exist, but they are reachable via the same access as the productive systems — an attacker encrypts both.
- Backups are created, but recovery is never fully tested; when it matters, parts are missing or the process takes too long.
- "Immutable" is only configured but not verified; too short a lock period makes the copy attackable.
- The backup does not cover all required data (such as configurations or keys), so that a complete recovery fails.

9. Delineation, residual risk and references

- Delineation: concerns backup and recovery; continuous availability via redundancy is covered by A.8.14, restart planning by A.5.29/A.5.30.
- Residual risk: a network-separated copy is by nature less current; a very slow or rarely tested recovery can still lead to downtime when it matters.
- Effectiveness limit: immutability protects the copy, not the recovery. The retention lock must lie outside the attacker's control plane — dedicated authorisations, separate identity, no deletion or shortening rights for compromised admin accounts; what matters is the regularly tested recovery against a defined RTO, not the mere existence of immutable copies.

- ISO/IEC 27002:2022: A.8.13 Information backup; A.8.14 Redundancy of information processing facilities; A.5.29 Information security during disruption; A.5.30 ICT readiness for business continuity; A.5.33 Protection of records.
- Framework: C5:2026 OPS-06 to OPS-09; DORA Art. 12; NIS2 IR No. 4.1; NIST SP 800-34 Rev. 1; NIST SP 1800-11 and 1800-25 (data integrity / ransomware recovery, NCCoE); industry practice 3-2-1-1-0.

MHC-09 — AI-Supported Security Testing in the Pipeline

At a glance

- Operational benefit: Finds vulnerabilities before delivery — automatically with every change — and removes the attacker's time window between the disclosure of a flaw and its closure.
- Affected policy: Policy on secure software development.
- Dependencies: presupposes an automated build/CI pipeline; uses the bills of materials from MHC-02 for the component reconciliation.
- Effort drivers: depends on the number and structure of the pipelines and the volume of code produced. The concrete effort depends on the organisation's resources.
- Primary metric: time from the disclosure of an actively exploited vulnerability (KEV) to the rolled-out patch; additionally, the share of pipelines with security testing and pre-merge block.
- Standards touched (without claim of fulfilment): ISO/IEC 27002 A.8.28/A.8.29; NIST SP 800-218 (SSDF) and SP 800-218A (AI); OWASP ASVS; C5:2026 OPS-25; CRA Annex I.

1. Control statement

Security checks run automatically in the development pipeline: code, dependencies and the running application are checked with every change; severe findings block the merge. AI-generated code is additionally checked as a dedicated risk category.

2. Purpose and threat linkage

Previously, there was time to patch between the disclosure of a vulnerability and a working attack. AI shortens this window drastically — sometimes to hours. Anyone who finds vulnerabilities late is too slow. In addition, AI coding assistants regularly produce insecure patterns (hard-coded credentials, missing input validation, insecure defaults, outdated building blocks). The protection objective is to find and fix vulnerabilities as early as possible — already with every change — automatically, before they reach operations.

3. Organisational implementation (CISO perspective)

- Policy: The development policy prescribes automatic security checks in the pipeline, the blocking of severe findings and the separate checking of AI-generated code.
- Responsibilities: Assigned in the Statement of Applicability (template MRIS Annex H); AI-generated code is maintained as a dedicated risk category.
- Risk analysis and SoA: Thresholds for blocking findings and the handling of actively exploited vulnerabilities (KEV) are defined. Risk treated: exploitable vulnerabilities that reach operations undetected (linkage to the risk assessment bridge, MRIS Annex F).
- Processes: a regulated procedure for blocking findings and for KEV cases (accelerated remediation); regular evaluation of the check results; penetration tests with scenarios tailored to the current threat situation.

4. Technical implementation (for IT specialists)

- Check classes in the pipeline: static code analysis (SAST), dynamic testing of the running application (DAST) and checking of the dependencies (SCA) against vulnerability databases — automatically with every change.
- Blocking (pre-merge block): severe findings (high/critical) above a defined confidence prevent the merge; actively exploited vulnerabilities (KEV) block immediately.
- Running operations: regular (ideally daily) vulnerability scans of the systems already running too, not only of the new code.
- AI code as a dedicated category: additional check rules against the typical weaknesses of AI-generated code (e.g. via pre-commit checks); regular evaluation with trend tracking; AI-generated code is designated as a dedicated risk category in the Statement of Applicability.
- Deep dive: secure development practices in NIST SP 800-218 (SSDF), supplemented for AI model development by SP 800-218A; application security verification criteria in OWASP ASVS; scan requirements in C5:2026 OPS-25.
- Effectiveness test: In a test branch, a known insecure element is deliberately introduced (e.g. a hard-coded credential); the pipeline must report the finding and block the merge.
- Effectiveness limit: AI-supported testing does not replace expert assessment. In particular, it misses logic, authorisation and architecture weaknesses — precisely the findings with high impact — and produces false positives as well as false negatives. "Tool ran" is not "vulnerability blocked": critical findings, blocking pipeline decisions and accepted exceptions require expert triage, a documented decision and a traceable approval.

5. Implementation examples

Example A — development/platform context. A software vendor has so far checked security only late, shortly before a release. With a newly disclosed vulnerability it is therefore regularly too slow. It builds security checks directly into its pipeline: code, dependencies and the running application are checked automatically with every change, and severe findings prevent the merge. For actively exploited vulnerabilities there is an accelerated path to close them immediately. Since its developers use AI coding assistants, it adds check rules against the typical errors of such tools and evaluates these findings separately. Flaws are thus found before they are shipped.

Example B — general department. In a business department, employees build small applications and automations of their own with low-code tools and AI coding assistants. They are often unaware that AI-generated code can contain insecure patterns. The organisation therefore stipulates that such self-built solutions do not go into production unchecked: everything beyond simple aids passes through a functional and security review, and central IT must be involved for sensitive data or interfaces. The benefit of fast in-house development is thus preserved without unchecked AI code reaching operations uncontrolled.

6. Maturity path (cumulative)

- Initial: SAST/DAST/SCA in the pipeline.
- Defined: additional checking of AI-generated code; blocking on severe findings.
- Managed: daily vulnerability scans of the running systems; separate evaluation of the AI code; accelerated path for actively exploited vulnerabilities (KEV).

7. Measurement and audit evidence

- Metric: time from the disclosure of an actively exploited vulnerability (KEV) to the rolled-out patch (guideline: ideally under 24 hours); additionally, the share of pipelines with security testing and pre-merge block.

- Evidence: pipeline configuration, check and blocking logs, evaluation of the AI code findings, penetration test reports.
- Audit logic per ISO/IEC 27005: documented? — development policy, pipeline configuration; responsible? — development/product owners; frequency? — with every change, scans daily; tolerance? — no delivery with open severe findings.

8. Typical mistakes

- The checks run but do not block; severe findings are reported and shipped anyway.
- Only the new code is checked, not the systems already running; vulnerabilities disclosed there remain open.
- AI-generated code is treated like hand-written code; its typical weaknesses are not deliberately sought.
- The thresholds are set so strictly that the pipeline blocks constantly; as a result, checks are circumvented instead of findings fixed.

9. Delineation, residual risk and references

- Delineation: concerns automated testing during development; the transparency of the components is covered by MHC-02, the detection of active attacks by MHC-05.
- Residual risk: automated checks find known patterns, not every novel or logical vulnerability; penetration tests and manual reviews remain necessary as supplements.
- ISO/IEC 27002:2022: A.8.8 Management of technical vulnerabilities; A.8.25 Secure development life cycle; A.8.26 Application security requirements; A.8.28 Secure coding; A.8.29 Security testing in development and acceptance; A.8.32 Change management.
- Framework: C5:2026 OPS-25 (with sharpening OPS-25.01AS, daily scans); NIST SP 800-218 (SSDF v1.1; Revision 1.2 in draft) and SP 800-218A (secure development for generative AI); OWASP ASVS; CRA Annex I Part II.

MHC-10 — Continuous Control Monitoring and Policy as Code

At a glance

- Operational benefit: Shortens the time to detect a security deviation from months (audit cadence) to minutes, because requirements are checked automatically on an ongoing basis.
- Affected policy: Policy on monitoring the effectiveness of and compliance with security requirements.
- Dependencies: no precondition; presupposes requirements that can be formulated machine-readably and access to system/configuration data.
- Effort drivers: depends on the number and type of requirements to be checked and the heterogeneity of the system landscape. The concrete effort depends on the organisation's resources.
- Primary metric: share of the controls maintained in the SoA whose compliance is checked automatically (at least daily) (continuous monitoring coverage).
- Standards touched (without claim of fulfilment): ISO/IEC 27002 A.5.35/A.5.36; C5:2026 COM-03/04; NIS2 IR No. 2.2; DORA Art. 6(5); NIST SP 800-137.

1. Control statement

Compliance with security requirements is checked continuously and automatically against defined target states, rather than only in periodic audits. Requirements are, where possible, formulated as executable rules (policy as code) and integrated into the provisioning and operations processes; deviations automatically trigger a response.

2. Purpose and threat linkage

Periodic audits check only at fixed points in time (e.g. quarterly); in between, deviations remain undetected. AI-supported attackers exploit precisely this gap: they create — or find — a misconfiguration and use it long before the next audit takes place. Moreover, agentic attackers decompose their approach into steps that each look compliant on their own. A continuous, automated check detects deviations immediately and brings them straight to a response. The protection objective is to reduce the time span between a deviation and its detection to a minimum.

3. Organisational implementation (CISO perspective)

- Policy: The monitoring policy stipulates that compliance with essential requirements is checked continuously and automatically and that deviations are handled immediately — as a supplement to, not a replacement for, the audits required anyway.
- Responsibilities: Maintenance of the rules and evaluation are assigned in the Statement of Applicability (template MRIS Annex H); functionally, the ISMS management is usually responsible.
- Risk analysis and SoA: It is defined which controls are monitored automatically as a priority — derived from the risk assessment and protection needs. Risk treated: deviations from the target state that remain undetected for a long time (linkage to the risk assessment bridge, MRIS Annex F).
- Processes: a defined procedure for how a detected deviation automatically leads to a case (ticket) and to remediation; regular maintenance and extension of the rule set; reporting on coverage and open deviations.

4. Technical implementation (for IT specialists)

- Policy as code: security requirements are formulated as executable rules (e.g. with Open Policy Agent/Rego or HashiCorp Sentinel) and managed in versioned form like source code.
- Integration at two points: before provisioning in the CI pipeline (check before something goes into operation) and in running operations (continuous checking of the actual state against the target requirements), ideally with enforcement (runtime enforcement).
- Target states (baselines): the checked target requirements are defined and maintained; machine-readable control catalogues (e.g. in NIST's OSCAL format) facilitate automated checking and evidence.
- Response: deviations are displayed on real-time overviews with clear metrics and automatically trigger a case (incident ticket).
- Deep dive: building a continuous monitoring programme in NIST SP 800-137 (ISCM) and its assessment in SP 800-137A; machine-readable controls via NIST OSCAL.
- Effectiveness test: In a test area, a deviation from the target state is deliberately created (e.g. a non-permitted configuration); the automated check must report the deviation within minutes and trigger a case.

5. Implementation examples

Example A — development/platform context. A company has so far checked compliance with its security requirements on a quarterly basis via audit. Between the checks, a faulty approval in a cloud environment only comes to light weeks later. The company therefore formulates its most important requirements as executable rules and integrates them into both provisioning and running operations. If a configuration violates a rule, this appears immediately on an overview and automatically creates a case for remediation. The quarterly snapshot thus becomes a continuous control.

Example B — general department. An organisation without in-house development uses mainly cloud services. It cannot program rules of its own, but activates the functions available in its platforms for continuous configuration checking (such as a security or compliance status that runs permanently). Deviations from the recommended settings are continuously displayed there; the organisation defines who receives and remedies them. It thus obtains a continuous check via the on-board means of its services, without developing itself.

6. Maturity path (cumulative)

- Initial: compliance overviews with manual maintenance.
- Defined: policy as code in the CI pipeline; real-time checks against target states.
- Managed: enforcement in operations (runtime enforcement); automatically created cases on deviations.

7. Measurement and audit evidence

- Metric: continuous monitoring coverage — share of the controls maintained in the SoA with active automated checking (at least daily, with alerting on deviation). Guidelines: ≥ 60 % after 12 months, ≥ 80 % after 24 months.
- Evidence: rule repository (policy as code), overview of the automatically checked controls, logs of cases triggered on deviations.
- Audit logic per ISO/IEC 27005: documented? — rule set and coverage overview; responsible? — ISMS management; frequency? — ongoing (at least daily); tolerance? — essential controls without automated checking only temporarily and with justification.

8. Typical mistakes

- The automated check is understood as a replacement for audits; in fact it supplements them — the required reviews remain necessary.
- Overviews are built, but deviations trigger no response; the finding has no consequence.
- Only easily checkable requirements are automated, but not the essential ones; the coverage appears high but does not cover what matters.
- The target states are defined once and not maintained, so that the check measures against outdated requirements.

9. Delineation, residual risk and references

- Delineation: concerns the continuous checking of compliance with requirements; the detection of active attacks is covered by MHC-05, the automated response by MHC-11.
- Residual risk: only what is formulated as a rule is checked — unclear or non-machine-checkable requirements are left out; policy-compliant but malicious activity only becomes visible in combination with behaviour-based detection (MHC-05).
- ISO/IEC 27002:2022: A.5.18 Access rights; A.5.35 Independent review of information security; A.5.36 Compliance with policies, rules and standards for information security; A.8.9 Configuration management.
- Framework: C5:2026 COM-03/COM-04; NIS2 IR No. 2.2; DORA Art. 6(5); OWASP SAMM; NIST SP 800-137 and 800-137A (ISCM); NIST OSCAL (machine-readable controls); policy engines Open Policy Agent (OPA/Rego), HashiCorp Sentinel.

MHC-11 — SOAR-Based Tier-1 Automation and Parallel Response Playbooks

At a glance

- Operational benefit: Brings the response to unambiguous incidents from hours to minutes by having the fast, clear containment steps run automatically — humans concentrate on the ambiguous and severe cases.
- Affected policy: Policy on handling security incidents (incident response).
- Dependencies: presupposes reliable detection (MHC-05 supplies the signals); the automation layer itself must be hardened.
- Effort drivers: depends on whether a SOAR platform is operated in-house or procured as a service, and on the number of playbooks. The concrete effort depends on the organisation's resources.
- Primary metric: time to containment of an unambiguous incident (Mean Time to Containment, MTTC).
- Standards touched (without claim of fulfilment): ISO/IEC 27002 A.5.24/A.5.26; C5:2026 SIM-02/03; NIS2 IR No. 3.5; DORA Art. 17; NIST SP 800-61 Rev. 3.

1. Control statement

The organisation responds to unambiguous incidents automatically with predefined procedures (playbooks) that initiate containment in minutes; ambiguous or severe cases go deliberately to humans. The automation layer itself is secured. The organisation is prepared for several simultaneous incidents.

2. Purpose and threat linkage

AI-supported attacks run at machine speed: often only minutes pass from initial access to damage. A purely human response chain — review the alert, assess, escalate, act — is too slow for this. At the same time, attackers increasingly run several attacks in parallel to overload the response. The protection objective is to automate the fast, unambiguous containment steps so that they take effect in minutes, and to be prepared for several simultaneous incidents. Important: the automation itself becomes an attack target and must be protected accordingly.

3. Organisational implementation (CISO perspective)

- Policy: The incident response policy defines which incidents are contained automatically, when escalation to humans occurs and how the automation layer is protected.
- Responsibilities: Assigned in the Statement of Applicability (template MRIS Annex H); security operations are oriented towards analysis and incident command, not pure alert review.
- Risk analysis and SoA: Which actions may run automatically is defined by unambiguousness and possible damage. Risk treated: too slow a response against machine-speed and parallel attacks (linkage to the risk assessment bridge, MRIS Annex F).
- Processes: regular exercises for three to five simultaneous incidents (at least every six months); maintenance of the playbooks; clear escalation paths; where procured as a service, a contractually assured response time.

4. Technical implementation (for IT specialists)

- Automated containment: a SOAR platform executes predefined playbooks on unambiguous signals (e.g. locking an account, isolating a system); ambiguous cases are passed on for human review.

- Prepared scenarios: playbooks for typical Mythos situations, such as mass data leakage, parallel multi-attack, supply chain compromise, attack waves on logins.
- Hardening of the automation layer (central): The SOAR pipeline is itself an attack surface. Protective measures: only authenticated, signed triggers from trusted sources; complete evidence of every playbook execution including the executing identity; limitation of the trigger rate per playbook (against false-alarm floods); human approval (human in the loop) for actions with a large blast radius (e.g. mass account lock, isolation of larger network areas, certificate revocation); protection of the playbooks like source code (versioning, code review, signed releases).
- Deep dive: structure and lifecycle of incident handling in NIST SP 800-61 Revision 3 (aligned with CSF 2.0); requirements in C5:2026 SIM-02/03 and NIS2 IR No. 3.5.
- Effectiveness test: An unambiguous attack signal is triggered in a test environment; the matching playbook must initiate containment automatically and within the target time — and an action with a large blast radius may only take place after human approval.

5. Implementation examples

Example A — development/platform context. A company with its own security operations discovers that the time from alert to first countermeasure is too long, because every step runs manually. It therefore automates the unambiguous containment steps: on clear signals, an affected account is locked immediately or a system is automatically isolated, while ambiguous cases go to the analysts. So that the automation does not itself become a weak point, the platform accepts only authenticated triggers, logs every action completely and requires human approval for far-reaching interventions. Twice a year the team practises several simultaneous incidents. The time to containment falls from hours to minutes.

Example B — general department. An organisation without its own 24/7 operations cannot build and maintain such automation itself. It therefore procures detection and fast response as a service (managed detection & response) and, when selecting, looks for a contractually assured response time and traceable, agreed containment measures. Internally it defines who is reachable when it matters and which far-reaching actions are approved in advance. It thus obtains a fast, practised response without operating its own platform.

6. Maturity path (cumulative)

- Initial: response playbooks documented; manual execution.
- Defined: SOAR with partially automated playbooks; time to containment under 60 minutes.
- Managed: fully automatic containment of unambiguous incidents, time under 10 minutes; six-monthly exercises with several simultaneous incidents — alternatively an MDR service with a contractually assured response time.

7. Measurement and audit evidence

- Metric: time to containment of an unambiguous incident (MTTC; guideline: under 10 minutes for high unambiguousness).
- Evidence: playbook definitions and logs, evidence of the hardening of the automation layer (trigger authentication, audit trail, approval rules), protocols of the multi-incident exercises or the service provider SLA.
- Audit logic per ISO/IEC 27005: documented? — incident response policy, playbooks; responsible? — security operations/SOC management; frequency? — ongoing, exercises at least every six months; tolerance? — no automatic action with a large blast radius without human approval.

8. Typical mistakes

- Broad automation is introduced without protecting the automation layer; unauthenticated triggers or false-alarm floods trigger unwanted containments.
- Far-reaching actions run without human approval; a false alarm then locks many accounts or isolates whole network areas.
- Playbooks exist, but there is no exercise for several simultaneous incidents; when it matters, a parallel attack overloads the response.
- The playbooks are not maintained like source code (no versioning, no review); errors and manipulations go unnoticed.

9. Delineation, residual risk and references

- Delineation: concerns the response to detected incidents; the detection itself is covered by MHC-05, the continuous compliance checking by MHC-10.
- Residual risk: the automation can become a target — attackers can reconnoitre trigger conditions, deliberately provoke false alarms or turn containment actions against one's own organisation; without the hardening described, the measure is not effective.
- Effectiveness limit: SOAR does not replace resilient detection, asset context and incident governance. Automation only takes effect once triggers, data quality, criticality logic, approval rules and fallback paths are defined and tested; actions with potentially high blast radius provide by default for human approval, simulation or staged execution.
- ISO/IEC 27002:2022: A.5.5 Contact with authorities; A.5.24 Information security incident management planning and preparation; A.5.25 Assessment and decision on information security events; A.5.26 Response to information security incidents.
- Framework: C5:2026 SIM-02/03; NIS2 IR No. 3.5; DORA Art. 17; NIST SP 800-61 Revision 3 (incident handling, aligned with CSF 2.0).

MHC-12 — Threat-Led Penetration Testing with Mythos Scenarios

At a glance

- Operational benefit: Tests whether the protective measures actually work against realistic, AI-typical attack scenarios — not just whether they are documented — and thereby uncovers false-positive security assumptions.
- Affected policy: Policy on security testing and independent review.
- Dependencies: no precondition; tests the effectiveness of the other controls (inter alia MHC-05, MHC-11) under realistic conditions.
- Effort drivers: depends strongly on the chosen format (full TLPT with an external red team vs. more cost-effective purple team and simulation formats). The concrete effort depends on the organisation's resources.
- Primary metric: share of the identified vulnerabilities remediated within the agreed time window (closure rate).
- Standards touched (without claim of fulfilment): ISO/IEC 27002 A.5.35/A.8.29; DORA Art. 26/27 with TLPT RTS; TIBER-EU; C5:2026 OPS-22; NIS2 IR No. 3.5.5.

1. Control statement

The effectiveness of the protective measures is regularly checked through threat-led tests (threat-led penetration testing) that replicate realistic attack scenarios tailored to the current situation — including AI-typical approaches. Between the cycles, joint exercises of attack and defence (purple team) take place.

2. Purpose and threat linkage

Documented protective measures are not the same as effective protective measures. Under AI-supported attacks — which today demonstrably find and exploit vulnerabilities autonomously — only a realistic test shows whether the defence actually holds. Classic penetration tests with a fixed standard scope often do not reflect the new approaches. The protection objective is to check, with realistic, threat-led scenarios, whether the protective measures work against real attack patterns, and thereby uncover false-positive security assumptions.

3. Organisational implementation (CISO perspective)

- Policy: The testing policy prescribes regular threat-led tests with realistic scenarios, as well as purple team exercises between the cycles.
- Responsibilities: Assigned in the Statement of Applicability (template MRIS Annex H); for the highest maturity, the timely remediation of the findings is reported to management.
- Risk analysis and SoA: Scope and scenarios are derived from the threat situation and the critical functions; the appropriate format is chosen by protection needs and resources. Risk treated: protective measures that only work on paper (linkage to the risk assessment bridge, MRIS Annex F).
- Processes: regulated commissioning and execution (for full TLPT with separate providers for threat analysis and red team); tracking and timely closure of the findings; regular repetition (at least annually for critical functions).

4. Technical implementation (for IT specialists)

- Threat-led tests: external red teamers replicate real attack chains, derived from current threat analysis — tailored to the organisation, against the productive systems.
- Include Mythos scenarios: e.g. AI-supported spear phishing with a faked voice (deepfake voice), attack chains decomposed into micro-steps, parallel multi-attacks, supply chain compromises, misuse of cloud access via taken-over service accounts.
- Purple team between the cycles: attack and defence sides work together and map what has been practised to the techniques per MITRE ATT&CK, in order to improve the detection coverage (MHC-05) in a targeted way.
- Graduated formats: full TLPT per TIBER-EU methodology (elaborate, for regulated/critical entities) or more cost-effective formats — purple team exercises with ATT&CK scenarios, open-source attack simulation (e.g. Caldera, Atomic Red Team) and breach-and-attack simulation platforms (e.g. AttackIQ, SafeBreach, Picus).
- Deep dive: TLPT approach in the DORA TLPT RTS (Delegated Regulation (EU) 2025/1190) and in the ECB's TIBER-EU framework (aligned with DORA since February 2025, with mandatory purple teaming); penetration test requirements in C5:2026 OPS-22.
- Effectiveness test: In a threat-led scenario (e.g. an attack chain decomposed into micro-steps), it is checked whether the existing detection and response actually notices and stops the attack — an unnoticed run is the finding.

5. Implementation examples

Example A — development/platform context. A company has a standard penetration test carried out annually, which runs similarly each time. New, AI-typical approaches are not tested in it. The company therefore moves to threat-led tests: external red teamers replicate real attack chains — such as an approach decomposed into many small steps or phishing with a faked voice — and check against the real systems whether detection and response hold. Between the tests, attack and defence sides practise together and close identified gaps in detection. It thus becomes visible which protective measures really work and which are only documented.

Example B — general department. A smaller organisation cannot shoulder a full, elaborate test programme. It therefore chooses a more cost-effective but effective format: regular joint exercises of attack and defence based on realistic scenarios, supported by open-source attack simulation or a simulation platform that automatically replicates typical attack techniques. The results show concretely where detection needs improvement. It too thus checks the effectiveness of its protective measures, adapted to its resources.

6. Maturity path (cumulative)

- Initial: annual penetration tests with a standard scope.
- Defined: threat-led tests with Mythos scenarios; purple team exercises between the cycles.
- Managed: continuous attack simulation, simulation platform in production; timely remediation of the findings (closure rate ≥ 90 % within the agreed time window).

7. Measurement and audit evidence

- Metric: closure rate — share of the identified vulnerabilities remediated within the agreed time window (guideline: ≥ 90 %); additionally, the frequency and realism of the tests.
- Evidence: test reports with scenarios and findings, remediation evidence, ATT&CK mapping of the purple team exercises.
- Audit logic per ISO/IEC 27005: documented? — testing policy, test reports; responsible? — security management; frequency? — at least annually for critical functions, purple team in between; tolerance? — no open critical findings beyond the agreed time window.

8. Typical mistakes

- Testing is done, but always with the same standard scope; realistic, new attack patterns are left out.
- Findings are reported but not remediated in time; the test only confirms what is known without improving the situation.
- The test only checks the technology, not the detection and response; whether an attack would be noticed remains open.
- The format is oversized (expensive TLPT where a purple team format suffices) or undersized (superficial scan where realistic scenarios would be needed).

9. Delineation, residual risk and references

- Delineation: concerns the checking of effectiveness; automated testing during development is covered by MHC-09, the continuous detection by MHC-05.
- Residual risk: a test is a snapshot of a chosen scenario; untested paths remain open, and the situation changes continuously — hence the regular repetition and the purple team exercises in between.
- Format delineation: MHC-12 requires the checking of realistic attack chains, not necessarily a DORA TLPT in the narrow regulatory sense. For DORA-obligated entities, the regulated DORA TLPT (RTS (EU) 2025/1190) remains binding; all others also achieve the objective via threat-led penetration tests, purple team exercises, attack simulations or scenario-based control tests.
- ISO/IEC 27002:2022: A.5.35 Independent review of information security; A.8.29 Security testing in development and acceptance.
- Framework: DORA Art. 26/27 with the TLPT RTS (Delegated Regulation (EU) 2025/1190, applicable since July 2025); the ECB's TIBER-EU framework (aligned with DORA since February 2025); C5:2026 OPS-22; NIS2 IR No. 3.5.5; attack simulation inter alia with Caldera, Atomic Red Team, AttackIQ, SafeBreach, Picus.

MHC-13 — AI Agent Governance and Harness Security

At a glance

- Operational benefit: Brings self-operated AI agents under control — limited access, complete traceability, human approval before risky actions — and thereby closes an otherwise uncontrolled attack surface.
- Affected policy: new policy on the use of AI agents; additionally the access control and procurement policy.
- Dependencies: presupposes the identity foundation from MHC-04; touches MHC-09 (insecure AI code) and the shadow AI inventory (A.5.9).
- Effort drivers: depends on the number and reach of the agents and integrations in use; later expansion stages presuppose suitable tools available in one's own environment. The concrete effort depends on the organisation's resources.
- Primary metric: share of productive agents with a documented threat model and limited access, as well as the coverage of the logging.
- Standards touched (without claim of fulfilment): inter alia ISO/IEC 42001 Annex A, NIST AI RMF 1.0, OWASP threat taxonomies; depending on the area of use, also the EU AI Act.

1. Control statement

Productively deployed AI agents (coding agents, agentic workflows) are treated like privileged systems: a limited blast radius per agent, an identity limited to the necessary rights, complete and traceable logging with a named human owner and kill switch, an allowlist for agentic components (MCP servers, IDE extensions, skills), and code review for the harness (the agent's control scaffolding).

Minimum requirements per productive agent: a technical, capability-scoped identity instead of personal user accounts or blanket API keys; explicitly no unrestricted tool access; tamper-proof, verifiable execution logging — the log is security-relevant evidence and is connected to the central, tamper-proof logging from A.8.15 (MHC-05); documented purpose limitation and approved tool scope per function; defined approval thresholds for actions with a high blast radius. The identity foundation is supplied by MHC-04 (workload identity), on which the capability-scoped limitation builds.

2. Purpose and threat linkage

AI agents combine the reasoning of a language model with tool execution, persistent memory and multi-stage planning. This creates a privileged attack surface outside the established security measures (Mythos-Ready, CSA/SANS/OWASP, April 2026: "Unmanaged AI Agent Attack Surface", classification CRITICAL). Two risk sides: first, over-privileged or insecure agents in one's own operations; second, the supply chain — taken-over MCP servers, IDE extensions or skills. Substantiated by real incidents: EchoLeak (CVE-2025-32711, undetected data leakage via prompt injection in Microsoft 365 Copilot) and CurXecute (CVE-2025-54135/-54136, code execution via the MCP integration of the coding IDE Cursor).

3. Organisational implementation (CISO perspective)

- Policy: A policy on the use of AI agents is created. It regulates the permitted purposes of use, the obligation of a named human owner per productive agent, the exclusive use of approved components, and the obligation of human approval before irreversible actions.
- Responsibilities: An AI governance function is named; the CISO is accountable; the board is involved, because approving productive agents touches risk acceptance decisions above the CISO mandate (MRIS Annex H, Ch. 10.5).

- Risk analysis and SoA: Before going into production, a threat and risk analysis is created and documented per purpose of use; the target maturity level is derived from it (linkage to the risk assessment bridge, MRIS Annex F).
- Processes: (a) an approval process before going into production (documented purpose of use, risk analysis, fallback plan); (b) an inventory of the AI agents and their components; (c) a procedure for complete traceability with a named owner and retention of at least 12 months; (d) a procedure for the fast revocation of taken-over agents.
- Procurement and suppliers: requirements for external agentic components (provenance, update path, security evidence) are defined; inclusion only via the allowlist.
- Training: development and business teams are sensitised to the risks, in particular to indirect prompt injection.
- Harness as code: It is stipulated that the control components of an agent (instructions, tool definitions, rule files) are subject to the same approval and versioning rules as source code; the technical implementation is in Chapter 4.

4. Technical implementation (for IT specialists)

- Limited identity: Each agent runs under its own workload identity (linkage to MHC-04) with fine-grained, time-limited rights per tool call (read, write, network rights with explicit scope); no personal developer tokens.
- Limitation of the blast radius: rate limits per agent identity; a circuit breaker that aborts on unusual action sequences; enforced approval thresholds (human in the loop) for irreversible actions (production deployment, data deletion, spending above a threshold).
- Logging: every tool call as well as file and network access is logged as a repeatably traceable record, bound to session, human owner and data source; audit-proof storage.
- Allowlist: technical blocking of unapproved MCP servers and extensions; controlled rather than automatic updates; signature verification where available.
- Hardening against prompt injection: separation of instruction and data channel as far as technically possible; input and output filtering; treatment of tool outputs as untrusted input (protection against misused tool authorisation, "confused deputy"); pen tests against prompt injection before going into production.
- Harness as code: instructions, tool definitions, retrieval pipelines and escalation logic in the version control system, with mandatory code review, signed releases and automated checks before going into production.
- Deep dive: MCP-specific controls (provenance logging, sandboxing, context isolation) in the OWASP guide "Practical Guide for Securely Using Third-Party MCP Servers"; threat taxonomy in OWASP "Agentic AI — Threats and Mitigations" and in the OWASP Top 10 for Agentic Applications (ASI01–ASI10); AI management controls in ISO/IEC 42001:2023, Annex A; adversarial robustness of the agent models in NIST AI 100-2e2025.
- Effectiveness test: A prepared email or a prepared ticket with an embedded command (injection probe) is fed in; the agent must not execute the impermissible action demanded in it (such as external sending), and the attempt must appear in the log.

5. Implementation examples

Example A — development/platform context. In an organisation with its own software development, the developers use an AI assistant directly in their working environment. This assistant is connected to several internal systems — the ticket system, the source code management and an internal knowledge base — and operates under the developers' personal access. The risk: a covert command can be smuggled into a ticket that leads the assistant to release data externally or alter source code without the developer noticing. The organisation first maintains a complete register of all assistants in use and their integrations, admits only verified integrations, gives each assistant

its own, narrowly limited access instead of full developer rights, and records completely what it does and who started it. In a further expansion stage, the assistant's control specifications are reviewed and versioned like program code, regularly tested for manipulability, and a misused access can be revoked within a few minutes.

Example B — general department (no-code/SaaS). A business department without its own development — such as sales — uses Microsoft 365 Copilot and sets up its own assistant in it with onboard means (or uses a self-created GPT in ChatGPT Business). This assistant may read the mailbox, file storage and customer database and send messages, and does so with the full rights of the employee who set it up. The risk: an incoming email or a document can contain a covert command that leads the assistant to send confidential content externally — Precisely this attack pattern was demonstrated in the publicly documented EchoLeak case for Microsoft 365 Copilot. Here the levers lie not in programming but in the administration settings of the platform: the department records which assistants exist and what they may access, admits only approved integrations, limits the assistants' rights to what is necessary instead of the full user rights, names a responsible person for each productive assistant and switches on the platform's logging. For irreversible steps — such as sending externally or deleting data — explicit human confirmation is required.

6. Maturity path (cumulative)

- Initial: AI coding agents documented; manual inventory of the MCP servers and extensions.
- Defined: limited identities and complete logging in production; threat model per purpose of use; allowlist for extensions and MCP servers.
- Managed: complete code review process for the harness; automated checks before going into production; time to revocation under 5 minutes; regular pen tests against prompt injection.
- Realism: most organisations start at Initial; realistically 12–18 months to Defined, 18–24 months to Managed. Prioritise first: inventory, logging and limited identities — these three address the largest part of the risk. Pen tests of the harness and checks of the robustness of the models follow in later expansion stages, depending on the tools available in one's own environment.

7. Measurement and audit evidence

- Metrics: share of production-adjacent coding agents with a documented threat model and a defined blast radius (target value: 100 %); coverage of the logging for actions with write or network rights (target value: 100 %); time to revocation of taken-over agent identities (target value: under 5 minutes).
- Evidence: agent inventory, allowlist configuration, logs, pre-production check reports, threat model documents.
- Audit logic per ISO/IEC 27005: documented? — via inventory and threat models; responsible? — a named human per agent; frequency? — ongoing logging, check before every go-live; tolerance? — no productive agents with write or network rights without logging.

8. Typical mistakes

- Agents run under personal developer accounts — no limited identity, no clean traceability.
- MCP servers and extensions without an allowlist and without update control — automatic updates open a path via the supply chain.
- The harness (instructions, rule files) is treated as configuration instead of code — no approval, no version control.

- Human approval in name only (a confirmation dialog without a real ability to check) — the too-broad freedom of action remains.
- The "Managed" target is set across the board, even though the tools and procedures required for it are not yet established in one's own environment.

9. Delineation, residual risk and references

- Delineation: covers governance and security of the self-operated agents; the identity foundation is supplied by MHC-04; insecure, AI-generated code is treated in MHC-09; the inventory of unapproved AI tools (shadow AI) is in A.5.9 (MRIS extension).
- Residual risk: prompt injection is not fully solvable as long as the instruction and data channels cannot be separated; detection is not prevention; suitable hardening tools are not available in every environment.
- ISO/IEC 27002:2022: A.5.16 Identity management; A.8.27 Secure system architecture and engineering principles.
- AI-specific: ISO/IEC 42001:2023, Annex A (AI management controls via the Statement of Applicability); NIST AI RMF 1.0; NIST AI 100-2e2025 (adversarial machine learning).
- Threat models: OWASP Top 10 for LLM Applications 2025 (LLM01 Prompt Injection; LLM06 Excessive Agency — with the root causes of excessive functionality, excessive rights, excessive autonomy; LLM03 Supply Chain); OWASP Top 10 for Agentic Applications (ASI01 Agent Goal Hijack, ASI02 Tool Misuse, ASI03 Identity & Privilege Abuse); OWASP "Agentic AI — Threats and Mitigations"; OWASP "Practical Guide for Securely Using Third-Party MCP Servers"; MITRE ATLAS (AML.T0051 LLM Prompt Injection, AML.T0053 LLM Plugin Compromise, AML.T0086 Exfiltration via AI Agent Tool Invocation, AML.T0110 AI Agent Tool Poisoning).

Glossary

- **3-2-1-1-0:** backup rule of thumb (three copies, two media types, one offsite, one immutable/offline, zero errors in the test).
- **AAL2 / AAL3:** protection levels for authentication per NIST SP 800-63B; AAL3 requires a device-bound, non-exportable key.
- **Admission controller:** platform control point (e.g. in Kubernetes) that admits only permitted, verified images at deployment.
- **Adversary emulation / BAS:** replication of real attack techniques, automated via breach-and-attack simulation platforms.
- **Air-gapped:** backup copy physically separated from the network.
- **Baseline (target state):** defined target state against which the actual state is checked.
- **Blast radius:** area of damage — how far a successful attack can spread.
- **Capability scoping:** limitation of an access to exactly the rights and capabilities needed for the respective task.
- **Conditional access:** access rules that check user, device and situation on every login before access is granted.
- **Confidential computing / TEE:** protected execution environment that encrypts data even during processing in memory (e.g. Intel TDX, AMD SEV-SNP, ARM CCA).
- **Container / image / registry:** a container is a standardised, packaged application; the image is its immutable template; a registry is the storage location for images.
- **Continuous control monitoring:** ongoing, automated checking of compliance with requirements instead of periodic audits.
- **CVE/NVD, OSV, EUVD:** vulnerability databases (NVD: US database; OSV: open-source vulnerabilities; EUVD: EU database of ENISA).
- **CycloneDX / SPDX:** the two established SBOM formats (CycloneDX as ECMA-424, SPDX standardised as ISO/IEC 5962).
- **Deepfake voice:** artificially generated, deceptively genuine voice, e.g. for spear phishing.
- **Containment:** immediate measure that limits an ongoing incident (e.g. lock an account, isolate a system).
- **FIDO2 / WebAuthn:** open standard for phishing-resistant login with cryptographic binding to the service's address.
- **Harness:** the control scaffolding of an AI agent — instructions, tool definitions, rule files and escalation logic.
- **"harvest now, decrypt later":** attack pattern in which data encrypted today is intercepted and retained for later decryption.
- **Human in the loop:** mandatory human approval before actions with a large blast radius.
- **Hybrid cryptography:** combination of a classical and a quantum-safe method during the transition period.
- **Image signature (Sigstore/cosign):** cryptographic signature of an image, verified before start to rule out manipulation.
- **KEV (Known Exploited Vulnerabilities):** catalogue (CISA) of actively exploited vulnerabilities; triggers accelerated patching.
- **AI-generated code (vibe-coded):** code produced with AI coding assistants, to be checked as a dedicated risk category.
- **Kill chain correlation:** linkage of individual events into a connected attack chain.
- **Living off the land:** approach in which attackers use on-board system tools instead of imported malware.
- **Tenant / multi-tenancy:** a customer on a shared platform; multi-tenancy denotes the operation of several customers on shared infrastructure.

- **MCP (Model Context Protocol):** open interface via which an AI assistant is connected to external tools and data sources.
- **MDR (Managed Detection & Response):** procurement of detection and fast response as a service with a contractual response time.
- **MITRE ATT&CK:** recognised catalogue of real attack techniques towards which detection is oriented.
- **ML-KEM / FIPS 203:** quantum-safe method for key exchange (replaces e.g. RSA/ECDH).
- **mTLS (mutual TLS):** encrypted connection in which both sides mutually identify themselves by certificate — not only one side.
- **MTTC (Mean Time to Containment):** average time to containment of an incident.
- **OSCAL:** machine-readable NIST format for control catalogues and their evidence.
- **Passkey:** login credential based on FIDO2; device-bound or synchronisable across several devices.
- **Playbook:** predefined response procedure for a particular incident type.
- **Policy as code:** security requirements as executable, versioned rules (e.g. OPA/Rego, Sentinel).
- **PQC (post-quantum cryptography):** encryption and signature methods intended to withstand future quantum computers too.
- **Pre-merge block:** automatic block that prevents code with severe findings from being merged into the main branch.
- **Prompt injection (indirect):** a command hidden in content (email, document, ticket) that induces an AI assistant to unwanted behaviour.
- **Red / blue / purple team:** attack side, defence side and their joint exercise.
- **Remote attestation:** proof that a protected execution environment is genuine and unaltered before it is used.
- **Runtime enforcement:** enforcement of requirements in running operations, not merely their checking.
- **SAST / DAST / SCA:** static code analysis, dynamic testing of the running application, checking of the dependencies.
- **SBOM (software bill of materials):** machine-readable list of all components of a piece of software including their dependencies.
- **Service mesh:** an infrastructure layer that automatically handles connection and protection between services without changing the application code.
- **SLSA / build provenance:** framework and evidence of the sources and process from which an artefact was built.
- **SOAR:** platform that automatically executes defined response procedures (playbooks) (Security Orchestration, Automation and Response).
- **SPIFFE / SPIRE:** open standard (SPIFFE) and associated reference implementation (SPIRE) to give each software component a unique, technical identity.
- **SSDF (Secure Software Development Framework):** NIST framework for secure software development (SP 800-218); supplemented for AI by SP 800-218A.
- **SVID:** the identity document of a component issued by SPIFFE (as an X.509 certificate or as a token), usually with a short validity.
- **Threat hunting:** targeted, hypothesis-driven search for as-yet-undetected attacks.
- **TIBER-EU:** the ECB's framework for threat-led testing in the financial sector; aligned with DORA since February 2025.
- **TLPT (Threat-Led Penetration Testing):** threat-led, realistic test that replicates real attack scenarios.
- **Transitive dependency:** an indirectly contained building block (a library that in turn uses further libraries).

- **UEBA:** behaviour-based detection on the basis of a learned baseline for users and systems.
- **Immutable / WORM:** storage in which data cannot be altered or deleted for a defined period.
- **VEX / CSAF:** formats for communicating the exploitability of vulnerabilities, separate from the static SBOM.
- **ZTNA / identity-aware proxy:** access to applications on the basis of the verified identity of user and device rather than via network membership; replacement for classic VPN.

Mapping Implementation Guide ↔ MRIS 1.6, Chapter 9

- **MHC-01 — Post-quantum strategy and cryptographic inventory:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; metric: cryptographic inventory coverage.
- **MHC-02 — SBOM and build provenance:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; metric: SBOM coverage.
- **MHC-03 — Phishing-resistant multi-factor authentication:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; gap cluster 6 (identity/Zero Trust) in 9.3; metric: phishing-resistant MFA share.
- **MHC-04 — Workload identity and Zero Trust network architecture:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; gap cluster 6 in 9.3; effectiveness via implementation stage gates instead of a continuous metric.
- **MHC-05 — Behaviour-based detection and kill chain correlation:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; gap cluster 7 (automation/resilience) in 9.3; metric: ATT&CK coverage (structural), supplemented by threat hunts.
- **MHC-06 — Container security and confidential computing:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; gap cluster 3 (containers/confidential computing/multi-tenancy) in 9.3; effectiveness via implementation stage gates (signature/attestation coverage).
- **MHC-07 — Multi-tenancy isolation with demonstrable separation:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; gap cluster 3 in 9.3; effectiveness via implementation stage gates (result of the separation tests).
- **MHC-08 — Immutable backups and recovery validation:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; intersection of gap clusters 3/4 in 9.3; metric: restore test success rate.
- **MHC-09 — AI-supported security testing in the pipeline:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; gap cluster 7 in 9.3; metric: patch latency for KEV listings.
- **MHC-10 — Continuous control monitoring and policy as code:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; gap cluster 5 (continuous assurance) in 9.3; metric: continuous monitoring coverage.
- **MHC-11 — SOAR-based tier-1 automation and parallel response playbooks:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; gap clusters 4/7 in 9.3; metric: Mean Time to Containment (MTTC).
- **MHC-12 — Threat-led penetration testing with Mythos scenarios:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; gap cluster 5 in 9.3; metric: TLPT findings closure rate.
- **MHC-13 — AI agent governance and harness security:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; gap clusters 2 and 6 in 9.3; effectiveness via implementation stage gates (inter alia mean time to revoke).

For all MHC additionally: RACI assignment in MRIS Annex H, KPI definitions in Annex J/G, individual assessment of the touched ISO controls in MRIS Chapters 4–6.