MRIS IMPLEMENTATION GUIDE

Implementation guideline for the Mythos Hardening Controls

Companion document to MRIS, Chapter 9 (version 1.8) — Mythos-Resistant Information Security

ISO 27002 depth · Evidence-based · Practice-oriented

Version 1.4 | July 2026

Author: Richard Peddi

TARGET AUDIENCE

CISOs and ISMS managers who want to implement the thirteen "Mythos Hardening Controls" from MRIS for ISO 27001, Chapter 9 in practice.

Purpose, independence and conventions

This document is the implementation layer for MRIS, Chapter 9. MRIS 1.8 lists the thirteen Mythos Hardening Controls (MHC) concisely in Annex A logic; this companion document supplies, for each MHC, the implementation guideline in ISO 27002 logic — such that a CISO function can follow the purpose, organisational and technical implementation, maturity level and evidence trail without additional specialist literature. It does not replace MRIS 1.8 but carries out its catalogue in practice. The source references in this guide have already been brought to the verified current status (July 2026).

Fixed structure per MHC

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

For MHC-05, MHC-09 and MHC-11, an additional section "Managed service variant" supplements the implementation for organisations without security operations of their own.

The sections "Measurement and audit evidence" contained in this guide for each MHC are deliberately kept lean. They are intended to enable a practicable evidence trail without impeding implementation through excessive documentation requirements.

Organisations with elevated audit, regulatory or customer requirement levels can extend the evidence trail voluntarily. For this purpose, a supplementary audit scheme can be used for each 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 verified
- Evidence type: policy, configuration, log, report, test record, ticket, risk acceptance
- Test frequency: occasion and frequency of effectiveness testing
- Sampling logic: sampling approach for large system landscapes
- Exceptions: documented deviations, exceptions and compensating measures
- Risk acceptance: accepted residual risks including the approving authority
- KPI/KRI: metrics used and target values
- Last effectiveness review: date and result of the most recent effectiveness test

This extended scheme is not a mandatory component of the MRIS Implementation Guide. It is intended as an optional addition for organisations that need greater audit depth, for example because of regulatory requirements, customer audits, certification preparation or a higher ISMS maturity level. For organisations at a lower maturity level, the minimum evidence trail described for each MHC is sufficient to begin with.

Licence and Disclaimer

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

Suggested citation
Peddi, Richard (2026): MRIS IMPLEMENTATION GUIDE for ISO 27001, Version 1.4.

Contents

Inhaltsverzeichnis

Prioritising 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 that is exactly what makes the implementation decision difficult. What is prioritised is not the control as such but its contribution to reducing the threats most relevant to the organisation. To that end, this chapter combines two complementary views: the measurable coverage breadth of each MHC (how many degraded controls it carries) and the threat- and effort-related sequence (threat correlation). The prioritisation remains compatible with the overarching MRIS principle: structural controls before new detection or automation capabilities.

At a glance

- Purpose: a comprehensible, threat-oriented implementation roadmap instead of a linear MHC sequence.
- Basis: coverage breadth 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 may be P0 depending on architecture and data situation.

1. Coverage breadth: how many degraded controls each MHC carries

The following overview shows how many of the controls degraded under Mythos (categories "partially degraded" and "pure friction", 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 pure friction / 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 controls that are already robust — several MHC therefore harden robust controls further, beyond the degraded ones.

MHC	Degraded controls	Also flanks robust	Character
MHC-05 Behaviour-based detection	10	A.8.15	broad, cross-cutting
MHC-02 SBOM/build provenance	6	—	supply chain
MHC-03 Phishing-resistant MFA	5	—	identity/authentication
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	verification
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 penetration test	1	A.8.29	narrow, validating
MHC-01 Post-quantum	0	A.8.24	flanks robust controls only
MHC-13 AI agent governance	0	A.5.9, A.5.16, A.8.27	flanks robust controls 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 how they relate

- Coverage breadth: unconditional leverage — how many degraded controls an MHC carries. High coverage means high leverage across many risks at once.
- Threat and effort relation: the sequence — depending on threat landscape, 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 last because they flank only robust controls. That would be wrong: MHC-13 is potentially P0 for any organisation with AI agents (it addresses the core of the Mythos threat landscape), MHC-01 for long-lived confidential data. The rule is therefore: order by leverage, but with a threat and applicability gate — never by coverage alone. "Low-hanging fruit" are then controls with high leverage, a large control gap and low effort, taking dependencies into account.

3. Assessment logic: Threat Priority Score

For the sequence, a simple, transparent score is formed per threat scenario. It makes visible whether an MHC is prioritised because of a high threat landscape rating, 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 exposed is the organisation (internet-facing, cloud, suppliers, admin access, external users)?	1 = low … 5 = highly exposed
Control gap	How large is the gap relative to the MHC target state?	1 = well covered … 5 = barely covered

Example: AI-assisted 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 substitute 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-assisted phishing / credential theft	MHC-03, MHC-05, MHC-11	MHC-03 prevents credential misuse, MHC-05 detects unusual use, MHC-11 automates the first response.	very high
Lateral movement after initial compromise	MHC-04, MHC-05, MHC-11, MHC-12	MHC-04 limits lateral movement through 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 response time; TLPT/purple teaming 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; separation must be technically enforced and tested.	high, where applicable
AI agent misuse / prompt injection / tool abuse	MHC-13, MHC-04, MHC-10, MHC-09	Agents need their own identities, limited tool permissions, policy control and a secure pipeline.	high, where AI agents are in use
Harvest now, decrypt later / quantum risk	MHC-01	Particularly relevant for long-lived confidential data and crypto-agility.	medium to high
Undetected control deviations	MHC-10	Continuous control monitoring detects deviations on an ongoing basis rather than only at audit points.	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 when they only become effective, or sensibly testable, through other MHC.

Dependency	Meaning
MHC-02 → MHC-09	AI-assisted security testing in the pipeline needs a build/CI basis 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 sound if detection rests on reliable, central and tamper-proof logs.
MHC-05 + MHC-11 → MHC-12	Threat-led tests validate whether detection and response function in realistic attack chains.
MHC-02 + MHC-04 → MHC-06	Container security becomes stronger when image provenance and workload identities are cleanly controlled.
MHC-04 + MHC-06 → MHC-07	Multi-tenancy isolation draws on 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 landscape, asset criticality and applicability. MHC-01 and MHC-13 sit formally in P2 but move up where there is long-lived confidential data (MHC-01) or productive AI agent use (MHC-13). The standard tiering is not a universal implementation plan but a starting point; each organisation must adapt it to exposure, architecture, asset criticality, existing controls, resource situation and regulatory context.

Priority	MHC	Control	Rationale
P0 — immediate	MHC-03	Phishing-resistant MFA	High likelihood of occurrence, rapid risk reduction for admins and external access.
P0 — immediate	MHC-08	Immutable backups and recovery validation	Baseline capability against ransomware and destructive attacks.
P0 — immediate	MHC-05	Behaviour-based detection and kill chain correlation	Without detection there is no sound response and no meaningful SOAR automation.
P0 — immediate	MHC-04	Workload identity and zero trust	Reduces lateral movement and is an enabler for AI agents.
P1 — next	MHC-02	SBOM and build provenance	Basis for supply chain transparency and pipeline testing.
P1 — next	MHC-09	AI-assisted security testing	Effective where a pipeline and SBOM basis exist.
P1 — next	MHC-11	SOAR tier-1 automation	Automate only once detection is stable.
P1 — next	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 in use; upgrade to P0 in that case.
P2 — risk-based	MHC-01	Post-quantum strategy	Prioritise higher for long-lived confidential data.
P2 — risk-based	MHC-12	Threat-led penetration testing	Validation after baseline implementation, regularly thereafter.

Note on the market situation (July 2026): the multi-vendor development and the disclosure waves now beginning (MRIS Ch. 2.5) raise the threat landscape values of the supply chain and exploit volume scenarios for most profiles; upgrading MHC-02 and MHC-09 within the tiering is then justified.

7. Approach in four steps

- Step 1 — establish the threat landscape: identify relevant attacker types, TTPs and attack scenarios; take sector-specific threats into account (financial services, critical services, supply chain, cloud, AI); assess threats by currency, likelihood of occurrence and proximity to the organisation.
- Step 2 — map threats to MHC: for each threat, determine which MHC act preventively, detectively, reactively or in a validating capacity; weight controls with several threat relations more heavily; provide a sound rationale in the Statement of Applicability for controls that are not applicable.
- Step 3 — calculate the score: rate threat relevance, business impact, exposure and control gap from 1 to 5 each; form the score and take effort and dependencies into account; where risk is equal, 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 technical groundwork), P2 (context-dependent controls and validation); reassess regularly as soon as the threat landscape, 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, an assessment is made of which controls act preventively, detectively, reactively or in a validating capacity. Prioritisation follows 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 particular point in time.

8. Sensitivity of the Threat Priority Score

The score is a qualitative ordinal index; its ordering must be robust against changes in weighting. Verification step per organisation: vary all factor weights individually by ±20%; if the P0 set changes, the prioritisation must be re-justified or adjusted (documented in the risk process to ISO/IEC 27005).

For the standard tiering in this guide: each of the four P0 controls rests on at least two dominant factors (threat landscape and impact, or enabler role) as well as on the dependency rules from Section 5. A ±20% variation in individual weights is therefore expected to shift only ranks within P1/P2 (typical candidates for change: MHC-06 and MHC-12, depending on exposure), not the P0 set. Organisation-specific deviations remain permissible and must be documented.

MHC-01 — Post-quantum strategy and cryptographic inventory

At a glance

- Operational benefit: protects long-lived confidential data against the "harvest now, decrypt later" pattern and creates the ability to replace cryptographic methods swiftly where needed.
- Policy affected: 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 using 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 recorded 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 for migrating to quantum-safe methods in good time. During the transition, hybrid methods are used.

2. Purpose and threat context

Future quantum computers will be able to break encryption in widespread use today (such as RSA and elliptic curves). Attackers are therefore already intercepting encrypted data today in order to decrypt it later — the "harvest now, decrypt later" pattern. Data that must remain confidential for many years is particularly affected. AI additionally accelerates the discovery and exploitation of weak or outdated crypto implementations. The protection objective is to migrate long-lived confidential data 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 extended with the obligation to maintain a central crypto inventory and with a migration strategy towards quantum-safe methods.
- Responsibilities: a function is named for maintaining the inventory and updating the strategy, and is assigned in the Statement of Applicability (template: MRIS Annex E).
- Risk analysis and SoA: the migration sequence is derived from the risk assessment — data with a long confidentiality horizon first. Risk treated: later decryption of data exfiltrated today (linked 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, resources 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: record all algorithms, key lengths, protocols, certificates and underlying libraries in use; prioritise by protection need and confidentiality horizon.
- 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) in preparation as a further signature scheme.
- 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 replaced without deep changes to applications (central crypto libraries, configurable algorithms).
- Further reading: 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, verify that the method actually used 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 provider does not initially know exactly which encryption is used in its services and libraries. It therefore first creates a central register of all methods, keys and certificates used, and records for each data holding how long it must remain confidential. For the connections carrying particularly long-lived data, it switches the key exchange to a combined method using classical and quantum-safe cryptography at the same time. This keeps that data 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 development of its own uses mainly off-the-shelf software and cloud services. It cannot switch the encryption itself, but it gains an overview of where particularly long-lived confidential data resides (for example HR or contract documents) and asks its providers for their roadmap to quantum-safe migration. It includes support for quantum-safe methods as a requirement in new contracts. It thus steers the topic through overview and procurement, even without technical implementation of its own.

6. Maturity path (cumulative)

- Initial: cryptographic inventory documented.
- Defined: migration strategy adopted; hybrid methods piloted in one area.
- Managed: hybrid or quantum-safe methods in production use; annual review automated.

7. Measurement and audit evidence

- Metric: share of the methods in use that are recorded in the inventory and classified by migration priority (target: complete coverage of the areas requiring protection).
- Evidence: crypto inventory, migration strategy with triggers and timeline, pilot and production evidence for hybrid methods.

- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — inventory and strategy; responsible? — named function; frequency? — annual or event-driven review; tolerance? — no long-lived confidential data without a migration plan.

8. Typical mistakes

- A strategy is formulated without a complete inventory being in place — the most important locations remain unknown.
- Hybrid methods are introduced across the board without prioritising by confidentiality horizon — much effort without a discernible focus of benefit.
- Encryption is hard-wired into the code, so that a later change of method requires intervention in the application each time.
- The migration is treated purely as a future topic, even though today's data exfiltration already determines tomorrow's risk.

9. Delimitation, residual risk and references

- Delimitation: concerns the cryptography in use; the management of 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 also comparatively young and continue to be monitored.
- 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 coordinated PQC migration.

MHC-02 — SBOM and build provenance

At a glance

- Operational benefit: makes immediately visible which third-party components are contained in your own software, so that when a new vulnerability emerges it is clear within minutes rather than days whether and where you are affected. Precondition: SBOM generation and vulnerability correlation at least at the Defined level.
- Policy affected: 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 build pipelines and on the depth of the supply chain. The concrete effort depends on the organisation's resources.
- Primary metric: share of delivered 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 in-house and passed on, a bill of materials of the components it contains (SBOM) is generated automatically and reconciled against vulnerability databases. For critical artefacts, comprehensible evidence is additionally provided of how and from what they were built (build provenance).

2. Purpose and threat context

Modern software consists largely 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 — it is often unclear for days, without an overview, whether and where you are affected. AI drastically shortens the time between a vulnerability becoming known and a working attack. A current bill of materials makes your own exposure immediately checkable; evidence of provenance 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 mandates automatic SBOM generation for in-house and onward-supplied artefacts, as well as a contractual SBOM requirement for critical software suppliers.
- Responsibilities: the measure is assigned to a function in the Statement of Applicability (template: MRIS Annex E); development and procurement work together.
- Risk analysis and SoA: critical artefacts and suppliers are prioritised from the risk assessment. Risk treated: undetected vulnerabilities or manipulation in third-party components (linked to the risk assessment bridge, MRIS Annex F).
- Processes: a defined procedure for how, when a new vulnerability becomes known, exposure is determined from the bills of materials and remediated; 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). Indirect (transitive) dependencies are recorded as well, 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 under version control.
- Vulnerability reconciliation: automated correlation of 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, a comprehensible and tamper-proof record is kept of which sources and which build process they originated from (SLSA build provenance, a higher level for critical artefacts).
- Tools (non-binding examples): generation with Syft, cdxgen or Trivy; central management and vulnerability correlation with, for example, Dependency-Track.

- Further reading: 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 widely used library (e.g. a Log4j-type flaw), check from the bills of materials alone whether and which of your own artefacts contain the affected component — the result must be available within minutes.

5. Implementation examples

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

Example B — general department. An organisation without development of its own cannot generate an SBOM 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 follow up with the manufacturer specifically on whether the product in use is affected, instead of waiting for vague collective notifications. It thus uses the principle through procurement, even without a pipeline of its own.

6. Maturity path (cumulative)

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

7. Measurement and audit evidence

- Metric: SBOM coverage — share of delivered artefacts with a current, automatically generated bill of materials (target: complete coverage of your own artefacts).
- Evidence: SBOM files per version, pipeline configuration, vulnerability reconciliation reports, provenance evidence for critical artefacts.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — pipeline configuration and bills of materials; responsible? — development/product owners; frequency? — at every build, reconciliation ongoing; tolerance? — no delivered artefact without a current bill of materials.

8. Typical mistakes

- Bills of materials are generated but never evaluated — when a vulnerability arises an SBOM exists, but nobody reconciles it.
- Only top-level dependencies are recorded; precisely the deeply nested building blocks in which vulnerabilities reside are missing.
- The SBOM is created once and not maintained per version, so that it no longer matches what was actually delivered.

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

9. Delimitation, residual risk and references

- Delimitation: concerns transparency and provenance of the software building blocks; the actual closing of vulnerabilities is part of vulnerability and patch management, and 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 — that is what provenance is for.
- 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 duties from 2026, main obligations for products placed on the market after 11.12.2027); BSI TR-03183-2; DORA Art. 28; NIS2-DVO 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 sites and thereby removes the basis for AI-assisted, deceptively authentic phishing.
- Policy affected: access control and authentication policy.
- 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 using 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-DVO No. 11.2.

1. Control statement

Privileged access, externally reachable services and access to systems requiring particular protection 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 context

Phishing has increased sharply in quality and volume through AI: fake login pages, deceptively authentic emails and codes relayed in real time bypass classical two-factor methods such as SMS or app confirmation. Phishing-resistant methods bind the login credential cryptographically 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 mandates phishing-resistant methods for privileged, externally reachable and particularly protection-worthy access; SMS and simple push methods are excluded for new access, and legacy methods are replaced under a transition plan.
- Responsibilities: the migration is assigned to a function in the Statement of Applicability (template: MRIS Annex E).
- Risk analysis and SoA: the target maturity level is derived from the risk assessment; privileged and externally reachable access first.
- Processes: issue and withdrawal of tokens or passkeys as a governed process; fallback procedures for loss that do not undermine the protection level; decommissioning of legacy methods after the transition.
- Training: users are supported during set-up 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 credentials being passed to fake sites.
- Assurance levels under NIST SP 800-63B Revision 4: at AAL2, synchronisable passkeys (across several devices) are also permitted; for the highest level, AAL3, a device-bound, non-exportable key is required (hardware token or smart card), and synchronisable passkeys are not permitted there.
- Switching off 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 through the identity platform rather than through login screens of their own.
- Further reading: requirements per assurance 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 carried out; 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 the access of its administrators and developers with a password and a second factor confirmed via an app. A well-executed phishing attack intercepts both password and confirmation in real time. The company first upgrades 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 stage, login becomes passwordless organisation-wide and the old SMS and app methods are switched off.

Example B — general department. An organisation without development of its own uses Microsoft 365 and a few other cloud services. Staff have so far logged in with a password and an SMS code — vulnerable to phishing. The organisation enables passkey login in its identity platform and issues hardware security keys for roles requiring particular protection. Login now takes place without a password and without SMS; a central access rule expressly requires a phishing-resistant method for

sensitive applications. The protection can thus be enforced through the settings of the platform in use, even without development of its own.

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 using phishing-resistant methods, broken down by privileged, externally reachable and other access (target: complete for privileged and externally reachable access).
- Evidence: identity platform configuration, list of permitted methods per access class, evidence that SMS methods have been switched off.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — authentication policy and platform configuration; responsible? — identity/IT management; frequency? — ongoing; tolerance? — no privileged or externally reachable access relying solely on SMS or simple push confirmation.

8. Typical mistakes

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

9. Delimitation, residual risk and references

- Delimitation: concerns the login credential of users; the identity of services and workloads is covered by MHC-04.
- Residual risk: attacks may shift to recovery and fallback procedures; session hijacking after successful login also remains a topic in its own right.
- Effectiveness limit: phishing-resistant does not mean phishing-proof. FIDO2/passkeys devalue classical credential phishing and MFA fatigue, but not session or token theft via malware, adversary-in-the-middle after login, or social engineering at the help desk.
- 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.8.23 Web filtering.
- Framework: NIST SP 800-63B Revision 4 (assurance levels AAL2/AAL3); FIDO2/WebAuthn (FIDO Alliance, W3C); C5:2026 IAM-08; NIS2-DVO No. 11.2.

MHC-04 — Workload identity and zero trust network architecture

At a glance

- Operational benefit: prevents lateral spread by an attacker after initial compromise; a service that has been taken over can no longer move on to neighbouring services.
- Policy affected: access control and network security policy.
- Dependencies: forms the identity basis for MHC-13; has no precondition itself.
- Effort drivers: migrating existing services ties up more resources than a greenfield build; the concrete effort depends on the organisation's available resources.
- Primary metric: share of production service-to-service connections using workload identity and mTLS.
- Standards touched (without claim of fulfilment): among others ISO/IEC 27002 A.8.20/A.8.21/A.8.22, NIST SP 800-207 and 800-207A, NIS2-DVO 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 context

Attackers accelerated by AI move from initial compromise to lateral movement within minutes. Classical perimeter, VPN and jump host models trust a service purely because of its network position; a host that has been taken over thereby effectively gains access to neighbouring services. An identity of its own per workload removes the basis for this pattern: without a valid, short-lived identity no connection is established. The objective is to close the classical paths for lateral spread structurally. The protection rests on a cryptographic barrier, not on merely delaying an attack.

3. Organisational implementation (CISO perspective)

- Policy: the access control and network security policy is extended with the principle that production services identify themselves through their own, technically managed identity; trust based purely on network addresses is permissible only on a time-limited and documented basis.
- Responsibilities: the measure is assigned to a named function in the Statement of Applicability (template MRIS Annex E: IT management responsible for the build-out, 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 protection-worthy data first. Risk treated: lateral spread after initial compromise (linked to the risk assessment bridge, MRIS Annex F).
- Processes: a documented process is needed for issuing, regularly renewing and revoking service identities; the time to revoke an identity is maintained as a metric.
- Training: development teams are bound to the changed requirement (no static access keys in code) and briefed accordingly.
- AI agents: it is established that AI agents operate under their own, technically managed identity and not under personal user accounts; the detailed rules are set out 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 a SPIRE agent on each 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 authenticate each other 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): a sidecar proxy per service handles mTLS, identity verification and policy enforcement transparently to the application code. Authorisation is via identity policies (which service identity may call which service) instead of IP allowlists.
- Segmentation: identity-based authorisation with default-deny between services supplements or replaces network-based micro-segmentation.
- External and privileged access: ZTNA or an identity-aware proxy instead of a classical VPN; access is bound to user identity, device identity and context.
- Migration: permissive mode (cleartext and mTLS in parallel) serves only as a short measurement period; strict mode is enforced thereafter. Sequence: the most critical service-to-service paths first.
- AI agents: time-limited, capability-scoped identities per tool call instead of developer credentials.
- Further reading: architecture model (identity tier vs network tier, ingress/egress gateways, multi-cloud) in NIST SP 800-207A, sections 3–4; zero trust 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, attempt a connection to a protected service — it must be refused. In addition: an expired credential must no longer permit a connection.

5. Implementation examples

Example A — development/platform context. A mid-sized software provider runs its application as many individual, interconnected services. Until now these services trusted one another purely because they ran on the same internal network; if an attacker took over one service, they moved from there to all the others unimpeded. The company changes this basis: each service receives its own, continuously rotating technical credential, and without a valid credential no other service accepts a connection. It starts with the few services that process customer data; the migration is later extended to all of them. The result: a service that has been taken over stands alone and can no longer move sideways to neighbouring services.

Example B — general department. An organisation without software development of its own runs 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 deemed trustworthy. This carries the same underlying problem — a compromised device on the network gains far-reaching access. In future, the organisation binds access to internal applications to the verified identity of user and device rather than to mere network membership; an upstream access service checks login, device and context at every access. It starts with the most protection-worthy applications and the administrative access paths. *Applicability note:* an organisation that uses only off-the-shelf cloud services and operates no infrastructure of its own is barely affected by MHC-04; the control is carried in the Statement of

Applicability as not applicable. Protection here is the provider's responsibility and is required contractually through 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 production microservices; ZTNA for privileged remote access.
- Managed: full coverage including dev/test; identity-based micro-segmentation; AI agents with capability-scoped identities.

7. Measurement and audit evidence

- Metrics: share of production service-to-service connections with mTLS/workload identity (target for Defined: 100% in production); time to revoke a workload identity.
- Stage gate (indicative value): Initial ≥ 80% of privileged service workloads with a SPIFFE/SPIRE identity; Defined mTLS mandatory for east-west traffic ≥ 95%; Managed attestation-backed access policies in production. Gate fulfilment as a yes/no record in project controlling (MRIS Ch. 10.6).
- Evidence: SPIRE registry (inventory of issued identities), service mesh policy configuration, logs of mTLS enforcement.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — via the SPIRE registry; responsible? — platform team; frequency? — ongoing (renewal at minute to hour intervals); tolerance? — no production shared secrets stored in cleartext on critical paths.

8. Typical mistakes

- mTLS remains in permissive mode (cleartext continues to be accepted) — no protective effect.
- Identities with excessive validity (days instead of minutes) — the benefit of short lifetimes is lost.
- AI agents continue to operate under personal developer accounts — traceability and limitation of permissions are missing.
- IP-based allowances as a permanent parallel path instead of a time-limited, documented exception.

9. Delimitation, residual risk and references

- Delimitation: protects the service-to-service level; authentication of end users is covered by MHC-03; governance of AI agents is deepened in MHC-13.
- Residual risk: a compromised issuing infrastructure (SPIRE) or a legitimately credentialed but compromised service remain effective attack paths; an identity does not replace the minimisation of permissions (least privilege remains necessary).
- Effectiveness limit: zero trust is an architectural path, not a product. Partially implemented segmentation or workload identity can create false security; effectiveness arises only when authentication, authorisation and least privilege take effect 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 Network 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 catches disguised attacks carried out with legitimate means.
- Policy affected: 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 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 under 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-DVO No. 3.2; DORA Art. 10.

1. Control statement

Attacks are detected on the basis of behavioural anomalies and the linking of connected events, not solely on the basis of individual known signatures. Detection is oriented to 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 context

AI-assisted attackers proceed inconspicuously: they use built-in system tools instead of imported malware and disguise their control channels, so that each individual step appears harmless in isolation. Classical signature-based detection does not trigger. In addition, agentic attackers break their approach into many small actions, each unremarkable on its own — the danger only becomes apparent in context. The protection objective is to detect attacks by their behaviour and by the sequence of connected steps, even where no known signature exists.

3. Organisational implementation (CISO perspective)

- Policy: the monitoring policy establishes 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 E); threat hunting is anchored as a function of its own with a minimum capacity.
- Risk analysis and SoA: the most important techniques to be detected are derived from the threat landscape and the risk assessment. Risk treated: disguised attacks carried out with legitimate means that remain below the threshold of individual alerts.

- Processes: a documented threat hunting methodology (e.g. PEAK or TaHiTI) with regular, hypothesis-based runs; regular review and extension of detection coverage; a governed handover of detected incidents into response.
- Capacity: sufficient analysis capacity for hunting and evaluation; alignment of security operations towards analysis rather than mere 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 under MITRE ATT&CK: detection rules are aligned to the techniques of the ATT&CK catalogue (continuously updated, currently v19) rather than to isolated individual signatures; coverage is measured (e.g. with DeTT&CT) and confirmed by controlled attack tests (e.g. Atomic Red Team). Indicative value: 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 individually unremarkable steps stands out as a whole.
- Detection priorities: use of built-in tools (e.g. unusual scripting or task scheduling activity, unusual system processes) and disguised control channels (e.g. unusual DNS or encrypted traffic with long dormant periods).
- Robustness of your own detection AI: where detection itself relies on machine learning, the models are tested against deception (adversarial examples) and data poisoning.
- Further reading: 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 individually unremarkable steps (e.g. via Atomic Red Team) is carried out; the chain must be detected as a connected incident, not merely as separate, unrelated events.

5. Implementation examples

Example A — development/platform context. A company with its own security operations finds that its existing detection only triggers on known malware. An attacker using exclusively built-in system tools remains undetected. The company therefore aligns its detection rules to a recognised catalogue of attack techniques and measures what share of the most important techniques is actually covered. Suspicious sequences of steps are linked into a chain across time and systems. In addition, a small team regularly searches in targeted fashion for traces of previously undetected attacks. As a result, an attacker moving quietly with legitimate means is now noticed as well.

Example B — general department. An organisation without 24/7 operations of its own cannot build such detection itself. It therefore procures security monitoring as a service (managed detection and response) and, when selecting a provider, ensures that detection is behaviour- and technique-based, that connected attack chains are evaluated, and that a fixed response time is contractually assured. Internally it names a contact person who receives reported incidents and steers remediation. It thus achieves a comparable level of protection through 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: learning detection that is robust against deception; ongoing review and confirmation of coverage.

7. Measurement and audit evidence

- Metric: coverage of the most important attack techniques under MITRE ATT&CK (measured and confirmed by attack tests); in addition, the number and results of threat hunts carried out.
- Evidence: coverage overview (e.g. DeTT&CT), results of attack tests, documented hunting runs with technique mapping.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — monitoring policy, coverage overview, hunting records; responsible? — detection/SOC management; frequency? — ongoing operations, regular hunts and coverage review; tolerance? — no important techniques left permanently unmonitored.

8. Typical mistakes

- Detection remains signature-based; attacks using built-in tools raise no alert.
- 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 effectively never carried out.

9. Delimitation, residual risk and references

- Delimitation: concerns the detection of attacks; the subsequent automated response is covered by MHC-11, automated testing of your own defences by MHC-09.
- Residual risk: detection is not prevention — very slow, heavily disguised or novel attacks may remain undetected; where detection relies on machine learning, 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-DVO No. 3.2; DORA Art. 10; NIST SP 800-94 (2007 edition); methodology examples PEAK and TaHiTI, coverage measurement with DeTT&CT, attack testing with Atomic Red Team.

Managed service variant (for organisations without a SOC of their own)

- Service to be procured: managed detection and response (MDR/XDR) with behaviour-based detection, an ATT&CK-oriented use case library and threat hunting as a contractual component.
- Evidence to be required contractually: documented use case coverage with ATT&CK mapping (at minimum the most important techniques for your own sector); regular MTTD/MTTC reporting; hunting reports with hypothesis and result; results of controlled attack tests.

- Minimum service levels: continuous alert handling (24/7); a contractually defined containment time for high-confidence alerts; named escalation paths with response times; quarterly service review with evidence of coverage.
- Remaining in-house effort: connecting log and telemetry sources, maintaining asset context, approval decisions; responsibility in the Statement of Applicability remains with the organisation.

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.
- Policy affected: policy on secure development and secure operations (container and cloud operations).
- Dependencies: no precondition; presupposes a container/cloud platform; works together with MHC-02 (image provenance) 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 production containers running from signed images verified before start-up; in addition, confidential computing coverage for highly sensitive workloads.
- Standards touched (without claim of fulfilment): ISO/IEC 27002 A.5.23/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 verified before start-up and drawn from controlled sources, with enforcement through 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 advanced hardening for workloads requiring particular protection, 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 context

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

3. Organisational implementation (CISO perspective)

- Policy: the operations policy mandates signed, verified images from controlled sources, and confidential computing for clearly named workloads requiring particular protection.
- Responsibilities: assigned in the Statement of Applicability (template: MRIS Annex E); the platform/operations team and security work together.
- Risk analysis and SoA: which workloads require confidential computing is derived from protection needs (not everything needs it). Risk treated: manipulated images and access to data during processing (linked to the risk assessment bridge, MRIS Annex F).
- Processes: controlled image sources and approvals; a governed approach to vulnerability findings before deployment; where services are drawn from the cloud, evidence of provider capabilities (confidential computing, attestation).

4. Technical implementation (for IT specialists)

- Signed images: base images and your own images are signed (e.g. with Sigstore/cosign) and drawn from controlled registries; the signature is verified before start-up.
- Enforcement at deployment: a platform admission controller (e.g. OPA Gatekeeper or Kyverno in Kubernetes) permits only signed, verified images and blocks the rest.
- Image verification: automated vulnerability scans of 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 in memory as well; before use, remote attestation proves that the environment is genuine and unaltered.
- Further reading: container risks and protective measures in NIST SP 800-190 (2017 edition, still the authoritative reference); vendor-neutral fundamentals from the Confidential Computing Consortium; requirements in C5:2026 OPS-32 to OPS-35.
- Effectiveness test: attempt to start an unsigned or unverified image in the production environment — start-up must be prevented by the platform.

5. Implementation examples

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

Applicability note (in place of example B). Container security and confidential computing presuppose that the organisation builds or operates containers itself. An organisation using only off-the-shelf SaaS services cannot implement this measure itself; for it, the measure becomes a question of procurement and evidence: the provider is asked whether it signs and verifies images, prevents the start-up of unapproved images, and offers protected execution environments for particularly sensitive data — evidenced, for example, by certificates or audit reports (e.g. C5).

6. Maturity path (cumulative)

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

7. Measurement and audit evidence

- Metric: share of production containers running from signed images verified before start-up (target: complete); in addition, confidential computing coverage for the workloads classified as highly sensitive.
- Evidence: admission controller configuration, signature and scan logs, attestation evidence for the protected environments.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — operations policy, platform configuration; responsible? — platform/operations management; frequency? — at every deployment, scans ongoing; tolerance? — no production container without a valid signature and verification.

8. Typical mistakes

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

9. Delimitation, residual risk and references

- Delimitation: concerns the integrity of containers and the protection of data during processing; the provenance of the components they contain is covered by MHC-02, the separation of multiple tenants by MHC-07.
- Residual risk: confidential computing protects processing but not flaws 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 are to be planned, assessed and evidenced separately.
- ISO/IEC 27002:2022 (indirect): A.5.23 Information security for use of cloud services; A.8.31 Separation of development, test and production environments.
- 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 the data of other tenants — the impact remains confined to a single tenant.
- Policy affected: policy on secure architecture and tenant separation (for multi-tenant services).
- Dependencies: no precondition; concerns organisations running several customers on a shared platform; draws on key separation (reference 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: results of regular, ideally automated separation tests (no demonstrable cross-tenant access).
- 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 evidenced through regular, ideally automated tests — not merely asserted conceptually.

2. Purpose and threat context

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 they can reach the data of other tenants from there. AI-assisted attackers search systematically for precisely such gaps in the separation. Conceptual separation alone is not sufficient; what matters is that the separation actually takes effect and that this is evidenced. The protection objective is to confine the impact of an attack demonstrably to a single tenant.

3. Organisational implementation (CISO perspective)

- Policy: the architecture policy mandates, for multi-tenant services, documented separation concepts for data, networks and compute resources, together with regular evidence of them.
- Responsibilities: assigned in the Statement of Applicability (template: MRIS Annex E); architecture and operations work together.
- Risk analysis and SoA: the required level of separation (logical or physical) is derived from the protection needs of the customer data. Risk treated: cross-tenant access (linked to the risk assessment bridge, MRIS Annex F).
- Processes: regular separation tests with records; treatment of identified weaknesses; provision of evidence 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 data holdings.

- 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.
- Evidence: regular, ideally automated tests that deliberately attempt to access another tenant from within one tenant; the result (no access possible) is documented.
- Further reading: 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, deliberately attempt to access data or resources of another tenant — access must fail, and the result is recorded.

5. Implementation examples

Example A — development/platform context. A SaaS provider runs many customers on a shared platform. Until now the separation has been described only in architecture documents and never verified. The provider introduces tenant-specific keys, separates the network areas per customer and ensures that compute loads of different customers are not shared in an uncontrolled way. Above all, however, it sets up regular, automated tests that attempt to force access to other tenants from within one tenant. If such an attempt fails, the separation is evidenced; if it succeeds, there is a clear finding to remediate. An asserted separation thus becomes a demonstrated one.

Applicability note (in place of example B). This measure is aimed at organisations that operate multi-tenant services themselves. An organisation that merely uses such services does not implement the separation itself; for it, the measure becomes a question of procurement and evidence: the provider is asked how it implements and evidences tenant separation (for example 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 reviews.

7. Measurement and audit evidence

- Metric: results of the regular separation tests (no demonstrable cross-tenant access); frequency and coverage of the tests.
- Stage gate (indicative value): one passed, documented tenant separation test per level (Initial annually, internal; Defined per major release; Managed additionally by independent third parties). Evidence in project controlling (MRIS Ch. 10.6).
- Evidence: separation concepts, test records, remediation evidence for identified weaknesses.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — separation concepts and test records; responsible? — architecture/operations management; frequency? — regular tests; tolerance? — no demonstrated cross-tenant access.

8. Typical mistakes

- The separation is only described conceptually and never evidenced through tests.

- Data separation is implemented but network or resource separation is not — the crossover occurs via the unprotected path.
- Tests examine only the normal case, not the deliberate circumvention attempt from within a tenant.
- On extensions (new features, new tenants), the separation is not re-verified.

9. Delimitation, residual risk and references

- Delimitation: concerns the separation between tenants; the protection of processing itself is covered by MHC-06, the identity of workloads by MHC-04.
- Residual risk: a shared platform retains a shared base layer; flaws in that shared layer can affect several tenants despite 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.
- Policy affected: 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-DVO No. 4.1; NIST SP 800-34, SP 1800-11/-25.

1. Control statement

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

2. Purpose and threat context

Modern, AI-assisted attacks encrypt or corrupt data and deliberately target backups as well, in order to prevent recovery. An attacker with far-reaching access can encrypt or delete backups reachable online; an immutable or network-isolated copy remains unaffected. The protection objective is to be able to fall back on a clean, unaltered data basis at any time after an incident — and to be sure that recovery actually succeeds.

3. Organisational implementation (CISO perspective)

- Policy: the backup and recovery policy mandates 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 function in the Statement of Applicability (template: MRIS Annex E); the results of recovery tests are reported.
- Risk analysis and SoA: recovery objectives (how quickly, to what data state) are derived from the risk and impact analysis; particularly business-critical data first.
- Processes: a fixed plan for regular recovery tests with records; separate administration of 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 off-site, at least one immutable or network-isolated (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 isolated from the network.
- Separate access paths: the backup infrastructure uses its own accounts and permissions, separate from production administrative access, so that a compromised production account cannot reach the backups.
- Integrity checking: regular, ideally automated verification that the backups are complete and unaltered.
- Further reading: approach and building blocks in NIST SP 1800-11 (recovery from ransomware) and SP 1800-25 (protecting 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: perform a full recovery test from an immutable copy; in addition, attempt to alter or delete a backup using a production administrative account — this must be refused.

5. Implementation examples

Example A — development/platform context. A provider backs up its systems regularly, but all backups are reachable through the same administrative access paths as the production systems. An attacker who takes over such access could encrypt the backups as well. The provider 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 production access. Every quarter 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 IT development of its own backs up its files and mailboxes via a cloud service. It ensures that at least one copy is retained immutably (many services offer a retention or lock function for this) and that the administration of this backup does not depend on the same accounts as day-to-day administration. Once a quarter it restores files from the backup on a sample basis and records that it worked. This achieves the core of the protection even without a backup infrastructure of its own.

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: 100%).
- Evidence: records of the recovery tests, evidence of immutability (retention and lock settings), separate rights assignment for the backup infrastructure, integrity checking reports.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — policy and test records; responsible? — IT/backup owners; frequency? — quarterly tests, ongoing integrity checking; tolerance? — no business-critical data holding without an immutable or isolated copy and without a passed recovery test.

8. Typical mistakes

- Backups exist but are reachable through the same access paths as the production systems — an attacker encrypts both.
- Backups are created but recovery is never tested in full; when it matters, parts are missing or the process takes too long.
- "Immutable" is merely configured but not verified; too short a lock period leaves the copy vulnerable.
- The backup does not cover all necessary data (such as configurations or keys), so that a complete recovery fails.

9. Delimitation, residual risk and references

- Delimitation: concerns backup and recovery; continuous availability through redundancy is covered by A.8.14, restart planning by A.5.29/A.5.30.
- Residual risk: a network-isolated copy is by its nature less current; 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 — separate permissions, separate identity, no deletion or shortening rights for compromised admin accounts; what is decisive is 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-DVO 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-assisted security testing in the pipeline

At a glance

- Operational benefit: finds vulnerabilities before delivery — automatically on every change — and denies the attacker the window between a flaw becoming known and its being closed.
- Policy affected: policy on secure software development.
- Dependencies: presupposes an automated build/CI pipeline; uses the bills of materials from MHC-02 for component reconciliation.
- Effort drivers: depends on the number and structure of pipelines and the volume of code produced. The concrete effort depends on the organisation's resources.
- Primary metric: time from an actively exploited vulnerability (KEV) becoming known to the patch being rolled out; in addition, share of pipelines with security testing and a 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 on every change; serious findings block the merge. AI-generated code is additionally checked as a risk category in its own right.

2. Purpose and threat context

In the past there was time to patch between a vulnerability becoming known and a working attack. AI shortens this window drastically — in some cases to hours. Anyone who finds vulnerabilities late is too slow. In addition, AI coding assistants regularly produce insecure patterns (hardcoded credentials, missing input validation, insecure defaults, outdated building blocks). The protection objective is to find and remediate vulnerabilities automatically as early as possible — on every change — before they reach production.

3. Organisational implementation (CISO perspective)

- Policy: the development policy mandates automatic security checks in the pipeline, blocking of serious findings, and separate checking of AI-generated code.
- Responsibilities: assigned in the Statement of Applicability (template: MRIS Annex E); AI-generated code is carried as a risk category of its own.
- Risk analysis and SoA: thresholds for blocking findings and the handling of actively exploited vulnerabilities (KEV) are defined. Risk treated: exploitable vulnerabilities that reach production undetected (linked to the risk assessment bridge, MRIS Annex F).
- Processes: a governed procedure for blocking findings and for KEV cases (expedited remediation); regular evaluation of test results; penetration tests with scenarios tailored to the current threat landscape.

4. Technical implementation (for IT specialists)

- Test classes in the pipeline: static code analysis (SAST), dynamic testing of the running application (DAST) and dependency testing (SCA) against vulnerability databases — automatically on every change.

- Blocking (pre-merge block): serious findings (high/critical) above a defined confidence level prevent the merge; actively exploited vulnerabilities (KEV) block immediately.
- Running operations: regular (ideally daily) vulnerability scans of systems already in operation, not only of new code.
- AI code as a category of its own: additional test rules against the typical weaknesses of AI-generated code (e.g. via pre-commit checks); regular evaluation with trend tracking; AI-generated code is identified in the Statement of Applicability as a risk category of its own.
- Further reading: secure development practices in NIST SP 800-218 (SSDF), supplemented for AI model development by SP 800-218A; application security test criteria in OWASP ASVS; scan requirements in C5:2026 OPS-25.
- Effectiveness test: deliberately introduce a known insecure element in a test branch (e.g. a hardcoded credential); the pipeline must report the finding and block the merge.
- Effectiveness limit: AI-assisted testing does not replace expert judgement. It particularly misses logic, authorisation and architecture weaknesses — precisely the findings with high impact — and produces both false positives and false negatives. "The tool ran" is not "the vulnerability is 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 provider has so far tested security late, shortly before a release. When a new vulnerability becomes known it is therefore regularly too slow. It builds security checks directly into its pipeline: code, dependencies and the running application are checked automatically on every change, and serious findings prevent the merge. For actively exploited vulnerabilities there is an expedited path to close them immediately. Since its developers use AI coding assistants, it adds test rules against the typical errors of such tools and evaluates those findings separately. Flaws are thus found before they are shipped.

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

6. Maturity path (cumulative)

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

7. Measurement and audit evidence

- Metric: time from an actively exploited vulnerability (KEV) becoming known to the patch being rolled out (indicative value: ideally under 24 hours); in addition, share of pipelines with security testing and a pre-merge block.

- Evidence: pipeline configuration, test and blocking logs, evaluation of AI code findings, penetration test reports.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — development policy, pipeline configuration; responsible? — development/product owners; frequency? — on every change, scans daily; tolerance? — no delivery with open serious findings.

8. Typical mistakes

- The checks run but do not block; serious findings are reported and shipped anyway.
- Only 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 specifically sought.
- Thresholds are set so strictly that the pipeline blocks constantly; as a result, checks are circumvented rather than remediated.

9. Delimitation, residual risk and references

- Delimitation: concerns automated testing during development; transparency of components is covered by MHC-02, 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 a supplement.
- ISO/IEC 27002:2022: A.8.8 Management of technical vulnerabilities; A.8.25 Secure development lifecycle; 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 the tightening 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.

Managed service variant (for organisations without security testing of their own)

- Service to be procured: AI-assisted vulnerability and penetration testing as a service (PTaaS) or continuous attack simulation (BAS) operated by the provider.
- Evidence to be required contractually: test reports with methodology and coverage; reporting time for critical findings; re-test confirmation after remediation; reconciliation of findings with KEV/EPSS.
- Minimum service levels: continuous or at least monthly test cycles for externally reachable systems; reporting of critical findings within 24 hours; an SLA for re-tests after remediation.
- Remaining in-house effort: prioritisation, remediation, vulnerability ownership and escalation; logic, authorisation and architecture flaws remain the subject of manual review ("the tool ran" is not "the vulnerability is blocked").

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 cycle) to minutes, because requirements are checked continuously and automatically. Target state from the Defined level; presupposes automated target specifications (policy as code).
- Policy affected: policy on monitoring the effectiveness and observance of security requirements.
- Dependencies: no precondition; presupposes requirements that can be formulated in machine-readable form and access to system/configuration data.
- Effort drivers: depends on the number and nature of the requirements to be checked and on the heterogeneity of the system landscape. The concrete effort depends on the organisation's resources.
- Primary metric: share of controls carried in the SoA whose observance 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-DVO No. 2.2; DORA Art. 6(5); NIST SP 800-137.

1. Control statement

Observance of 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 deployment and operational processes; deviations automatically trigger a response.

2. Purpose and threat context

Periodic audits check only at fixed points in time (quarterly, say); in between, deviations remain undetected. AI-assisted attackers exploit precisely this gap: they create — or find — a misconfiguration and use it long before the next audit takes place. In addition, agentic attackers break their approach into steps that each look compliant in isolation. Continuous, automated checking detects deviations immediately and brings them directly to response. The protection objective is to reduce the interval between a deviation and its detection to a minimum.

3. Organisational implementation (CISO perspective)

- Policy: the monitoring policy establishes that observance of material requirements is checked continuously and automatically and that deviations are dealt with immediately — in addition to, not instead of, the audits that are required in any case.
- Responsibilities: maintenance of the rules and evaluation are assigned in the Statement of Applicability (template: MRIS Annex E); professional responsibility normally lies with ISMS management.
- Risk analysis and SoA: it is determined 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 long periods (linked to the risk assessment bridge, MRIS Annex F).

- Processes: a defined procedure for how a detected deviation automatically becomes a case (ticket) and leads 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 under version control like source code.
- Integration at two points: before deployment in the CI pipeline (checking before anything goes into operation) and in running operations (continuous checking of the actual state against the target specifications), ideally with enforcement (runtime enforcement).
- Target states (baselines): the target specifications to be checked are defined and maintained; machine-readable control catalogues (e.g. in NIST's OSCAL format) facilitate automated checking and evidence.
- Response: deviations are presented on real-time dashboards with clear metrics and automatically trigger a case (incident ticket).
- Further reading: 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: deliberately create a deviation from the target state in a test area (e.g. a disallowed 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 observance of its security requirements quarterly through audits. Between reviews, an incorrect permission in a cloud environment is only noticed weeks later. The company therefore formulates its most important requirements as executable rules and integrates them into both deployment and running operations. If a configuration breaches a rule, this appears immediately on a dashboard and automatically creates a case for remediation. The quarterly snapshot thus becomes continuous control.

Example B — general department. An organisation without development of its own uses mainly cloud services. It cannot program rules itself but activates the continuous configuration checking functions available in its platforms (such as a security or compliance status that runs permanently). Deviations from recommended settings are displayed there on an ongoing basis; the organisation determines who picks them up and remediates them. It thus obtains continuous checking through the built-in capabilities of its services, without developing anything itself.

6. Maturity path (cumulative)

- Initial: compliance dashboards maintained manually.
- Defined: policy as code in the CI pipeline; real-time checks against target states.
- Managed: enforcement in operations (runtime enforcement); automatically created cases on deviation.

7. Measurement and audit evidence

- Metric: continuous monitoring coverage — share of controls carried in the SoA with active automated checking (at least daily, with alerting on deviation). Indicative values: ≥ 60% after 12 months, ≥ 80% after 24 months.

- Evidence: rule repository (policy as code), overview of the controls checked automatically, records of cases triggered on deviation.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — rule set and coverage overview; responsible? — ISMS management; frequency? — continuous (at least daily); tolerance? — material controls without automated checking only on a time-limited and justified basis.

8. Typical mistakes

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

9. Delimitation, residual risk and references

- Delimitation: concerns the continuous checking of compliance with requirements; the detection of active attacks is covered by MHC-05, 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 becomes visible only 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-DVO 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 running the fast, clear containment steps automatically — people concentrate on the ambiguous and serious cases. Target state from the Defined level (semi-automated) or Managed (fully automated).
- Policy affected: incident response policy.
- 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 contain 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-DVO 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 within minutes; ambiguous or serious cases are routed deliberately to people. The automation layer itself is secured. The organisation is prepared for several simultaneous incidents.

2. Purpose and threat context

AI-assisted attacks run at machine speed: often only minutes pass between initial access and 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 in order to overload the response. The protection objective is to automate the fast, unambiguous containment steps so that they take effect within 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 people occurs and how the automation layer is protected.
- Responsibilities: assigned in the Statement of Applicability (template: MRIS Annex E); security operations is aligned to analysis and incident command rather than to mere alert review.
- Risk analysis and SoA: which actions may run automatically is determined by unambiguity and potential impact. Risk treated: response that is too slow against machine-speed and parallel attacks (linked to the risk assessment bridge, MRIS Annex F).
- Processes: regular exercises for three to five simultaneous incidents (at least half-yearly); 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 exfiltration, parallel multi-vector attack, supply chain compromise, waves of attacks against logins.
- Hardening 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; rate limiting per playbook (against floods of false alerts); human approval (human in the loop) for actions with a large impact (e.g. mass account lockout, isolation of larger network areas, certificate revocation); protection of the playbooks like source code (versioning, code review, signed releases).
- Further reading: structure and lifecycle of incident handling in NIST SP 800-61 Revision 3 (aligned to CSF 2.0); requirements in C5:2026 SIM-02/03 and NIS2-DVO No. 3.5.

- Effectiveness test: trigger an unambiguous attack signal in a test environment; the matching playbook must initiate containment automatically and within the target time — and an action with large impact may only take place after human approval.

5. Implementation examples

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

Example B — general department. An organisation without 24/7 operations of its own cannot build and maintain such automation itself. It therefore procures detection and rapid response as a service (managed detection and response) and, when selecting a provider, looks for a contractually assured response time as well as comprehensible, agreed containment measures. Internally it determines who is contactable in an emergency and which far-reaching actions are pre-approved. It thus obtains a fast, rehearsed response without operating a platform of its own.

6. Maturity path (cumulative)

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

7. Measurement and audit evidence

- Metric: time to contain an unambiguous incident (MTTC; indicative value: under 10 minutes where confidence is high).
- Evidence: playbook definitions and logs, evidence of hardening of the automation layer (trigger authentication, audit trail, approval rules), records of the multi-incident exercises or the provider SLA.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — incident response policy, playbooks; responsible? — security operations/SOC management; frequency? — continuous, exercises at least half-yearly; tolerance? — no automatic action with large impact without human approval.

8. Typical mistakes

- Automation is applied broadly without protecting the automation layer; unauthenticated triggers or floods of false alerts cause unintended containment.
- Far-reaching actions run without human approval; a false alert then locks many accounts or isolates entire 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 manipulation go unnoticed.

9. Delimitation, residual risk and references

- Delimitation: concerns the response to detected incidents; detection itself is covered by MHC-05, continuous compliance checking by MHC-10.
- Residual risk: the automation can become a target — attackers can probe trigger conditions, deliberately provoke false alerts or turn containment actions against the organisation; without the hardening described, the measure is not effective.
- Effectiveness limit: SOAR does not replace sound detection, asset context and incident governance. Automation takes effect only once triggers, data quality, criticality logic, approval rules and fallback paths are defined and tested; actions with potentially high impact 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-DVO No. 3.5; DORA Art. 17; NIST SP 800-61 Revision 3 (incident response, aligned to CSF 2.0).

Managed service variant (for organisations without SOAR of their own)

- Service to be procured: automated first response as part of the MDR contract or as managed SOAR; predefined playbooks for limited tier-1 actions (account lockout, mail quarantine, host isolation).
- Evidence to be required contractually: playbook catalogue with approval levels; complete execution logs for every automated action; exercise records at least half-yearly; rate of false triggers.
- Minimum service levels: defined execution times per action; human approval for actions with a large impact anchored contractually; rollback capability for every automated action.
- Remaining in-house effort: approval and maintenance of the playbooks, definition of the limits of effect (blast radius), decision authority in escalations.

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 merely whether they are documented — and thus exposes false-positive security assumptions.
- Policy affected: policy on security testing and independent review.
- Dependencies: no precondition; tests the effectiveness of the other controls (including MHC-05, MHC-11) under realistic conditions.
- Effort drivers: depends strongly on the format chosen (full TLPT with an external red team vs lower-cost purple team and simulation formats). The concrete effort depends on the organisation's resources.
- Primary metric: share of identified vulnerabilities remediated within the agreed 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-DVO No. 3.5.5.

1. Control statement

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

2. Purpose and threat context

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

3. Organisational implementation (CISO perspective)

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

4. Technical implementation (for IT specialists)

- Threat-led tests: external red teamers recreate real attack chains, derived from current threat analysis — tailored to the organisation, against production systems.
- Include Mythos scenarios: for example AI-assisted spear phishing with a faked voice (deepfake voice), attack chains broken into micro-steps, parallel multi-vector attacks, supply chain compromises, misuse of cloud access via compromised service accounts.
- Purple teaming between cycles: attack and defence work together and map what has been exercised to MITRE ATT&CK techniques, in order to improve detection coverage (MHC-05) in targeted fashion.
- Graded formats: full TLPT to the TIBER-EU methodology (resource-intensive, for regulated/critical entities) or lower-cost 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).
- Further reading: TLPT approach in the DORA TLPT RTS (Delegated Regulation (EU) 2025/1190) and in the ECB's TIBER-EU framework (aligned to DORA since February 2025, with mandatory purple teaming); penetration testing requirements in C5:2026 OPS-22.
- Effectiveness test: in a threat-led scenario (e.g. an attack chain broken into micro-steps), check whether the existing detection and response actually notice and stop the attack — an undetected 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 every time. New, AI-typical approaches are not tested in the process. The company therefore moves to threat-led tests: external red teamers recreate real attack chains — such as an approach broken into many small steps, or phishing with a faked voice — and test against the real systems whether detection and response take effect. Between tests, attack and defence exercise together and close identified gaps in detection. This makes visible which protective measures really work and which are merely documented.

Example B — general department. A smaller organisation cannot sustain a full, resource-intensive testing programme. It therefore chooses a lower-cost 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 recreates typical attack techniques. The results show concretely where detection needs improvement. It thus also tests 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 cycles.
- Managed: continuous attack simulation, simulation platform in production; timely remediation of findings (closure rate ≥ 90% within the agreed window).

7. Measurement and audit evidence

- Metric: closure rate — share of identified vulnerabilities remediated within the agreed window (indicative value: ≥ 90%); in addition, the frequency and realism of the tests.

- Evidence: test reports with scenarios and findings, remediation evidence, ATT&CK mapping of the purple team exercises.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — testing policy, test reports; responsible? — security management; frequency? — at least annually for critical functions, purple teaming in between; tolerance? — no open critical findings beyond the agreed window.

8. Typical mistakes

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

9. Delimitation, residual risk and references

- Delimitation: concerns the verification of effectiveness; automated testing during development is covered by MHC-09, 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 delimitation: MHC-12 requires the testing of realistic attack chains, not necessarily a DORA TLPT in the narrow regulatory sense. For DORA-regulated entities the governed DORA TLPT (RTS (EU) 2025/1190) remains binding; all others achieve the objective through 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 to DORA since February 2025); C5:2026 OPS-22; NIS2-DVO No. 3.5.5; attack simulation with, among others, 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.
- Policy affected: a new policy on the use of AI agents; additionally the access control and procurement policies.
- Dependencies: presupposes the identity basis 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 connections in use; later stages presuppose suitable tools available in your own environment. The concrete effort depends on the organisation's resources.
- Primary metric: share of production agents with a documented threat model and limited access, plus logging coverage.
- Standards touched (without claim of fulfilment): among others ISO/IEC 42001 Annex A, NIST AI RMF 1.0, OWASP threat taxonomies; depending on the field of use, also the EU AI Act.

1. Control statement

AI agents in production use (coding agents, agentic workflows) are treated like privileged systems: a limited blast radius per agent, an identity restricted to the necessary permissions, complete and traceable logging with a named human owner and a 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 production agent: a technical, capability-scoped identity instead of personal user accounts or blanket API keys; expressly 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 an approved tool scope per function; defined approval thresholds for actions with high impact. The identity basis is supplied by MHC-04 (workload identity), on which the capability-scoped limitation builds.

2. Purpose and threat context

AI agents combine the reasoning of a language model with tool execution, persistent memory and multi-step planning. This creates a privileged attack surface outside the established security measures (Mythos-Ready, CSA/SANS/OWASP, April 2026: "unmanaged AI agent attack surface", classified CRITICAL). Two risk sides: first, over-privileged or insecure agents in your own operations; second, the supply chain — compromised MCP servers, IDE extensions or skills. Evidenced by real incidents: EchoLeak (CVE-2025-32711, undetected data exfiltration via prompt injection in Microsoft 365 Copilot) and CurXecute (CVE-2025-54135/-54136, code execution via the MCP connection of the coding IDE Cursor).

3. Organisational implementation (CISO perspective)

- Policy: a policy on the use of AI agents is created. It governs the permitted purposes of use, the requirement for a named human owner per production agent, the exclusive use of approved components, and the requirement for human approval before irreversible actions.
- Responsibilities: an AI governance function is named; the CISO is accountable; the board is involved, because approving production agents touches risk acceptance decisions above the CISO's mandate (MRIS Annex E, Ch. 10.5).
- Risk analysis and SoA: before going into production, a threat and risk analysis is prepared and documented for each purpose of use; the target maturity level is derived from it (linked to the risk assessment bridge, MRIS Annex F).
- Processes: (a) an approval process before production use (documented purpose, 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 rapidly revoking compromised agents.

- Procurement and suppliers: requirements for external agentic components (provenance, update channel, security evidence) are defined; adoption only via the allowlist.
- Training: development and business teams are made aware of the risks, particularly indirect prompt injection.
- Harness as code: it is established that an agent's control components (instructions, tool definitions, rules files) are subject to the same approval and version control rules as source code; the technical implementation is set out in section 4.

4. Technical implementation (for IT specialists)

- Limited identity: every agent runs under its own workload identity (connecting to MHC-04) with fine-grained, time-limited permissions per tool call (read, write, network permissions with an explicit scope); no personal developer tokens.
- Limiting 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, spend above a threshold).
- Logging: every tool call as well as file and network access is logged as a reproducibly 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 channels as far as technically possible; input and output filtering; treatment of tool outputs as untrusted input (protection against misused tool permissions, "confused deputy"); penetration tests against prompt injection before production use.
- 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 production use.
- Further reading: 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 agent models in NIST AI 100-2e2025.
- Effectiveness test: introduce a prepared email or ticket with an embedded command (injection probe); the agent must not carry out the impermissible action demanded in it (such as external transmission), and the attempt must appear in the log.

5. Implementation examples

Example A — development/platform context. In an organisation with its own software development, developers use an AI assistant directly in their working environment. This assistant is connected to several internal systems — the ticketing system, source code management and an internal knowledge base — and operates under the developers' personal access. The risk: a hidden command can be injected into a ticket that induces 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 connections, permits only verified connections, gives each assistant its own tightly limited access instead of full developer rights, and records completely what it does and who started it. In a further stage, the assistant's control specifications are reviewed and versioned

like program code, tested regularly for manipulability, and a misused access can be revoked within minutes.

Example B — general department (no-code/SaaS). A business department without development of its own — sales, say — uses Microsoft 365 Copilot and sets up its own assistant there with built-in means (or uses a self-created GPT in ChatGPT Business). This assistant may read the mailbox, file storage and customer database and send messages, with the full rights of the employee who set it up. The risk: an incoming email or document can contain a hidden command that induces the assistant to send confidential content externally — exactly 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 platform's administrative settings: the department records which assistants exist and what they may access, permits only approved connections, limits the assistants' rights to what is necessary instead of full user rights, names a responsible person for each production assistant, and enables the platform's logging. For irreversible steps — such as external transmission or deletion of data — explicit human confirmation is required.

6. Maturity path (cumulative)

- Initial: AI coding agents documented; manual inventory of MCP servers and extensions.
- Defined: limited identities and complete logging in production; a threat model per purpose of use; allowlist for extensions and MCP servers.
- Managed: full code review process for the harness; automated checks before production use; time to revoke under 5 minutes; regular penetration 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 greater part of the risk. Penetration tests of the harness and robustness testing of the models follow in later stages, depending on the tools available in your 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: 100%); logging coverage for actions with write or network permissions (target: 100%); time to revoke compromised agent identities (target: under 5 minutes).
- Evidence: agent inventory, allowlist configuration, logs, pre-production test reports, threat model documents.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — via the inventory and threat models; responsible? — a named human per agent; frequency? — continuous logging, verification before every production release; tolerance? — no production agents with write or network permissions without logging.

Market and tooling map (selection criteria, as at July 2026)

The market for agent governance tooling is young and consolidating; naming products dates quickly. For level 2 evidence, one tool per category that meets the criteria below is sufficient:

- Agent and NHI inventory: automatic capture of non-human identities (service accounts, API keys, agents) with owner assignment and export to CMDB/SoA; criterion: coverage measurement against IdP and platform data (metric: agent inventory coverage).

- Agent identity and capability scoping: IdP-native agent identities or workload identity (SPIFFE/SPIRE, cf. MHC-04) with short-lived credentials and fine-grained permissions; criterion: revocation within minutes (mean time to revoke).
- MCP/tool server security: signing and provenance verification of tool servers and manifests, registry control, testing for tool poisoning and prompt injection vectors; criterion: refusal of unsigned tools as an enforceable policy (logic transferred from MHC-02).
- Runtime guardrails and logging: policy-driven action approvals, session recording, tamper-proof execution logs with SIEM connection (A.8.15); criterion: complete, forgery-proof traceability per agent action.

Procurement note: the categories can initially be covered with built-in means (IdP, signing infrastructure from MHC-02/-06, SIEM); a specialist product only becomes necessary when the number of agents or compliance requirements exceed what manual maintenance can support.

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 supply chain path.
- The harness (instructions, rules files) is treated as configuration rather than code — no approval, no version control.
- Human approval in name only (a confirmation dialogue without a genuine opportunity to check) — excessive latitude for action remains.
- The Managed level is set as a blanket target even though the necessary tools and procedures are not yet established in your own environment.

9. Delimitation, residual risk and references

- Delimitation: covers governance and security of self-operated agents; the identity basis is supplied by MHC-04; insecure, AI-generated code is covered in MHC-09; the inventory of unapproved AI tools (shadow AI) is set out in A.5.9 (MRIS extension).
- Residual risk: prompt injection cannot be fully solved as long as instruction and data channels are not separable; detection is not prevention; suitable hardening tools are not available in every environment.
- ISO/IEC 27002:2022: A.5.9 Inventory of information and associated assets; 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 causes excessive functionality, excessive permissions, excessive autonomy; LLM03 supply chain); OWASP Top 10 for Agentic Applications (ASI01 agent goal hijack, ASI02 tool misuse, ASI03 identity and 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 off-site, one immutable/offline, zero errors in the test).
- **AAL2 / AAL3:** authentication assurance levels under NIST SP 800-63B; AAL3 requires a device-bound, non-exportable key.
- **Admission controller:** platform control point (e.g. in Kubernetes) that permits only allowed, verified images at deployment.
- **Adversary emulation / BAS:** recreating real attack techniques, automated via breach and attack simulation platforms.
- **Air-gapped:** backup copy physically isolated from the network.
- **Baseline (target state):** defined target state against which the actual state is checked.
- **Blast radius:** area of impact — how far a successful attack can spread.
- **Capability scoping:** limiting access to exactly the permissions and capabilities needed for the task at hand.
- **Conditional access:** access rules that check user, device and context at every login before granting access.
- **Confidential computing / TEE:** protected execution environment that encrypts data in memory during processing as well (e.g. Intel TDX, AMD SEV-SNP, ARM CCA).
- **Container / image / registry:** a container is an application packaged in a standardised way; the image is its immutable template; a registry is the storage location for images.
- **Continuous control monitoring:** continuous, automated checking of compliance with requirements instead of periodic audits.
- **CVE/NVD, OSV, EUVD:** vulnerability databases (NVD: US database; OSV: open-source vulnerabilities; EUVD: ENISA's EU database).
- **CycloneDX / SPDX:** the two established SBOM formats (CycloneDX standardised as ECMA-424, SPDX as ISO/IEC 5962).
- **Deepfake voice:** artificially generated, deceptively authentic voice, e.g. for spear phishing.
- **Containment:** immediate measure that limits an ongoing incident (e.g. locking an account, isolating 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, rules 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 impact.
- **Hybrid cryptography:** combination of a classical and a quantum-safe method during the transition period.
- **Image signing (Sigstore/cosign):** cryptographic signature of an image, verified before start-up to rule out manipulation.

- **KEV (known exploited vulnerabilities):** CISA catalogue of actively exploited vulnerabilities; triggers expedited patching.
- **AI-generated code (vibe-coded):** code produced with AI coding assistants, to be checked as a risk category of its own.
- **Kill chain correlation:** linking individual events into a connected attack chain.
- **Living off the land:** approach in which attackers use built-in 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 through which an AI assistant is connected to external tools and data sources.
- **MDR (managed detection and response):** procuring detection and rapid response as a service with a contractual response time.
- **MITRE ATT&CK:** recognised catalogue of real attack techniques to which detection is aligned.
- **ML-KEM / FIPS 203:** quantum-safe method for key exchange (replacing, for example, RSA/ECDH).
- **mTLS (mutual TLS):** encrypted connection in which both sides identify themselves to each other by certificate — not just one side.
- **MTTC (mean time to containment):** average time to contain 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, version-controlled rules (e.g. OPA/Rego, Sentinel).
- **PQC (post-quantum cryptography):** encryption and signature methods intended to withstand future quantum computers.
- **Pre-merge block:** automatic block preventing code with serious 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 behave in an unwanted way.
- **Red / blue / purple team:** the attacking side, the defending side, and their joint exercise.
- **Remote attestation:** proof that a protected execution environment is genuine and unaltered before it is used.
- **Runtime enforcement:** enforcing requirements in running operations, not merely checking them.
- **SAST / DAST / SCA:** static code analysis, dynamic testing of the running application, dependency testing.
- **SBOM (software bill of materials):** machine-readable register of all components of a piece of software together with their dependencies.
- **Service mesh:** an infrastructure layer that automatically handles connectivity 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) for giving every 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), normally with a short validity.
- **Threat hunting:** targeted, hypothesis-based search for previously undetected attacks.
- **TIBER-EU:** ECB framework for threat-led testing in the financial sector; aligned to DORA since February 2025.
- **TLPT (threat-led penetration testing):** threat-led, realistic testing that recreates real attack scenarios.
- **Transitive dependency:** an indirectly included building block (a library that itself uses further libraries).
- **UEBA:** behaviour-based detection against 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 network membership; a replacement for the classical VPN.

Effort classes per MHC (indicative values)

The following classes give an order of magnitude for the internal implementation effort per maturity level: S = up to approx. 10 person-days, mainly with built-in means; M = approx. 10–40 person-days, possibly with a tool licence; L = over 40 person-days or a platform or service investment. Licence and service costs are not included; the values are to be adapted organisation-specifically.

MHC	Initial	Defined	Managed
MHC-01	S	M	M
MHC-02	S	M	L
MHC-03	S	M	M
MHC-04	M	L	L
MHC-05	M	L	L
MHC-06	S	M	L
MHC-07	M	L	L
MHC-08	S	M	M
MHC-09	S	M	L
MHC-10	S	M	M
MHC-11	M	L	L
MHC-12	M	M	L

MHC	Initial	Defined	Managed
MHC-13	S	M	L

For MHC-05, MHC-09 and MHC-11, the managed service variant typically reduces in-house effort by one class; the effort shifts to contract and evidence management.

Model audit programme for auditors (MHC add-on module)

This module makes the MHC auditable within existing ISO/IEC 27001 audits — as an additional check in stage 2 or in internal audits under Clause 9.2; it does not establish a certification scheme of its own. Procedure per applicable MHC: pose the audit question, inspect the minimum evidence for the Defined level, take a sample, document the assessment against the SoA rationale. MHC that are not applicable require a documented rationale in the SoA.

MHC	Core audit question	Minimum evidence (Defined level)	Primary ISO reference
MHC-01	Crypto inventory complete and prioritised for migration?	Inventory extract, migration strategy, review record	A.8.24
MHC-02	SBOM per own artefact, query capability evidenced?	SBOM artefacts, pipeline configuration, correlation evidence for a vulnerability advisory	A.5.21, A.8.30
MHC-03	Phishing-resistant MFA enforced for privileged and external accounts?	IdP report, exception list with expiry dates	A.5.17, A.8.5
MHC-04	Workload identity and segmentation effective?	Stage gate records, mTLS coverage, architecture evidence	A.8.20–A.8.22
MHC-05	ATT&CK coverage measured and confirmed by attack testing?	Coverage overview, test results, hunting records	A.8.15, A.8.16
MHC-06	Signature and admission policies enforced in production?	Admission logs, policy configuration, container policy conformity metric	A.8.31, A.5.23
MHC-07	Tenant separation demonstrably tested?	Separation test report per level	A.5.23, A.8.22
MHC-08	Immutability configured and recovery rehearsed?	WORM/immutability configuration, restore records	A.8.13, A.8.14
MHC-09	AI-assisted testing in the pipeline, KEV SLA met?	Pipeline runs, KEV latency analysis, re-test evidence	A.8.29, A.8.8
MHC-10	Continuous control monitoring active, deviations alerted?	Monitoring coverage, drift alerts, remediation evidence	A.8.9, A.5.36
MHC-11	Tier-1 automation with approval levels and rollback capability?	Playbook catalogue, execution logs, MTTC analysis	A.5.26
MHC-12	TLPT with Mythos scenarios carried out, findings closed?	TLPT report, findings closure analysis	A.5.35, A.8.29
MHC-13	Agents inventoried, execution logged tamper-proof?	Agent inventory, execution logs, harness review evidence	A.5.9, A.8.15

Internal audits under Clause 9.2 can adopt the table unchanged as an audit programme; in certification audits it serves as a proposed list for the auditor. All evidence terms correspond to the "Measurement and audit evidence" sections of the respective MHC building blocks.

Mapping Implementation Guide ↔ MRIS 1.8, 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: share of phishing-resistant MFA.
- **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); metric: container policy conformity.
- **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 (results 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-assisted 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 testing) 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 (including mean time to revoke); metric: agent inventory coverage.

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

Version history

Version	Date	Changes
1.2	June 2026	Base edition of this guide (thirteen MHC building blocks, prioritisation, glossary, mapping).
1.3	July 2026	References to MRIS Annexes E/G corrected; evidence logic instead of "verification logic to ISO/IEC 27005"; managed service variant for MHC-05/-09/-11; effort class table; ISO anchors aligned to MRIS Annex A; reconciliation with MRIS 1.7; licence changed to CC BY-SA 4.0.
1.4	July 2026	Reconciliation with MRIS 1.8 (multi-vendor); sensitivity verification step for the Threat Priority Score; market situation note in the tiering; stage gate indicative values for MHC-04/-07; market and tooling map for MHC-13; model audit programme for auditors; "within minutes" statements given a precondition context.