MRIS Implementation Guide v2.0

Implementation guideline for the thirteen Mythos Hardening Controls

Companion document to MRIS, Chapter 9 — Mythos-Resistant Information Security

ISO 27002 depth · Evidence-based · Practice-oriented

Version 2.0 | August 2026

Author: Richard Peddi (produced with AI assistance)

AUDIENCE

CISOs, ISMS owners and security architects who want to implement the thirteen "Mythos Hardening Controls" from MRIS, Chapter 9 in practice.

Purpose, independence and conventions

This document is the implementation layer for MRIS, Chapter 9. In Chapter 9, MRIS 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 — in such a way that the ISMS ownership function can follow purpose, organizational and technical implementation, maturity and evidence without additional specialist literature. It does not replace MRIS but puts its catalogue into practice. The mapping is made through the stable MHC identifiers (MHC-01 to MHC-13); this guide is therefore valid independently of the respective MRIS version number. The source references in this guide are brought up to the verified status of August 2026.

Fixed structure per MHC

- Header "At a glance"
- 1. Control statement
- 2. Purpose and threat relevance
- 3. Organizational implementation (leadership)
- 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 organizations without their own security operations.

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

Organizations with elevated audit, regulatory or customer requirement levels may extend the evidence voluntarily. For that purpose, a supplementary audit grid can be used per applicable MHC:

- Control owner: the responsible role or organizational 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 the effectiveness check
- 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 check

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

License and disclaimer

Disclaimer

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

Note on how this document was produced

This document was produced with the support of AI tools. The selection and evaluation of sources, the technical classifications and the responsibility for the content rest with the author.

Recommended citation

Peddi, Richard (2026): MRIS IMPLEMENTATION GUIDE, Version 2.0. mris.info

Contents

Prioritizing 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 prioritization every control appears equally urgent — and that is precisely what makes the implementation decision harder. What is prioritized is not the control as such but its contribution to reducing the threats most relevant to the organization. This chapter combines two complementary views for that purpose: the measurable coverage breadth of each MHC (how many degraded controls it supports) and the threat- and effort-related sequence (threat correlation). The prioritization remains compatible with the overarching MRIS principle: structural controls before new detection or automation capabilities.

At a glance

- Purpose: a traceable, threat-oriented implementation roadmap instead of a linear MHC sequence.
- Basis: 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 supports

The following overview shows how many of the controls degraded under Mythos (categories "partially degraded" and "friction-only", 41 controls in total) each MHC flanks. The figures are verified against the assessment matrix in MRIS, Annex A (overall distribution 29 robust / 37 partially degraded / 4 friction-only / 23 not affected). Because 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	broadly 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	Degraded controls	Also flanks robust	Character
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 supports. High coverage means high leverage across many risks at once.
- Threat and effort relevance: the sequence — dependent on the 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 organization with AI agents (it addresses the core of the Mythos threat landscape), and 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" then means controls with high leverage, a large control gap and low effort, having regard to the dependencies.

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 being prioritized because of a high threat level, 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 organization?	1 = low … 5 = very high
Business impact	How severe would the consequences be for DORA/BIA-critical services, customer data, availability, reputation or finances?	1 = low … 5 = existential
Exposure	How strongly is the organization exposed (internet-facing, cloud, suppliers, admin access, external users)?	1 = low … 5 = strongly exposed
Control gap	How large is the gap relative to the MHC target state?	1 = well covered … 5 = barely covered

Example: AI-supported phishing with threat relevance 5, business impact 4, exposure 5 and control gap 4 gives 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 prioritization index, not a substitute for a formal risk analysis (ISO/IEC 27005) or an organization-specific risk matrix. Multiplying the factors makes relative action priorities visible; the value is ordinal, not metric — a score of 400 is not "twice as urgent" as 200 but serves ranking and triage.

4. Threat-to-MHC matrix

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

Threat / attack scenario	Primary MHC	Rationale	Priority
AI-supported phishing / credential theft	MHC-03, MHC-05, MHC-11	MHC-03 prevents credential abuse, MHC-05 detects anomalous 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 and purple teaming test 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 and multi-tenant platforms; separation must be technically enforced and tested.	high, where applicable
AI agent abuse / prompt injection / tool abuse	MHC-13, MHC-04, MHC-10, MHC-09	Agents need their own identities, limited tool rights, policy control and a secure pipeline.	high, where AI agents are deployed
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 meaningfully verifiable, through other MHC.

Dependency	Meaning
MHC-02 → MHC-09	AI-supported security testing in the pipeline needs a build and 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 robust where detection is based on reliable, central and tamper-resistant 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 is stronger where image provenance and workload identities are cleanly controlled.

Dependency	Meaning
MHC-04 + MHC-06 → MHC-07	Multi-tenancy isolation draws on identity, network, resource and runtime separation.

6. Standard prioritization 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 formally sit 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; it is to be adjusted per organization to exposure, architecture, asset criticality, existing controls, resource position 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 robust 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	The basis for supply chain transparency and pipeline testing.
P1 — next	MHC-09	AI-supported 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	A cross-cutting control for ongoing deviation detection.
P2 — risk-based	MHC-06	Container security and confidential computing	Highly relevant for container and 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	Prioritize higher for long-lived confidential data.
P2 — risk-based	MHC-12	Threat-led penetration testing	Validation after baseline implementation, then on a regular basis.

Note on the market situation (July 2026): the multi-vendor development and the disclosure waves now beginning (MRIS Chapter 2.5) raise the threat-level 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 account of sector-specific threats (financial services, critical services, supply chain, cloud, AI); assess threats by currency, likelihood of occurrence and proximity to the organization.
- Step 2 — map threats to MHC: determine for each threat which MHC act preventively, detectively, reactively or in a validating role; weight controls with several threat relationships more heavily; document 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 account of effort and dependencies; where risk is equal, take quick wins first — those with 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 prioritization of the Mythos Hardening Controls, the MHC are correlated with relevant threat scenarios. For each threat it is assessed which controls act preventively, detectively, reactively or in a validating role. The prioritization follows from threat relevance, business impact, exposure, the existing control gap, dependencies and implementation effort. The result is 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.

9. Reading the maturity levels

The **Initial** level documents the implementation path begun and does not necessarily represent full fulfilment of the respective MHC control statement. Only from **Defined** onwards is an MHC deemed implemented in principle; **Managed** describes the advanced or continuously effective implementation. The control statements of the modules describe the minimum requirements of the Defined level; requirements going beyond that are expressly marked in the statement as advanced hardening (Managed level). For reporting this means: an MHC at the Initial level is recorded as "in implementation", not as implemented. Those wishing to connect the levels to a common maturity model can read them as follows: not implemented – Initial (partial) – Defined (implemented) – Managed (governed and optimized).

10. Sensitivity of the threat priority score

The score is a qualitative ordinal index without factor weights: the four factors are multiplied with equal standing. Because ordinal values are multiplied, the result is a ranking and not a metric quantity — decimal places or percentage comparisons would be false precision. Check step per organization: the rating of each factor is varied individually by one rating step, so far as that is substantively plausible; if the priority class changes as a result, the classification must be specifically justified (documented in the risk process under ISO/IEC 27005).

For the standard tiering of this guide the following applies: the four P0 controls each rest on at least two dominant factors (threat level and impact, or an enabler role) as well as on the dependency rules from Section 5. If the priority class of an MHC changes on variation of a single factor by one rating step, the classification is to be marked as sensitive and justified separately. Without a documented organization-specific sensitivity calculation, no statement is made here about the stability of the P0 set. What can be expected to react are mainly the ranks within P1 and P2 (typical candidates for change: MHC-06 and MHC-12 depending on exposure). Organization-specific deviations remain permissible and are to 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 exchange cryptographic methods swiftly when required.
- 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 exchangeability of the libraries used. The actual effort depends on the organization'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 (no 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 organization 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 relevance

Future quantum computers will be able to break encryption that is widespread today (such as RSA and elliptic curves). Attackers are therefore already intercepting encrypted data now in order to decrypt it later — the "harvest now, decrypt later" pattern. Data that must remain confidential over many years is particularly affected. AI additionally accelerates the discovery and exploitation of weak or outdated crypto implementations. The harvesting pattern concerns not only encrypted data but also signatures valid today: certificates, code signatures and long-lived signed documents can be harvested and later retroactively forged or repudiated, once a sufficiently powerful quantum computer becomes available. During migration there is a further risk that a hybrid connection is forced back by an attacker onto its purely classical component and thereby loses its quantum safety. The protective objective is to migrate long-lived confidential data to quantum-safe methods in good time and to create the ability to change methods swiftly when required.

3. Organizational implementation (leadership perspective)

- Policy: the cryptography policy is extended by an obligation to maintain a central crypto inventory and by a migration strategy towards quantum-safe methods.
- Responsibilities: a function is named for maintaining the inventory and updating the strategy, and assigned in the Statement of Applicability (template MRIS Annex E).
- Risk analysis and SoA: the migration order is derived from the risk assessment — data with a long confidentiality period first. Risk treated: later decryption of data exfiltrated today (linked to the risk assessment bridge, MRIS Annex F).
- Migration order within the key hierarchy: migration begins at the root (master or key encryption key, root certification authority) and works towards the derived keys; signed artefacts with long validity also belong in the inventory and the migration scope.
- Processes: regular (annual or event-driven) review of 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 exchangeable methods is included as a requirement.

4. Technical implementation (for IT specialists)

- Crypto inventory: recording of all algorithms, key lengths, protocols, certificates and underlying libraries in use; prioritization by protection requirement and confidentiality period.
- Quantum-safe methods: migration to the finalized NIST standards — FIPS 203 (ML-KEM) for key exchange and FIPS 204 (ML-DSA) and FIPS 205 (SLH-DSA) for signatures. HQC is in standardization 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 (for example in the TLS key exchange), so that the connection remains secure as long as at least one of the two methods holds.
- Downgrade protection: in hybrid negotiation, quantum-safe negotiation is enforced, a fallback to purely classical methods is rejected, and every downgrade attempt is logged and alerted.
- Migration sequence: the root of the key hierarchy is migrated first (master or KEK keys, root CA); PQC at derived levels only, with a classical root, offers no protection, because the classical trust anchor remains attackable — particularly critical for archived data with a very long protection requirement.
- Quantum-safe timestamps: for long-lived signatures, both the signature certificate and the timestamp certificate (TSA) are migrated to quantum-safe methods; otherwise a classical timestamping service weakens the evidence chain irrespective of the signature scheme. Long-term archive formats allow periodic renewal of timestamps.
- Crypto agility: encryption is encapsulated so that methods can be exchanged without deep intervention in the 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 specifications in FIPS 203/204/205.
- Effectiveness test: for one prioritized system, verify that the method actually used matches the inventory and the target specification (for example the negotiated key exchange in the TLS handshake) and that a method can be switched without a code change.

5. Implementation examples

Example A — development and platform context. A software vendor initially does not know precisely which encryption is used in its services and libraries at all. It therefore first creates a central register of all methods, keys and certificates in use 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. That keeps this data protected even if the classical method is later broken by quantum computers; at the same time the method can be changed centrally when required.

Example B — general department. An organization without its own development work 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 personnel or contract records) and asks its providers for their roadmap to quantum-safe migration. Into new contracts it includes support for quantum-safe methods as a requirement. In that way it manages 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 schedule, pilot and production evidence for hybrid methods.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — inventory and strategy; responsible? — the 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 available — the most important points remain unknown.
- Hybrid methods are introduced across the board without prioritizing by confidentiality period — much effort without a recognizable focus of benefit.
- Encryption is hard-wired into the code, so that a later change of method requires intervention in the application each time.
- Migration is treated purely as a topic for the future, even though today's data outflow already determines tomorrow's risk.
- PQC is introduced at derived levels while the root of the key hierarchy (master key, root CA) remains classical — the unprotected trust anchor renders the measures below it ineffective.

9. Delimitation, residual risk and references

- Delimitation: concerns the cryptography in use, including the PQC-specific migration order within the key hierarchy and the downgrade resistance of hybrid negotiation; routine key management (generation, rotation, storage) 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 later be decrypted; quantum-safe methods are moreover 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.
- Further reading on key management: Cloud Security Alliance, Cloud Key Management Working Group – Post-Quantum Cryptography Key Management (draft, August 2026).

MHC-02 — SBOM and build provenance

At a glance

- Operational benefit: makes immediately visible which third-party components are contained in the organization's own software, so that when a new vulnerability appears it is clear in minutes rather than days whether and where the organization is 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 and CI pipeline to realize the full benefit.
- Effort drivers: depends on the number and structure of the build pipelines and the depth of the supply chain. The actual effort depends on the organization's resources.
- Primary metric: share of delivered artefacts with a current, automatically generated bill of materials (SBOM coverage).
- Standards touched (no claim of fulfilment): ISO/IEC 27002 A.5.21/A.8.30 primarily, supplemented by A.5.19, A.5.20, A.5.22 (supplier and service management) and A.8.4 (access to source code); for the complete mapping see MRIS Annex A. CRA Annex I; BSI TR-03183-2; C5:2026 DEV-13; SLSA.

1. Control statement

For every piece of software developed in-house and distributed onwards, a bill of materials of the contained components (SBOM) is generated automatically and reconciled against vulnerability databases. Advanced hardening (Managed level): for critical artefacts it is additionally documented in a traceable way how and from what they were built (build provenance), and the provenance attestations are verified in automated form.

2. Purpose and threat relevance

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

3. Organizational implementation (leadership perspective)

- Policy: the development and supplier policy prescribes automatic SBOM generation for the organization's own and onward-distributed artefacts, together with a contractual SBOM requirement on 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 prioritized on the basis of the risk assessment. Risk treated: undetected vulnerabilities or manipulation in third-party components (linked to the risk assessment bridge, MRIS Annex F).
- Processes: a fixed procedure for determining and remediating exposure on the basis of the bills of materials when a new vulnerability becomes known; retention of the bills of materials per software version.
- Procurement: SBOM and, where possible, provenance attestations are included as requirements in contracts with critical suppliers.

4. Technical implementation (for IT specialists)

- SBOM generation: automatically in the build and CI pipeline, in an established, machine-readable format — CycloneDX (standardized as ECMA-424) or SPDX (standardized as ISO/IEC 5962). Indirect (transitive) dependencies are recorded as well, not only 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 the components against vulnerability databases (CVE/NVD, OSV, ENISA's EUVD). Vulnerability data do not belong in the SBOM itself but are communicated separately via VEX or CSAF.
- Build provenance: for critical artefacts, it is recorded in a traceable and tamper-resistant way from which sources and with which build process they were produced (SLSA build provenance, a higher level for critical artefacts).
- Tools (non-binding examples): generation with Syft, cdxgen or Trivy; central management and vulnerability correlation for example with 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 (for example a Log4j-type flaw), determine solely from the bills of materials whether and which of the organization's own artefacts contain the affected component — the result must be available within minutes.

5. Implementation examples

Example A — development and platform context. A software vendor learns from the news of a serious vulnerability in a widely used library. Until now it has had to search all projects laboriously one by one, which takes days. It therefore extends its build pipeline so that every build automatically produces a complete bill of materials of all contained 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 records tamper-resistant evidence of what they were built from and how.

Example B — general department. An organization without its own development work 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 then follow up specifically with the manufacturer as to whether the product in use is affected, instead of waiting for vague collective notifications. In that way it 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 of the organization's own artefacts; automated vulnerability reconciliation.
- Managed: SBOM from critical suppliers as well; build provenance (SLSA) for critical artefacts; automated verification of the provenance attestations.

7. Measurement and audit evidence

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

8. Typical mistakes

- Bills of materials are generated but never evaluated — in a vulnerability event an SBOM exists, but nobody reconciles it.
- Only the top-level dependencies are recorded; precisely the deeply nested building blocks in which vulnerabilities reside are missing.
- The SBOM is produced once and not updated per version, so that it does not match the state actually delivered.
- Vulnerability data are mixed into the SBOM; that 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; actually closing the vulnerabilities is part of vulnerability and patch management, and automated testing is dealt with by MHC-09.
- Residual risk: a bill of materials detects known but not unknown vulnerabilities; correctly declared but manipulated components are not exposed by the SBOM alone — provenance serves that purpose.
- ISO/IEC 27002:2022: A.5.19 Information security in supplier relationships; A.5.20 Addressing information security within supplier agreements; A.5.21 Managing information security in the ICT supply chain; A.8.4 Access to source code; A.8.30 Outsourced development.
- Framework: C5:2026 DEV-13; CRA Annex I (SBOM obligation for manufacturers of products with digital elements; vulnerability notification obligations from 2026, main obligations for products placed on the market after 11 December 2027); BSI TR-03183-2; DORA Art. 28; NIS2 IR No. 5; SLSA framework; formats CycloneDX (ECMA-424) and SPDX (ISO/IEC 5962).

MHC-03 — Phishing-resistant multi-factor authentication

At a glance

- Operational benefit: renders intercepted or captured credentials worthless on spoofed pages and thereby removes the basis for AI-supported, deceptively convincing 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 actual effort depends on the organization's resources.
- Primary metric: phishing-resistant account coverage — the number of in-scope accounts secured with phishing-resistant methods divided by the number of all in-scope accounts. Target for privileged accounts: 100% or a documented, time-limited and risk-approved exception. The share of logins using phishing-resistant methods is collected only as a supplementary usage metric, because individual heavily used accounts would otherwise distort the picture.
- Standards touched (no claim of fulfilment): ISO/IEC 27002 A.5.17/A.8.5; NIST SP 800-63B Revision 4; FIDO2/WebAuthn; C5:2026 IAM-08; NIS2 IR No. 11.6 and No. 11.7, for privileged accounts additionally No. 11.3.2(a).

1. Control statement

Privileged access and externally reachable services are secured with phishing-resistant methods (FIDO2/WebAuthn, passkeys or hardware tokens). Terminological clarification: a device-bound passkey satisfies the multi-factor criterion only together with local user verification (PIN or biometrics); without it, only a possession factor is present. What matters for protection against AI-supported phishing is origin binding, not the number of factors. SMS and simple confirmation methods are excluded for new access. Advanced hardening (Managed level): extension to access to internal systems classified as particularly worthy of protection, up to organization-wide passwordless sign-in.

2. Purpose and threat relevance

AI has sharply increased both the quality and the volume of phishing: spoofed sign-in pages, deceptively convincing emails and codes relayed in real time bypass classic two-factor methods such as SMS or app confirmation. Phishing-resistant methods bind the sign-in proof cryptographically to the genuine address of the service; on a spoofed page the proof is therefore worthless. The protective objective is that intercepted or captured credentials are of no use to the attacker.

3. Organizational implementation (leadership perspective)

- Policy: the authentication policy prescribes phishing-resistant methods for privileged and externally reachable access — extension to internal systems classified as particularly worthy of protection is carried as advanced hardening (Managed level); 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: issuing and withdrawing tokens or passkeys as a governed process; fallback procedures for loss that do not undercut the protection level; decommissioning of the legacy methods after the transition.
- Training: users are supported during set-up and use of the new methods, to secure acceptance and a smooth switch.

4. Technical implementation (for IT specialists)

- Phishing-resistant methods: FIDO2/WebAuthn with passkeys or hardware security keys; the cryptographic binding to the service's domain (origin binding) prevents relaying to spoofed pages.
- Assurance levels under NIST SP 800-63B Revision 4: at level AAL2, synchronizable passkeys (across several devices) are also permitted; for the highest level AAL3, a device-bound, non-exportable key is required (hardware token or smartcard), and synchronizable 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 through the central identity platform and conditional access; applications are connected via the identity platform rather than through sign-in screens of their own.
- Authenticator attestation: for privileged and particularly sensitive access, permit only attested, device-bound authenticator models; the identity platform verifies the attestation and rejects synchronizable passkeys and software authenticators (passkey vaults), because their private key is exportable.
- Relying-party-side checks (the organization's own services as relying party): for self-operated sign-in services, the core WebAuthn checks are enforced server side — a challenge that is single-use and bound to the session, origin and RP ID verification, and evaluation of the signature counter for clone detection. These checks rest with the service provider, not with the authenticator.
- Further reading: requirements per assurance level (AAL2/AAL3) in NIST SP 800-63B Revision 4; procedural detail in the FIDO2/WebAuthn specifications (FIDO Alliance, W3C WebAuthn).
- Effectiveness test: a controlled phishing attempt with a replicated sign-in page is carried out; the phishing-resistant sign-in proof must not be usable on the spoofed page (the sign-in fails).

5. Implementation examples

Example A — development and 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 the password and the confirmation in real time. The company first migrates all privileged and externally reachable access to hardware security keys whose proof is cryptographically bound to the genuine address of the service. On a spoofed sign-in page that proof is useless. In a further build-out stage, sign-in becomes passwordless organization-wide and the old SMS and app methods are switched off.

Example B — general department. An organization without its own development work uses Microsoft 365 and a few further cloud services. Staff have so far signed in with a password and an SMS code — vulnerable to phishing. The organization activates passkey sign-in in its identity platform and issues hardware security keys for roles that are particularly worthy of protection. Sign-in in future takes place without a password and without SMS; a central access rule expressly requires a phishing-resistant method for sensitive applications. In that way the protection can be enforced through the settings of the platform in use, even without development work of its own.

6. Maturity path (cumulative)

- Initial: FIDO2/passkeys for privileged access.
- Defined: phishing-resistant methods mandatory for all privileged access and all externally reachable services; password-only authentication without an additional factor excluded organization-wide.
- Managed: passwordless sign-in organization-wide; SMS methods switched off.

7. Measurement and audit evidence

- Metric: phishing-resistant account coverage — share of in-scope accounts with phishing-resistant protection, reported separately for privileged and externally reachable accounts (target: privileged 100% or a documented, time-limited and risk-approved exception; externally reachable 95% or above). The share of logins using phishing-resistant methods is collected only as a supplementary usage metric.
- Evidence: identity platform configuration, list of permitted methods per access class, evidence of the decommissioning of SMS methods.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — authentication policy and platform configuration; responsible? — identity and IT lead; frequency? — continuous; tolerance? — no privileged or externally reachable access with SMS or simple push confirmation only.

8. Typical mistakes

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

9. Delimitation, residual risk and references

- Delimitation: concerns the sign-in proof of users; the identity of services and workloads is covered by MHC-04.
- Residual risk: attacks may shift to the recovery and fallback procedures; session takeover after successful sign-in also remains a topic of its own. A compromised endpoint can moreover — even with a device-bound passkey — produce valid authentication proofs (abuse of the local authenticator interface after sign-in); that path is addressed not by MHC-03 but by endpoint hardening (A.8.7) and behaviour-based detection (MHC-05).
- Effectiveness limit: phishing-resistant does not mean phishing-proof. FIDO2/passkeys devalue classic credential phishing and MFA fatigue, but not session or token theft by malware, adversary-in-the-middle after sign-in, 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 IR No. 11.6 (authentication) and No. 11.7 (multi-factor authentication), for privileged accounts additionally No. 11.3.2(a).

MHC-04 — Workload identity and zero trust network architecture

At a glance

- Operational benefit: structurally reduces credential-based lateral movement after an initial compromise and limits the reachable blast radius; a service that has been taken over can no longer move on to neighbouring services through reused credentials. Within its own entitlements a compromised workload can still act — hence the combination with entitlement minimization and detection (MHC-05).
- 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 actual effort depends on the resources available to the organization.
- Primary metric: share of production service-to-service connections with workload identity and mTLS.
- Standards touched (no 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 IR No. 6.7/8/11.

1. Control statement

Every production workload (service, process, container, AI agent) receives a unique, cryptographically verifiable identity with short-lived, automatically rotating credentials. The identity itself is stable and uniquely attributable; short-lived and non-reusable are the credentials derived from it (for example SVIDs, certificates, tokens). Communication between services is authorized on the basis of this identity, not on the basis of network position (IP address, subnet, perimeter). Advanced hardening (Managed level): extension to development and test environments and identity-based micro-segmentation.

2. Purpose and threat relevance

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

3. Organizational implementation (leadership perspective)

- Policy: the access control and network security policy is extended by 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 lead 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 data worthy of protection first. Risk treated: lateral movement after an initial compromise (linked to the risk assessment bridge, MRIS Annex F).
- Processes: a documented process is needed for issuing, regularly renewing and withdrawing service identities; the time to withdrawal of an identity is tracked 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 per node. The workload is attested before issuance via platform selectors (for example 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 (for example Istio, Linkerd): a sidecar proxy per service handles mTLS, identity verification and policy enforcement transparently to the application code. Authorization is via identity policies (which service identity may call which service) rather than via IP allow lists.
- Segmentation: identity-based authorization 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 classic 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. Order: the most critical service-to-service paths first.
- AI agents: time-limited identities scoped to the necessary capabilities (capability-scoped) per tool call, instead of developer credentials.
- Further reading: architecture model (identity tier versus network tier, ingress and egress gateways, multi-cloud) in NIST SP 800-207A, Sections 3–4; zero trust core principles in NIST SP 800-207, Section 2; SPIFFE specification (SPIFFE ID, X.509 SVID, JWT SVID) at spiffe.io.
- Effectiveness test: a connection to a protected service is attempted from a service or host without a valid identity — it must be rejected. In addition: an expired credential must no longer permit a connection.

5. Implementation examples

Example A — development and platform context. A mid-sized software vendor runs its application from many individual, interconnected services. Until now those services trusted each other purely because they ran in the same internal network; if an attacker took over one service, they reached all the others unimpeded from there. The company changes that basis: each service receives its own, constantly changing technical credential, and without a valid credential no other service accepts a connection. It begins with the few services that process customer data; later the migration is extended to all 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 organization without its own software development runs a few internal applications, such as a file server, and has so far granted access through the internal network and a VPN: whoever is on the network counts as trustworthy. That carries the same underlying problem — a compromised device on the network gains far-reaching access. In future the organization binds access to internal applications to the verified identity of user and device rather than to mere membership of the network; an upstream access service checks sign-in, device and situation on every access. It begins with the applications most worthy of protection and with administrative access. *Applicability note:* an organization that uses only off-the-shelf cloud services and operates no infrastructure of its own is barely affected by MHC-04; the control is recorded as not applicable in the Statement of Applicability. 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 and for all production AI agents and agentic workflows falling under MHC-13; 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 at Defined: 100% in production); time to withdrawal of a workload identity.
- Stage gate (guide value): Initial — SPIFFE/SPIRE identities for 80% or more of privileged service workloads; Defined — mTLS mandatory for east-west traffic: 100% of in-scope connections or documented, time-limited and risk-approved exceptions (95% or above only as a transition threshold, not as full fulfilment of Defined); Managed — attestation-supported access policies in production. Gate fulfilment recorded as a yes/no attestation in project controlling (MRIS Chapter 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? — the platform team; frequency? — continuous (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 too long a validity (days instead of minutes) — the advantage of the short lifetime is lost.
- AI agents continue to operate under personal developer accounts — traceability and entitlement limitation are missing.
- IP-based allow-listing 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; the authentication of end users is covered by MHC-03; the governance of AI agents is developed further in MHC-13.
- Residual risk: a compromised issuance infrastructure (SPIRE) or a legitimately credentialed but compromised service remain effective attack paths; an identity does not replace entitlement minimization (least privilege remains necessary).
- Effectiveness limit: zero trust is an architectural path, not a product. Segmentation or workload identity implemented only in part can create false confidence; effectiveness arises only once authentication, authorization 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 Networks security; A.8.21 Security of network services; A.8.22 Segregation of networks.
- Framework: NIST SP 800-207 (zero trust principles); NIST SP 800-207A (cloud-native zero trust, workload identity, service mesh); SPIFFE/SPIRE (CNCF).

MHC-05 — Behaviour-based detection and kill chain correlation

At a glance

- Operational benefit: recognizes attackers by their behaviour and by the chain of connected steps rather than only by known signatures — and thereby also catches disguised attacks conducted with legitimate means.
- Policy affected: security monitoring policy (logging, monitoring, detection).
- Dependencies: presupposes central, tamper-resistant logging (A.8.15); supplies the signals for the automation in MHC-11.
- Effort drivers: depends on the range of log sources, the existing SIEM or detection platform and the available analysis capacity. The actual effort depends on the organization's resources.
- Primary metric: coverage of the ATT&CK techniques relevant to the documented organization-specific threat profile (share with effective detection validated by testing).

- Standards touched (no claim of fulfilment): ISO/IEC 27002 A.8.16/A.5.7 primarily, supplemented by A.8.15 (logging as the data basis), A.5.3, A.5.14, A.5.15, A.8.1, A.8.3, A.8.4, A.8.7, A.8.12; for the complete mapping see MRIS Annex A. MITRE ATT&CK; C5:2026 OPS-13; NIS2 IR No. 3.2; DORA Art. 10.

1. Control statement

Attacks are detected on the basis of behavioural anomalies and the linkage of connected events, not solely on the basis of individual known signatures. Detection is oriented to a recognized 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 relevance

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

3. Organizational implementation (leadership 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 conducted with legitimate means that remain below the threshold of individual alerts.
- Processes: a documented threat hunting methodology (for example 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; orientation 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 v19.2) rather than to isolated individual signatures; coverage is measured (for example with DeTT&CT) and confirmed through controlled attack tests (for example Atomic Red Team). Guide value: the techniques relevant to the documented organization-specific threat profile 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 (for example anomalous scripting or task scheduling activity, unusual system processes) and disguised command channels (for example unusual DNS or encrypted traffic with long sleep intervals).
- Identity and sign-in anomalies: abuse of phishing-resistant methods produces signals of its own — WebAuthn calls by programs that are not a browser; Windows Hello sign-ins without a device identifier; unexpected registration of new sign-in keys; creation of passkeys via administrative interfaces (for example Graph or Okta) rather than by the user.
- Robustness of the organization's own detection AI: where detection itself rests 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 consisting of individually unremarkable steps (for example via Atomic Red Team) is carried out; the chain must be recognized as a connected incident, not merely as individual, unlinked events.

5. Implementation examples

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

Example B — general department. An organization without its own 24/7 operations cannot build such detection itself. It therefore procures security monitoring as a service (managed detection and response) and takes care in the selection that the provider detects on a behaviour and technique basis, evaluates connected attack chains and contractually commits to a fixed response time. Internally it names a point of contact who takes up the reported incidents and steers remediation. In that way it 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 techniques relevant to the documented organization-specific threat profile reliably covered; threat hunting documented.
- Managed: learning detection robust against deception; continuous 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 through attack tests); supplementary, the number and outcome of the threat hunts carried out.
- Evidence: coverage overview (for example DeTT&CT), attack test results, documented hunting runs with technique mapping.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — monitoring policy, coverage overview, hunting logs; responsible? — detection or SOC lead; frequency? — continuous operation, regular hunts and coverage review; tolerance? — no important techniques left permanently unmonitored.

8. Typical mistakes

- Detection remains signature-based; attacks using built-in tools trigger no alert.
- ATT&CK coverage is asserted but never verified through controlled attack tests — detection does not take effect 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 want of capacity, is in effect not carried out.

9. Delimitation, residual risk and references

- Delimitation: concerns the detection of attacks; the subsequent automated response is dealt with by MHC-11, and automated testing of the organization's own defences by MHC-09.
- Residual risk: detection is not prevention — very slow, heavily disguised or novel attacks may remain undetected; where detection rests 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 IR 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 organizations without their own SOC)

- 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 techniques relevant to the documented organization-specific threat profile); regular MTTD and MTTC reporting; hunting reports with hypothesis and outcome; results of controlled attack tests.
- Minimum service level: continuous alert handling (24/7); a contractually defined containment time for high-confidence alerts; named escalation paths with response times; a quarterly service review with evidence of coverage.
- Work remaining in-house: connecting the log and telemetry sources, maintaining asset context, approval decisions; responsibility in the Statement of Applicability remains with the organization.

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 or 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 actual effort depends on the organization's resources.
- Primary metric: share of containers running in production from signed images verified before start-up; supplementary, the coverage of confidential computing for the workloads classified as particularly requiring protection.
- Standards touched (no 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, drawn from controlled sources, with enforcement through the platform. Advanced hardening (Managed level): for the workloads classified as particularly requiring protection, protected execution environments (confidential computing) with proof of authenticity (remote attestation) are deployed; the classification is documented.

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

2. Purpose and threat relevance

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 are encrypted in transit and at rest but are in principle visible in memory during processing — an attacker with access to the infrastructure could

capture them there. The protective objective is to execute only unaltered, verified containers and to protect particularly sensitive data during processing as well.

3. Organizational implementation (leadership perspective)

- Policy: the operations policy prescribes signed, verified images from controlled sources and confidential computing for clearly named workloads that are particularly worthy of protection.
- Responsibilities: assigned in the Statement of Applicability (template MRIS Annex E); the platform or operations team and security work together.
- Risk analysis and SoA: which workloads require confidential computing is derived from the protection requirement (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 procured from the cloud, evidence of the provider's capabilities (confidential computing, attestation).

4. Technical implementation (for IT specialists)

- Signed images: base images and the organization's own images are signed (for example with Sigstore/cosign) and drawn from controlled registries; the signature is verified before start-up.
- Enforcement at deployment: an admission controller of the platform (for example OPA Gatekeeper or Kyverno in Kubernetes) admits only signed, verified images and blocks the rest.
- Image verification: automated vulnerability scans of the images before deployment; regular re-scanning of stored images when new vulnerabilities become known.
- Confidential computing: for highly sensitive workloads, protected execution environments (TEE/secure enclaves, for example Intel TDX, AMD SEV-SNP, ARM CCA) that encrypt data in memory as well; before use, remote attestation demonstrates 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 foundations at the Confidential Computing Consortium; requirements in C5:2026 OPS-32 to OPS-35.
- Effectiveness test: an attempt is made to start an unsigned or unverified image in the production environment — the start-up must be prevented by the platform.

5. Implementation examples

Example A — development and platform context. A vendor runs its software in containers assembled largely from public base images. Until now nobody checks whether those images are unaltered. The vendor therefore introduces end-to-end image signing: only signed images previously scanned for vulnerabilities, drawn from its own controlled registries, may start, enforced through an admission controller of the platform. For the few services that process particularly sensitive customer data, it additionally uses a protected execution environment in which the data remain encrypted in memory as well, and demonstrates the environment's authenticity before use. Manipulated images are thereby rejected, 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 organization builds or operates containers itself. An organization that uses only off-the-shelf SaaS services cannot implement this measure itself; for it, the measure becomes a procurement and evidence question: it follows up with the provider as to whether the provider signs and verifies images, prevents the start-up of unapproved images, and offers protected execution environments for particularly sensitive data — demonstrated for example through certificates or audit reports (such as C5).

6. Maturity path (cumulative)

- Initial: container images signed; vulnerability scan before deployment.
- Defined: enforcement through admission controllers in operation; use cases for confidential computing named and documented.

- Managed: confidential computing in production for the workloads classified as particularly requiring protection; remote attestation automated.

7. Measurement and audit evidence

- Metric: share of production containers from signed images verified before start-up (target: complete); additionally the coverage of confidential computing 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 or operations lead; frequency? — with every deployment, scans continuous; 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 without effect.
- The admission controller is in place but runs only in warning mode; images that are not permitted 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 the containers and the protection of data during processing; the provenance of the contained components is dealt with by MHC-02, and the separation of several tenants by MHC-07.
- Residual risk: confidential computing protects the 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 containerized workloads; confidential computing is context-dependent and often to be justified as N/A in the Statement of Applicability. Both protective layers are to be planned, assessed and evidenced separately.
- ISO/IEC 27002:2022 (indirectly): 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 in one tenant from reaching across to the data of other tenants — the damage remains confined to one tenant.
- Policy affected: policy on secure architecture and tenant separation (for multi-tenant services).
- Dependencies: no precondition; concerns organizations running several customers on a shared platform; makes use of key separation (relation to A.8.24) and network separation.
- Effort drivers: depends on the architecture model (shared versus separate resources) and the number of tenants. The actual effort depends on the organization's resources.
- Primary metric: outcome of regular, ideally automated separation tests (no cross-tenant access demonstrable).
- Standards touched (no 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; the separation is documented and demonstrably implemented — not merely asserted conceptually. Advanced hardening (Managed level): regular evidence of the separation through tests that are automated so far as possible.

2. Purpose and threat relevance

In multi-tenant services, many customers share a common platform. If an attacker succeeds in gaining a foothold in one tenant, the central question is whether they can reach across from there to the data of other tenants. AI-supported 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 protective objective is to confine the damage of an attack demonstrably to a single tenant.

3. Organizational implementation (leadership perspective)

- Policy: the architecture policy prescribes, 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 requirement of the customer data. Risk treated: cross-tenant access (linked to the risk assessment bridge, MRIS Annex F).
- Processes: regular separation tests with records; handling of identified weaknesses; provision of evidence to customers and auditors.

4. Technical implementation (for IT specialists)

- Data separation: tenant-specific keys together with logical or — at the highest protection requirement — physical separation of the data holdings.
- Network separation: separate network areas per tenant (for example dedicated virtual networks/VPCs, namespaces, rule-based separation via software-defined networking).
- Separation of compute resources: tenant-based 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 one tenant from another; the outcome (no access possible) is documented.
- Further reading: separation requirements in C5:2026 OPS-30/31 (data separation) and PSS-10 (network); the control catalogue of the CSA Cloud Controls Matrix; network separation as a principle in A.8.22.
- Effectiveness test: from a test tenant, a deliberate attempt is made to access the data or resources of another tenant — the access must fail, and the outcome is recorded.

5. Implementation examples

Example A — development and platform context. A SaaS provider runs many customers on a shared platform. Until now the separation has been described only in architecture documents but never verified. The provider introduces tenant-specific keys, separates the network areas per customer and ensures that the 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 from one tenant to others. If such an attempt fails, the separation is evidenced; if it succeeds, there is a clear finding to remediate. An asserted separation thereby becomes a demonstrated one.

Applicability note (in place of Example B). This measure is directed at organizations that operate multi-tenant services themselves. An organization that merely uses such services does not implement the separation itself; for it, the measure becomes a procurement and evidence question: it follows up with the provider as to how the provider 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 audits.

7. Measurement and audit evidence

- Metric: outcome of the regular separation tests (no cross-tenant access demonstrable); frequency and coverage of the tests.
- Stage gate (guide value): one passed, documented tenant separation test per level (Initial annually, internally; Defined per major release; Managed additionally by independent third parties). Evidence recorded in project controlling (MRIS Chapter 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 or operations lead; frequency? — regular tests; tolerance? — no demonstrated cross-tenant access.

8. Typical mistakes

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

9. Delimitation, residual risk and references

- Delimitation: concerns the separation between tenants; the protection of the processing itself is dealt with by MHC-06, and the identity of the workloads by MHC-04.
- Residual risk: a shared platform retains a common underlying layer; flaws in that common layer can affect several tenants despite the separation.
- ISO/IEC 27002:2022 (indirectly): 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; supplementary 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 entitlement management.
- Effort drivers: depends on data volume, recovery objectives and the existing backup infrastructure. The actual effort depends on the organization's resources.
- Primary metric: success rate of the regular recovery tests from immutable backups.
- Standards touched (no claim of fulfilment): ISO/IEC 27002 A.8.13/A.8.14; C5:2026 OPS-06 to OPS-09; DORA Art. 12; NIS2 IR No. 4.1; NIST SP 800-34, SP 1800-11/-25.

1. Control statement

The organization maintains at least one immutable or network-separated backup copy and verifies regularly, through documented recovery tests, that the data can be restored from it without error. The backup infrastructure is administered through separate administration access. Advanced hardening (Managed level): a dedicated administration and identity plane for the backup infrastructure, together with automated integrity and recovery checks.

2. Purpose and threat relevance

Modern, AI-supported attacks encrypt or corrupt data and deliberately seek out the 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-separated copy remains unaffected. The protective objective is to be able to fall back at any time on a clean, unaltered data basis after an incident — and to be certain that the recovery actually succeeds.

3. Organizational implementation (leadership perspective)

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

4. Technical implementation (for IT specialists)

- 3-2-1-1-0 principle: three copies of the data, on two different media types, at least one offsite, at least one immutable or network-separated (air-gapped), zero errors in the regular recovery test.
- Immutability: backups are stored in such a way that they cannot be altered or deleted for a defined period (WORM storage, object locks), or are held physically separated from the network.
- Separate access paths: the backup infrastructure uses its own accounts and entitlements, separate from the production administration 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: a full recovery test is carried out from an immutable copy; in addition, an attempt is made using a production administration account to alter or delete a backup — this must be refused.

5. Implementation examples

Example A — development and platform context. A provider has so far backed up its systems regularly, but all backups are reachable through the same administration access as the production systems. An attacker who takes over such an 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 administration of the backup infrastructure from the production access. Every quarter it restores the data fully as a test and documents the outcome. In that way a clean data basis is preserved even in the event of a far-reaching compromise.

Example B — general department. An organization without its own IT development backs up its files and mailboxes through a cloud service. It ensures that at least one copy is retained immutably (many services offer a retention or lock function for that purpose) and that administration of this backup does not depend on the same accounts as day-to-day administration. Once a quarter it restores sample files from the backup and records that it worked. In that way it 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, at least one immutable or network-separated copy; separate backup administration access (dedicated admin accounts with phishing-resistant MFA and separate entitlement assignment); documented recovery tests quarterly.
- Managed: a dedicated administration and identity plane for the backup infrastructure; automated integrity checking; orchestrated, automated recovery validation; regular adversarial deletion tests; continuous monitoring of immutability.

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 entitlement assignment for the backup infrastructure, integrity checking reports.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — policy and test records; responsible? — IT or backup owner; frequency? — quarterly tests, continuous integrity checking; tolerance? — no business-critical data holding without an immutable or separated copy and without a passed recovery test.

8. Typical mistakes

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

9. Delimitation, residual risk and references

- Delimitation: concerns backup and recovery; continuous availability through redundancy is dealt with by A.8.14, and restart planning by A.5.29/A.5.30.
- Residual risk: a network-separated copy is by its nature less current; a very slow or rarely tested recovery can still lead to downtime when it matters.
- Effectiveness limit: immutability protects the copy, not the recovery. The retention lock must lie outside the attacker's control plane — its own entitlements, a separate identity, no deletion or shortening right for compromised admin accounts; what is decisive is the regularly tested recovery against a defined RTO, not the mere existence of immutable copies.
- ISO/IEC 27002:2022: A.8.13 Information backup; A.8.14 Redundancy of information processing facilities; A.5.29 Information security during disruption; A.5.30 ICT readiness for business continuity; A.5.33 Protection of records.
- Framework: C5:2026 OPS-06 to OPS-09; DORA Art. 12; NIS2 IR No. 4.1; NIST SP 800-34 Rev. 1; NIST SP 1800-11 and 1800-25 (data integrity / ransomware recovery, NCCoE); industry practice 3-2-1-1-0.

MHC-09 — AI-supported security testing in the pipeline

At a glance

- Operational benefit: finds vulnerabilities before delivery — automatically with every change — and removes the attacker's window between a flaw becoming known and its closure.
- Policy affected: secure software development policy.
- Dependencies: presupposes an automated build and CI pipeline; uses the bills of materials from MHC-02 for component reconciliation.

- Effort drivers: depends on the number and structure of the pipelines and the volume of code produced. The actual effort depends on the organization's resources.
- Primary metric: time from the availability of a robust remediation for an actively exploited vulnerability (KEV) to the demonstrated remediation or effective compensation. Supplementary effectiveness metrics: share of pipelines with security checking and a pre-merge block, share of findings that reach production despite pipeline checking (escape rate), share of suitable applications with an AI-supported test method in production.
- Standards touched (no claim of fulfilment): ISO/IEC 27002 A.8.28/A.8.29; NIST SP 800-218 (SSDF) and SP 800-218A (AI); OWASP ASVS; C5:2026 OPS-25; CRA Annex I.

1. Control statement

Security checks run automatically in the development pipeline: code, dependencies and the running application are checked with every change; serious findings block the merge. For suitable applications, AI-supported test methods are additionally deployed — in particular for the dynamic generation of test cases, the detection of multi-stage vulnerability chains, hypothesis-based examination of attack paths, or agentic security analysis. AI-generated code is checked additionally as a risk factor in its own right.

2. Purpose and threat relevance

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

3. Organizational implementation (leadership perspective)

- Policy: the development policy prescribes 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 established. Risk treated: exploitable vulnerabilities that reach operations undetected (linked to the risk assessment bridge, MRIS Annex F).
- Processes: a governed procedure for blocking findings and for KEV cases (accelerated remediation); regular evaluation of the 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 checking (SCA) against vulnerability databases — automatically with 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 the systems already running as well, not only of new code.
- AI code as a category of its own: additional test rules against the typical weaknesses of AI-generated code (for example through pre-commit checks); regular evaluation with trend tracking; AI-generated code is taken into account as a risk factor of its own in the secure SDLC standard and in the risk assessment (risk register or application risk classification) — not in the Statement of Applicability, which documents the applicability of controls and is not a risk register.
- 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: a known insecure element is deliberately introduced in a test branch (for example a hard-coded credential); the pipeline must report the finding and block the merge.
- Effectiveness limit: AI-supported testing does not replace expert assessment. It particularly overlooks logic, authorization and architecture weaknesses — precisely the findings with high impact — and produces both false positives and false negatives. "The tool ran" is not "the vulnerability was blocked": critical findings, blocking pipeline decisions and accepted exceptions require expert triage, a documented decision and traceable approval.

5. Implementation examples

Example A — development and platform context. A software vendor has so far tested security only 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 with every change, and serious findings prevent the merge. For actively exploited vulnerabilities there is an accelerated path to close them immediately. Since its developers use AI coding assistants, it adds test rules against the typical errors of such tools and evaluates those findings separately. In that way gaps are 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 organization therefore establishes that such self-built solutions do not go into production unchecked: anything going beyond simple aids passes through a functional and security review, and for sensitive data or interfaces central IT must be involved. In that way the benefit of rapid self-development is retained without unchecked AI code reaching operations uncontrolled.

6. Maturity path (cumulative)

- Initial: SAST/DAST/SCA in the pipeline.
- Defined: additional checking of AI-generated code; blocking on serious findings; at least one AI-supported test method demonstrably in production use for suitable applications.
- Managed: daily vulnerability scans of the running systems; separate evaluation of AI code; an accelerated path for actively exploited vulnerabilities (KEV).

7. Measurement and audit evidence

- Metric: remediation latency for actively exploited vulnerabilities (KEV) — time from the availability of a robust remediation or effective compensating option to its verified implementation; patch, workaround, compensating control, isolation or replacement count as remediation (guide value: ideally below 24 hours on average); supplementary, the share of pipelines with security checking and a pre-merge block, the escape rate, and the share of suitable applications with an AI-supported test method in production.
- Evidence: pipeline configuration, test and blocking logs, evaluation of the AI code findings, penetration test reports.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — development policy, pipeline configuration; responsible? — development and product owners; frequency? — with 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 searched for specifically.
- The thresholds are set so strictly that the pipeline blocks constantly; as a consequence the checks are circumvented rather than remediated.

9. Delimitation, residual risk and references

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

Managed service variant (for organizations without their own security testing)

- Service to be procured: AI-supported 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; notification time for critical findings; re-test confirmation after remediation; reconciliation of the findings with KEV/EPSS.
- Minimum service level: continuous or at least monthly test cycles for externally reachable systems; notification of critical findings within 24 hours; an SLA for re-tests after remediation.
- Work remaining in-house: prioritization, remediation, vulnerability ownership and escalation; logic, authorization and architecture flaws remain the subject of manual review ("the tool ran" is not "the vulnerability was blocked").

MHC-10 — Continuous control monitoring and policy as code

At a glance

- Operational benefit: shortens the time to detection of a security deviation from months (the audit cadence) to minutes, because requirements are checked automatically on an ongoing basis. 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 expressed in machine-readable form and access to system and 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 actual effort depends on the organization's resources.
- Primary metric: continuous monitoring coverage — share of the SoA requirements classified as automatable with active automated checking (at least daily). The reference base is expressly automation eligibility, not the total number of SoA controls; both metrics are defined in Section 7 (MRIS Annex G.6a/G.6b).
- Standards touched (no claim of fulfilment): ISO/IEC 27002 A.5.35/A.5.36; C5:2026 COM-03/04; NIS2 IR No. 2.2; DORA Art. 6(5); NIST SP 800-137.

1. Control statement

Observance of security requirements is checked on an ongoing, automated basis against defined target states, rather than only in periodic audits. Requirements are, where possible, expressed as executable rules (policy as code) and integrated into the deployment and operations processes; deviations are alerted and tracked. Advanced hardening (Managed level): enforcement in operation (runtime enforcement) and automatically raised work items on deviation.

2. Purpose and threat relevance

Periodic audits check only at fixed points in time (quarterly, for instance); in between, deviations remain undetected. AI-supported attackers exploit precisely that gap: they create — or find — a misconfiguration and

use it long before the next audit takes place. In addition, agentic attackers decompose their approach into steps that each appear policy-conformant. Ongoing, automated checking detects deviations immediately and brings them straight to response. The protective objective is to reduce the interval between a deviation and its detection to a minimum.

3. Organizational implementation (leadership perspective)

- Policy: the monitoring policy establishes that observance of essential requirements is checked continuously in automated form and that deviations are dealt with immediately — in addition to, not instead of, the audits required in any case.
- Responsibilities: maintenance of the rules and evaluation are assigned in the Statement of Applicability (template MRIS Annex E); the ISMS leadership is normally responsible in substantive terms.
- Risk analysis and SoA: it is established which controls are monitored automatically as a priority — derived from the risk assessment and protection requirement. Risk treated: deviations from the target state that go 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 work item (ticket) and leads to remediation; regular maintenance and extension of the rule base; reporting on coverage and open deviations.

4. Technical implementation (for IT specialists)

- Policy as code: security requirements are expressed as executable rules (for example 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 (ongoing checking of the actual state against the target specifications), ideally with enforcement (runtime enforcement).
- Target states (baselines): the target specifications checked are defined and maintained; machine-readable control catalogues (for example in NIST's OSCAL format) make automated checking and evidence easier.
- Response: deviations are shown on real-time dashboards with clear metrics and automatically raise a work item (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: a deviation from the target state is deliberately created in a test area (for example a configuration that is not permitted); the automated check must report the deviation within minutes and raise a work item.

5. Implementation examples

Example A — development and platform context. A company has so far checked observance of its security requirements on a quarterly audit cadence. Between the checks, a faulty permission in a cloud environment comes to light only weeks later. The company therefore expresses its most important requirements as executable rules and integrates them both into deployment and into running operations. If a configuration breaches a rule, this appears immediately on a dashboard and automatically raises a work item for remediation. The quarterly snapshot thereby becomes an ongoing control.

Example B — general department. An organization without its own development work uses mainly cloud services. It cannot program rules of its own, but it activates the functions for ongoing configuration checking present in its platforms (for instance a security or compliance status that runs continuously). Deviations from the recommended settings are displayed there on an ongoing basis; the organization establishes who picks them up and remediates them. In that way it obtains continuous checking through the built-in facilities 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; deviations are alerted and tracked.

- Managed: enforcement in operation (runtime enforcement); automatically raised work items on deviation.

7. Measurement and audit evidence

- Metric 1: automation eligibility — share of the requirements carried in the SoA that are technically capable of automated checking. This classification is documented once and reviewed annually; it excludes requirements that do not lend themselves to daily automated checking (personnel matters, contracts, physical measures, governance, management review, contact with authorities, policy maintenance).
- Metric 2: continuous monitoring coverage — share of the requirements classified as automatable with active automated checking (at least daily, with alerting on deviation). Guide values: 60% or above after 12 months, 80% or above after 24 months. The denominator is expressly only the automatable requirements; measuring against all SoA controls would distort the metric and invites gaming. Optionally evaluate on a risk-weighted basis in addition.
- Evidence: rule repository (policy as code), overview of the automatically checked controls, logs of work items raised on deviation.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — rule base and coverage overview; responsible? — ISMS leadership; frequency? — continuous (at least daily); tolerance? — essential controls without automated checking only on a time-limited and justified basis.

8. Typical mistakes

- Automated checking is understood as a substitute for audits; in fact it supplements them — the required reviews remain necessary.
- Dashboards are built, but deviations trigger no response; the finding remains without consequence.
- Only easily checkable requirements are automated, while the essential ones are not; coverage appears 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 ongoing checking of observance of requirements; the detection of active attacks is dealt with by MHC-05, and automated response by MHC-11.
- Residual risk: only what is expressed as a rule is checked — unclear or non-machine-checkable requirements remain outside; policy-conformant 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 IR No. 2.2; DORA Art. 6(5); OWASP SAMM; NIST SP 800-137 and 800-137A (ISCM); NIST OSCAL (machine-readable controls); policy engines Open Policy Agent (OPA/Rego), HashiCorp Sentinel.

MHC-11 — SOAR-based tier 1 automation and parallel response playbooks

At a glance

- Operational benefit: brings the response to unambiguous incidents down 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 (partially 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 actual effort depends on the organization's resources.
- Primary metric: time to containment of an unambiguous incident (mean time to containment, MTTC).
- Standards touched (no claim of fulfilment): ISO/IEC 27002 A.5.24/A.5.26; C5:2026 SIM-02/03; NIS2 IR No. 3.5; DORA Art. 17; NIST SP 800-61 Rev. 3.

1. Control statement

To unambiguous incidents the organization responds with predefined, SOAR-supported and at least partially automated procedures (playbooks) that initiate containment in under 60 minutes; ambiguous or serious cases are routed deliberately to people. The automation layer itself is secured. Advanced hardening (Managed level): fully automated containment of unambiguous incidents in under 10 minutes, together with regularly exercised handling of several simultaneous incidents — alternatively an MDR service with a contractually assured response time.

2. Purpose and threat relevance

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

3. Organizational implementation (leadership perspective)

- Policy: the incident response policy establishes which incidents are contained automatically, when escalation to people takes place, and how the automation layer is protected.
- Responsibilities: assigned in the Statement of Applicability (template MRIS Annex E); security operations is oriented towards analysis and incident command rather than mere alert review.
- Risk analysis and SoA: which actions may run automatically is determined by unambiguity and potential damage. 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 (for example 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, a parallel multi-vector attack, supply chain compromise, and waves of attacks on sign-in.
- Hardening of the automation layer (central): the SOAR pipeline is itself an attack surface. Protective measures: only authenticated, signed triggers from trusted sources; a complete record of every playbook execution including the executing identity; rate limits per playbook (against floods of false alerts); human approval (human in the loop) for actions with a large blast radius (for example mass account lockout, isolation of larger network areas, certificate revocation); protection of the playbooks like source code (version control, code review, signed approvals).
- Further reading: structure and life cycle of incident handling in NIST SP 800-61 Revision 3 (aligned to CSF 2.0); requirements in C5:2026 SIM-02/03 and NIS2 IR No. 3.5.
- Effectiveness test: an unambiguous attack signal is triggered in a test environment; the matching playbook must initiate containment automatically and within the target time — and an action with a large blast radius may take place only after human approval.

5. Implementation examples

Example A — development and platform context. A company with its own security operations finds that the time from alert to first countermeasure is too long, because every step runs manually. It therefore automates the unambiguous containment steps: on clear signals an affected account is locked immediately or a system automatically isolated, while ambiguous cases go to the analysts. So that the automation does not itself become a weakness, the platform accepts only authenticated triggers, records 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 organization without its own 24/7 operations cannot build and maintain such automation itself. It therefore procures detection and rapid response as a service (managed detection and response) and, in selecting a provider, looks for a contractually assured response time and for traceable, agreed containment measures. Internally it establishes who is contactable when it matters and which far-reaching actions are pre-approved. In that way it obtains a fast, exercised response without operating a platform of its own.

6. Maturity path (cumulative)

- Initial: response playbooks documented; manual execution.
- Defined: SOAR with partially 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 containment of an unambiguous incident (MTTC; guide value: under 10 minutes at high confidence).
- Evidence: playbook definitions and logs, evidence of the hardening of the automation layer (trigger authentication, audit trail, approval rules), records of the multi-incident exercises or the service provider SLA.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — incident response policy, playbooks; responsible? — security operations or SOC lead; frequency? — continuous, exercises at least half-yearly; tolerance? — no automatic action with a large blast radius without human approval.

8. Typical mistakes

- Automation is applied broadly without protecting the automation layer; unauthenticated triggers or floods of false alerts set off unintended containment.
- Far-reaching actions run without human approval; a false alert then locks many accounts or isolates whole network areas.
- Playbooks exist but there is no exercise for several simultaneous incidents; when it matters, a parallel attack overloads the response.
- The playbooks are not maintained like source code (no version control, no review); errors and manipulation go unnoticed.

9. Delimitation, residual risk and references

- Delimitation: concerns the response to detected incidents; detection itself is dealt with by MHC-05, and ongoing 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 organization itself; without the hardening described, the measure is not effective.
- Effectiveness limit: SOAR does not replace robust 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 damage 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 IR No. 3.5; DORA Art. 17; NIST SP 800-61 Revision 3 (incident handling, aligned to CSF 2.0).

Managed service variant (for organizations without their own SOAR)

- 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: a playbook catalogue with approval levels; complete execution logs per automatic action; exercise records at least half-yearly; the rate of false triggers.
- Minimum service level: defined execution times per action; human approval for actions with a large blast radius anchored contractually; the ability to roll back every automatic action.
- Work remaining in-house: approval and maintenance of the playbooks, definition of the blast radius limits, decision authority on 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 thereby 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 and MHC-11) under realistic conditions.
- Effort drivers: depends strongly on the format chosen (a full TLPT with an external red team versus lower-cost purple team and simulation formats). The actual effort depends on the organization's resources.
- Primary metric: share of identified vulnerabilities remediated within the agreed time window (closure rate).
- Standards touched (no claim of fulfilment): ISO/IEC 27002 A.5.35/A.8.29; DORA Art. 26/27 with the TLPT RTS; TIBER-EU; C5:2026 OPS-22; NIS2 IR No. 3.5.5.

1. Control statement

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

2. Purpose and threat relevance

Documented protective measures are not the same as effective protective measures. Under AI-supported attacks — which demonstrably find and exploit vulnerabilities autonomously today — only a realistic test shows whether the defence actually takes hold. Classic penetration tests with a fixed standard scope often do not reflect the new approaches. The protective 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. Organizational implementation (leadership perspective)

- Policy: the testing policy prescribes regular threat-led tests with realistic scenarios and purple team exercises between the cycles.

- Responsibilities: assigned in the Statement of Applicability (template MRIS Annex E); at the highest maturity, timely remediation of the findings is reported to the leadership.
- Risk analysis and SoA: scope and scenarios are derived from the threat landscape and the critical functions; the appropriate format is chosen according to protection requirement 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 team); tracking and timely closure of the findings; regular repetition (at least annually for critical functions). The statement "at least annually" is an MRIS guide value and is not to be equated with the regulatory DORA TLPT cycle; that cycle applies only to the financial entities identified under Art. 26(8) and there generally provides for a multi-year rhythm.

4. Technical implementation (for IT specialists)

- Threat-led tests: external red teamers reproduce real attack chains derived from current threat analysis — tailored to the organization, against the production systems.
- Include Mythos scenarios: for example AI-supported spear phishing with a spoofed voice (deepfake voice), attack chains decomposed into micro-steps, parallel multi-vector attacks, supply chain compromises, abuse of cloud access through compromised service accounts.
- Purple team between the cycles: attack and defence sides work together and map what has been exercised to the techniques of MITRE ATT&CK, in order to improve detection coverage (MHC-05) in a targeted way.
- Protective requirements for tests against production systems (binding, to be fixed in writing before testing begins): written authorization by the leadership; rules of engagement with time windows, target systems and permitted techniques; defined abort and stop criteria with named emergency contacts and availability during the test; express exclusion of safety-critical and irreversible actions; protection of production data (no exfiltration of real datasets, evidence rather than a copy); required approvals from cloud and third-party providers before testing begins; prepared recovery and rollback procedures; governed evidence preservation and retention of the test artefacts. Linked to ISO/IEC 27002 A.8.34 (protection of information systems during audit testing).
- Graduated formats: a full TLPT under the TIBER-EU methodology (demanding, for regulated or critical entities) or lower-cost formats — purple team exercises with ATT&CK scenarios, open source attack simulation (for example Caldera, Atomic Red Team) and breach-and-attack simulation platforms (for example AttackIQ, SafeBreach, Picus).
- Further reading: the 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 test requirements in C5:2026 OPS-22.
- Effectiveness test: in a threat-led scenario (for example an attack chain decomposed into micro-steps), it is examined whether the existing detection and response actually notice and stop the attack — an unnoticed run is the finding.

5. Implementation examples

Example A — development and platform context. A company has an annual standard penetration test carried out that proceeds in a similar way each time. New, AI-typical approaches are not tested. The company therefore moves to threat-led tests: external red teamers reproduce real attack chains — for instance an approach decomposed into many small steps, or phishing with a spoofed voice — and check against the real systems whether detection and response take hold. Between the tests, attack and defence sides exercise together and close the detection gaps identified. In that way it becomes visible which protective measures really work and which are merely documented.

Example B — general department. A smaller organization cannot sustain a full, demanding testing programme. It therefore chooses a lower-cost but effective format: regular joint exercises of attack and defence using realistic scenarios, supported by open source attack simulation or a simulation platform that reproduces typical attack techniques automatically. The results show concretely where detection needs improvement. In that way it too 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 the cycles.
- Managed: continuous attack simulation, a simulation platform in production; timely remediation of the findings (closure rate 90% or above within the agreed time window).

7. Measurement and audit evidence

- Metric: closure rate — share of identified vulnerabilities remediated within the agreed time window (guide value: 90% or above); supplementary, 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 leadership; frequency? — at least annually for critical functions, purple team in between; tolerance? — no open critical findings beyond the agreed time window.

8. Typical mistakes

- Testing takes place, but always with the same standard scope; realistic, new attack patterns remain outside.
- Findings are reported but not remediated on time; the test merely confirms what was already 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 dealt with by MHC-09, and ongoing detection by MHC-05.
- Residual risk: a test is a snapshot of a chosen scenario; paths not tested 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 those DORA-regulated financial entities identified for TLPT under Art. 26(8), the regulatory DORA TLPT requirements (RTS (EU) 2025/1190) apply bindingly; all others achieve the objective equally 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 IR 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 production AI agents and agentic workflows under control — self-developed as well as SaaS, low-code and no-code agents: 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 deployed; later build-out stages presuppose suitable tools available in the organization's own environment. The actual effort depends on the organization's resources.
- Primary metric: share of production agents with a documented threat model and limited access, together with the coverage of logging.
- Standards touched (no 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

Production AI agents and agentic workflows — self-developed as well as SaaS, low-code and no-code agents — are treated like privileged systems: a limited blast radius per agent, an identity limited to the necessary entitlements, complete and traceable logging with a named human owner and a kill switch, and an allow list for agentic components (MCP servers, IDE extensions, skills, connectors).

Minimum requirements for self-developed or self-controlled agents: a technical, capability-scoped identity instead of personal user accounts or blanket API keys; expressly no unrestricted tool access; tamper-resistant, verifiable execution logging — the log is security-relevant evidence and is connected to the central, tamper-resistant logging from A.8.15 (MHC-05); documented purpose limitation and an approved tool scope per function; defined approval thresholds for actions with a high damage impact. The identity basis is supplied by MHC-04 (workload identity), on which the capability-scoped limitation builds. Advanced hardening (Managed level): code review for the harness components under the organization's own control.

Minimum requirements for SaaS, low-code, no-code and externally operated agents: an agent-specific identity so far as the platform supports it, otherwise dedicated service accounts with least-privilege delegation; admin consent and permission control at platform level; allow-listing of connectors and extensions; documented permission boundaries; activation and regular evaluation of the platform-side logging, with export where available; vendor assessment and contractual assurances (vendor assurance); configuration controls against unintended privilege escalation. Harness code review applies here only to components under the organization's own control (for example its own connectors, scripts or prompt and flow definitions).

2. Purpose and threat relevance

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

3. Organizational implementation (leadership 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 of human approval before irreversible actions.
- Responsibilities: an AI governance function is named. Accountability is established in accordance with the organization-specific AI and risk governance; the CISO, CIO, CTO, an AI governance function or the responsible business or service owner typically come into consideration. Risk acceptances above delegated thresholds remain with the leadership function authorized for that purpose; the board is involved, because approving production agents touches risk acceptance decisions above the CISO mandate (MRIS Annex E, Chapter 10.5).
- Risk analysis and SoA: before going into production, a threat and risk analysis is prepared and documented per 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 deployment (documented purpose of use, risk analysis, fallback plan); (b) an inventory of the AI agents and their components; (c) a procedure for complete traceability with a named owner; the retention period is established and documented on a risk basis and having regard to legal, contractual and forensic requirements — agent logs may contain personal data, prompts, credentials, source code or customer data, so that both too short and too long a period are problematic; for security-relevant agents an organization-specific minimum retention period should be defined; (d) a procedure for the rapid revocation of compromised agents.
- Procurement and suppliers: requirements for external agentic components (provenance, update path, security evidence) are established; admission only via the allow list.
- Training: development and business teams are made aware of the risks, in particular of indirect prompt injection.
- Harness as code: it is established that the control components of an agent (instructions, tool definitions, rule files) are subject to the same approval and version 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 entitlements per tool call (read, write and network rights with an explicit scope); no personal developer tokens.
- Blast radius limitation: 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, expenditure above a threshold).
- Logging: every tool call, together with file and network access, is logged as a reproducibly traceable record, bound to session, human owner and data source; audit-proof storage.
- Allow list: technical blocking of MCP servers and extensions that are not approved; controlled rather than automatic updates; signature verification where available.
- Hardening against prompt injection: separation of the instruction and data channels so far as technically possible; input and output filtering; treatment of tool outputs as untrusted input (protection against abused tool entitlements, the "confused deputy" problem); penetration tests against prompt injection before production deployment.
- Harness as code: instructions, tool definitions, retrieval pipelines and escalation logic held in the version control system, with mandatory code review, signed releases and automated checks before production deployment.
- 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 the agent models in NIST AI 100-2e2025.
- Effectiveness test: a prepared email or ticket with an embedded command (injection probe) is introduced; the agent must not carry out the impermissible action demanded in it (for example external transmission), and the attempt must appear in the log.

5. Implementation examples

Example A — development and platform context. In an organization with its own software development, the developers use an AI assistant directly in their working environment. That assistant is connected to several internal systems — the ticket system, source code management and an internal knowledge base — and works under the developers' personal accounts. The risk: a covert command can be smuggled into a ticket that induces the assistant to release data externally or alter source code, without the developer noticing. The organization first maintains a complete register of all assistants deployed and their connections, permits only vetted connections, gives each assistant its own, narrowly limited access instead of full developer rights, and records completely what it does and who started it. In a further build-out stage, the assistant's control specifications are reviewed and versioned like program code, tested regularly for susceptibility to manipulation, and a misused access can be revoked within a few minutes.

Example B — general department (no-code/SaaS). A business department without its own development work — sales, for instance — uses Microsoft 365 Copilot and sets up an assistant of its own within it using the built-in facilities (or uses a self-created GPT in ChatGPT Business). That assistant may read the mailbox, the file store and the customer database and may send messages, and does so with the full rights of the member of staff who set it up. The risk: an incoming email or document can contain a covert command that induces the assistant to send confidential content externally — precisely this attack pattern was demonstrated for Microsoft 365 Copilot in the publicly documented EchoLeak case. Here the levers lie not in programming but in the administrative settings of the platform: the department records which assistants exist and what they may access, permits only approved connections, limits the assistants' entitlements to what is necessary instead of full user rights, names a responsible person for each production assistant, and switches on the platform's logging. For irreversible steps — such as external transmission or the deletion of data — express human confirmation is required.

6. Maturity path (cumulative)

- Initial: production AI agents and agentic workflows are inventoried — including SaaS, low-code and no-code agents; a manual inventory of the MCP servers and extensions.
- Defined: limited identities and complete logging in production; a named human owner and a tested kill switch per agent; a threat model per purpose of use; an allow list for extensions, connectors and MCP servers.
- Managed: a full code review process for the harness components under the organization's own control; automated checks before production deployment; mean time to revoke below 5 minutes; regular penetration tests against prompt injection.
- Realism: most organizations start at Initial; realistically 12 to 18 months to Defined, 18 to 24 months to Managed. Prioritize first: inventory, logging and limited identities — these three address the bulk of the risk. Penetration testing of the harness and checks of model robustness follow in later build-out stages, depending on the tools available in the organization's own environment.

7. Measurement and audit evidence

- Metrics: share of production AI agents and agentic workflows within the organization's area of responsibility — including SaaS, low-code and no-code agents — with a documented threat model and a defined blast radius (target: 100%); collected separately as inventory coverage (fully inventoried) and governance coverage (owner, identity, permission boundaries and tamper-resistant log implemented); coverage of logging for actions with write or network rights (target: 100%); time to revocation of compromised agent identities (target: below 5 minutes).
- Evidence: agent inventory, allow list configuration, logs, pre-production test reports, threat model documents.
- Evidence logic (following ISO/IEC 27001, Clause 9): documented? — through the inventory and threat models; responsible? — a named human per agent; frequency? — continuous logging, a check before every production deployment; tolerance? — no production agents with write or network rights without logging.

Market and tooling map (selection criteria, status July 2026)

The market for agent governance tooling is young and consolidating; product names date quickly. For level 2 evidence, one tool per category that meets the criteria named is sufficient:

- Agent and NHI inventory: automatic capture of non-human identities (service accounts, API keys, agents) with owner assignment and export to, or reconciliation with, a CMDB, AI agent register or GRC/asset management platform (the Statement of Applicability documents the applicability of the control but does not take individual agents in as inventory objects); criterion: coverage measurement against IdP and platform data (the agent inventory coverage metric).
- Agent identity and capability scoping: IdP-native agent identities or workload identity (SPIFFE/SPIRE, cf. MHC-04) with short-lived credentials and fine-grained entitlements; criterion: revocation in minutes (mean time to revoke).

- MCP and tool server security: signing and provenance verification of tool servers and manifests, registry control, checking for tool poisoning and prompt injection vectors; criterion: refusal of unsigned tools as an enforceable policy (the logic from MHC-02 carried over).
- Runtime guardrails and logging: policy-driven action approvals, session recording, tamper-resistant execution logs with SIEM integration (A.8.15); criterion: complete, forgery-proof traceability per agent action.

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

8. Typical mistakes

- Agents run under personal developer accounts — no limited identity, no clean traceability.
- MCP servers and extensions without an allow list and without update control — automatic updates open a path through the supply chain.
- The harness (instructions, rule files) is treated as configuration rather than as code — no approval, no version control.
- Human approval in name only (a confirmation dialogue without a genuine opportunity to check) — the excessive freedom of action remains.
- The target "Managed" is set across the board even though the tools and procedures needed for it are not yet established in the organization's own environment.

9. Delimitation, residual risk and references

- Delimitation: covers the governance and security of production AI agents and agentic workflows within the organization's area of responsibility — self-developed as well as SaaS, low-code and no-code agents; the identity basis is supplied by MHC-04; insecure AI-generated code is dealt with in MHC-09; the inventory of unapproved AI tools (shadow AI) is set out in A.5.9 (MRIS extension).
- Residual risk: prompt injection is not fully solvable as long as the instruction and data channels cannot be separated; detection is not prevention; suitable hardening tools are not available in every environment.
- ISO/IEC 27002:2022: A.5.9 Inventory of information and other 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 2026 (LLM01 prompt injection; LLM03 excessive agency — with the causes excessive functionality, excessive permissions, excessive autonomy; LLM04 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 AI Agent Tool Invocation (formerly LLM Plugin Compromise; ATLAS status August 2026), AML.T0086 exfiltration via AI agent tool invocation, AML.T0110 AI agent tool poisoning).

Glossary

- **3-2-1-1-0:** backup rule of thumb (three copies, two media types, one offsite, one immutable/offline, zero errors in the test).
- **AAL2 / AAL3:** authentication assurance levels under NIST SP 800-63B; AAL3 requires a device-bound, non-exportable key.
- **Admission controller:** a control point of the platform (for example in Kubernetes) that admits only permitted, verified images at deployment.
- **Adversary emulation / BAS:** reproduction of real attack techniques, automated through breach-and-attack simulation platforms.
- **Air-gapped:** a backup copy physically separated from the network.
- **AI-generated code (vibe-coded):** code produced with AI coding assistants, to be checked as a risk category of its own.
- **Baseline (target state):** a defined target state against which the actual state is checked.
- **Blast radius:** the area of damage impact — how far a successful attack can spread.
- **Capability scoping:** limiting an access to exactly the entitlements and capabilities necessary for the task in hand.
- **Conditional access:** access rules that check user, device and situation at every sign-in before granting access.
- **Confidential computing / TEE:** a protected execution environment that encrypts data in memory during processing as well (for example Intel TDX, AMD SEV-SNP, ARM CCA).
- **Container / image / registry:** a container is an application packaged in a standardized way; the image is its immutable template; a registry is the storage location for images.
- **Containment:** an immediate measure that limits an incident in progress (for example locking an account, isolating a system).
- **Continuous control monitoring:** ongoing, automated checking of observance of requirements, in addition to periodic audits and other independent reviews.
- **CVE/NVD, OSV, EUVD:** vulnerability databases (NVD: the US database; OSV: open source vulnerabilities; EUVD: ENISA's EU database).
- **CycloneDX / SPDX:** the two established SBOM formats (CycloneDX standardized as ECMA-424, SPDX as ISO/IEC 5962).
- **Deepfake voice:** an artificially generated, deceptively realistic voice, for example for spear phishing.
- **FIDO2 / WebAuthn:** an open standard for phishing-resistant sign-in with cryptographic binding to the address of the service.
- **Harness:** the control scaffolding of an AI agent — instructions, tool definitions, rule files and escalation logic.
- **"Harvest now, decrypt later":** an attack pattern in which data encrypted today are intercepted and retained for later decryption.
- **Human in the loop:** mandatory human approval before actions with a large blast radius.
- **Hybrid cryptography:** the combination of a classical and a quantum-safe method during the transition period.
- **Image signature (Sigstore/cosign):** a cryptographic signature of an image, verified before start-up in order to exclude manipulation.
- **KEV (known exploited vulnerabilities):** the CISA catalogue of actively exploited vulnerabilities; triggers accelerated patching.
- **Kill chain correlation:** linkage of individual events into a connected attack chain.
- **Living off the land:** an approach in which attackers use built-in system tools instead of imported malware.
- **MCP (Model Context Protocol):** an open interface through which an AI assistant is connected to external tools and data sources.

- **MDR (managed detection and response):** procurement of detection and rapid response as a service with a contractual response time.
- **MITRE ATT&CK:** a recognized catalogue of real attack techniques to which detection is aligned.
- **ML-KEM / FIPS 203:** a quantum-safe method for key exchange (replacing for example RSA/ECDH).
- **mTLS (mutual TLS):** an encrypted connection in which both sides authenticate each other by certificate — not only one side.
- **MTTC (mean time to containment):** the average time to containment of an incident.
- **OSCAL:** NIST's machine-readable format for control catalogues and their evidence.
- **OSV, EUVD:** see CVE/NVD.
- **Passkey:** a sign-in credential based on FIDO2; device-bound or synchronizable across several devices.
- **Playbook:** a predefined response procedure for a particular incident type.
- **Policy as code:** security requirements as executable, versioned rules (for example OPA/Rego, Sentinel).
- **PQC (post-quantum cryptography):** encryption and signature methods intended to withstand future quantum computers as well.
- **Pre-merge block:** an automatic block that prevents 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 into unwanted behaviour.
- **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:** enforcement of requirements in running operations, not merely checking them.
- **SAST / DAST / SCA:** static code analysis, dynamic testing of the running application, dependency checking.
- **SBOM (software bill of materials):** a machine-readable register of all components of a piece of software together with their dependencies.
- **Service mesh:** an infrastructure layer that handles connection and protection between services automatically, without changing the application code.
- **SLSA / build provenance:** a framework and evidence of the sources and process from which an artefact was built.
- **SOAR:** a platform that executes defined response procedures (playbooks) automatically (security orchestration, automation and response).
- **SPIFFE / SPIRE:** an open standard (SPIFFE) and its reference implementation (SPIRE) for giving every software component a unique, technical identity.
- **SSDF (secure software development framework):** the 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.
- **Tenant / multi-tenancy:** a customer on a shared platform; multi-tenancy denotes the operation of several customers on shared infrastructure.
- **Threat hunting:** the targeted, hypothesis-based search for as-yet-undetected attacks.
- **TIBER-EU:** the ECB framework for threat-led testing in the financial sector; aligned to DORA since February 2025.
- **TLPT (threat-led penetration testing):** a threat-led, realistic test that reproduces real attack scenarios.
- **Transitive dependency:** an indirectly included building block (a library that itself uses further libraries).
- **UEBA:** behaviour-based detection using 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, separately from the static SBOM.

- **ZTNA / identity-aware proxy:** access to applications on the basis of the verified identity of user and device rather than through membership of the network; a replacement for the classic VPN.

Effort classes per MHC (guide values)

The following classes give an order of magnitude for the internal implementation effort per maturity level: S = up to roughly 10 person-days, largely using built-in facilities; M = roughly 10–40 person-days, possibly with a tool license; L = more than 40 person-days, or a platform or service investment. License and service costs are not included; the values are to be adjusted organization-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-13	S	M	L

For MHC-05, MHC-09 and MHC-11, the managed service variant typically reduces the in-house effort by one class; the effort shifts into 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 establishes no certification scheme of its own. Procedure per applicable MHC: put the audit question, inspect the minimum evidence for the Defined level, take a sample, and 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 prioritized for migration?	Inventory extract, migration strategy, review record	A.8.24
MHC-02	SBOM for each own artefact, query capability evidenced?	SBOM artefacts, pipeline configuration, correlation evidence against a vulnerability notification	A.5.21, A.8.30
MHC-03	Phishing-resistant MFA enforced for privileged and external accounts?	IdP evaluation, 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	Core audit question	Minimum evidence (Defined level)	Primary ISO reference
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 exercised?	WORM/immutability configuration, restore records	A.8.13, A.8.14
MHC-09	AI-supported testing in the pipeline, KEV SLA met?	Pipeline runs, KEV latency evaluation, re-test evidence	A.8.29, A.8.8
MHC-10	Continuous control checking 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 evaluation	A.5.26
MHC-12	TLPT with Mythos scenarios carried out, findings closed?	TLPT report, findings closure evaluation	A.5.35, A.8.29
MHC-13	Agents inventoried, execution logged tamper-resistantly?	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 list of proposals to the auditor. All evidence terms correspond to the "Measurement and audit evidence" sections of the respective MHC modules.

Mapping: Implementation Guide ↔ MRIS, Chapter 9

- **MHC-01 — Post-quantum strategy and cryptographic inventory:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; metric: cryptographic inventory coverage.
- **MHC-02 — SBOM and build provenance:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; metric: SBOM coverage.
- **MHC-03 — Phishing-resistant multi-factor authentication:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; gap cluster 6 (identity/zero trust) in 9.3; metric: phishing-resistant account coverage.
- **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 through implementation stage gates rather than 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 through implementation stage gates (signature and 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 through implementation stage gates (outcome of the separation tests).
- **MHC-08 — Immutable backups and recovery validation:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; derived primarily from the control degradation analysis of A.8.13, A.5.29 and A.5.30, with supplementary framework references under MRIS 9.2 and 9.3; metric: restore test success rate.
- **MHC-09 — AI-supported security testing in the pipeline:** catalogue 9.2, compact overview 9.4, maturity levels 9.5; gap cluster 7 in 9.3; metric: remediation 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 checking) 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 and 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 through implementation stage gates (among others mean time to revoke); metrics: agent inventory coverage and agent governance coverage.

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

Version history

Version	Date	Changes
1.2	June 2026	Base edition of this guide (thirteen MHC modules, prioritization, glossary, mapping).
1.3	July 2026	References to MRIS Annexes E and G corrected; evidence logic in place of "audit logic under ISO/IEC 27005"; managed service variant for MHC-05, MHC-09 and MHC-11; effort class table; ISO linkages aligned to MRIS Annex A; reconciliation with MRIS 1.7; license changed to CC BY-SA 4.0.
1.4	July 2026	Multi-vendor sharpening in line with MRIS 1.8: MHC-02 (disclosure waves and query capability), MHC-09 (vendor-independent tooling choice), MHC-12 (multi-system scenario pool).
1.5	August 2026	Passkey evidence update in line with MRIS 1.9: MHC-03 supplemented by authenticator attestation, relying-party-side checks and an effectiveness limit; MHC-05 supplemented by detection signals for WebAuthn and passkey abuse; MHC-13 references pinned to the OWASP Top 10 for LLM Applications 2026 (LLM03 excessive agency).
1.6	August 2026	PQC key management deepened in the MHC-01 module: trust-now-forge-later addressed in the threat relevance section; downgrade protection for hybrid negotiation, migration order within the key hierarchy (root and KEK first) and quantum-safe timestamps added to the technical implementation; migration order supplemented in the organizational implementation and the typical mistakes; CSA source added to the references.
1.7	August 2026	Referencing to MRIS switched over: the guide refers to MRIS through "Chapter 9" and the stable identifiers MHC-01 to MHC-13 rather than through a version number (title, introduction, mapping heading). The mapping thereby remains valid across MRIS version changes. Linguistic corrections from the technical review (Heldt): role reference opened throughout to the leadership or ISMS ownership function (introduction, audience, Section 3 of all thirteen modules); scope of the CRA SBOM obligation narrowed to manufacturers of products with digital elements; multi-factor criterion for device-bound passkeys clarified in MHC-03.
2.0	August 2026	Quality release after an external quality, CISO and audit review; joint version management with MRIS 2.0. New Section 9 "Reading the maturity levels" (Initial is not evidence of implementation); threat priority score without factor weights, sensitivity check through rating steps rather than percentage variation; duplicate section numbering in the framework part cleaned up. MHC-03 authentication references switched to NIS2 IR No. 11.6/11.7 and 11.3.2(a); MHC-04 identity and credentials made technically precise, effect against lateral movement no longer stated in absolute terms; MHC-09 AI code as a risk factor in the secure SDLC rather than in the Statement of Applicability; MHC-10 metric separated into two quantities (automation eligibility and continuous monitoring coverage); MHC-12 supplemented by binding protective requirements for tests against production systems and by A.8.34; MHC-13 retention

Version	Date	Changes
		period risk-based rather than blanket. Mappings to MRIS Annex A completed for MHC-02 and MHC-05; spelling of "AI" harmonized; damaged formatting cleaned up. Control statements and the Defined level synchronized across all thirteen modules: requirements previously reachable only at Managed are marked in the statement as advanced hardening (Managed level), and elements of the statement missing from Defined were added (MHC-03 privileged access and the exclusion of password-only sign-in, MHC-08 separate backup administration access, MHC-10 alerting and tracking of deviations, MHC-13 named owner and tested kill switch).