# MRIS for TISAX® 2027 Implementation Guide

## Mythos-resistant information security — Organizational and technical implementation, maturity and verification.

This document is the implementation layer for Part I.

**Cross-framework · Evidence-based**

Version 1.0 | June 2026
Creator: Richard Peddi

**TARGET GROUP**
CISOs and ISMS managers at organizations subject to TISAX®.

# License and Notices



## Independence Notice

TISAX® is a registered trademark of the ENX Association; The ENX Association administers TISAX® on behalf of the German Association of the Automotive Industry (VDA), which develops and publishes the underlying questionnaire (VDA ISA). This analysis is an independent, privately created work with no connection to the ENX Association or VDA. It is not part of the official TISAX® testing process, nor is it authorized, tested or confirmed by ENX, and does not replace a TISAX® maturity assessment by an accredited testing service provider.

## Disclaimer

The classifications in this document are a professional assessment as of July 2026, not legal or certification advice. The application is carried out without guarantee and under your own responsibility.

## Citation recommendation

Peddi, Richard (2026): MRIS for TISAX® 2027 Implementation Guide v1.0

# Table of Contents

**LICENSE AND NOTICES** ..... **2**

Independence Notice ..... 2
Disclaimer ..... 2
Citation recommendation ..... 2

**PURPOSE AND CONVENTIONS** ..... **6**

Fixed structure per MHC ..... 7

**PRIORITIZING MHC — THREAT CORRELATION AND ROADMAP** ..... **8**

1. Scope of action: how many individual findings each MHC carries ..... 8
2. Two Priority Views and Their Relationship ..... 9
3. Evaluation logic: Threat Priority Score ..... 9
4. Threat-to-MHC matrix ..... 10
5. Dependencies as roadmap rules ..... 11
6. Standard prioritization P0 / P1 / P2 ..... 11
7. Four-step approach ..... 12
8. Wording for governance documents ..... 13

**MHC-03 — PHISHING-RESISTANT MULTI-FACTOR AUTHENTICATION** ..... **13**

1. Curing Control Statement ..... 13
2. Purpose and threat relevance ..... 13
3. Organizational implementation (CISO perspective) ..... 14
4. Technical implementation (for IT professionals) ..... 14
5. Implementation examples ..... 14
6. Maturity path (cumulative) ..... 15
7. Measurement and audit evidence ..... 15
8. Typical mistakes ..... 15
9. Demarcation, residual risk and references ..... 15

**MHC-05 — BEHAVIOR-BASED DETECTION AND KILL CHAIN CORRELATION** ..... **16**

1. Curing Control Statement ..... 16
2. Purpose and threat relevance ..... 16
3. Organizational implementation (CISO perspective) ..... 17
4. Technical implementation (for IT professionals) ..... 17
5. Implementation examples ..... 17
6. Maturity path (cumulative) ..... 18
7. Measurement and audit evidence ..... 18
8. Typical mistakes ..... 18
9. Demarcation, residual risk and references ..... 19

**MHC-08 — IMMUTABLE BACKUPS AND RECOVERY VALIDATION** ..... **19**

1. Curing Control Statement ..... 19
2. Purpose and threat relevance ..... 19
3. Organizational implementation (CISO perspective) ..... 20
4. Technical implementation (for IT professionals) ..... 20

5. Implementation examples ..... 20
6. Maturity path (cumulative) ..... 20
7. Measurement and audit evidence ..... 21
8. Typical mistakes ..... 21
9. Demarcation, residual risk and references ..... 21

**MHC-09 — AI-POWERED SECURITY TESTING IN THE PIPELINE ..... 22**

1. Curing Control Statement ..... 22
2. Purpose and threat relevance ..... 22
3. Organizational implementation (CISO perspective) ..... 23
4. Technical implementation (for IT professionals) ..... 23
5. Implementation examples ..... 24
6. Maturity path (cumulative) ..... 24
7. Measurement and audit evidence ..... 24
8. Typical mistakes ..... 25
9. Demarcation, residual risk and references ..... 25

**MHC-10 — CONTINUOUS CONTROL MONITORING AND POLICY-AS-CODE ..... 26**

1. Curing Control Statement ..... 26
2. Purpose and threat relevance ..... 26
3. Organizational implementation (CISO perspective) ..... 26
4. Technical implementation (for IT professionals) ..... 27
5. Implementation examples ..... 27
6. Maturity path (cumulative) ..... 28
7. Measurement and audit evidence ..... 28
8. Typical mistakes ..... 28
9. Demarcation, residual risk and references ..... 28

**MHC-11 — SOAR-BASED TIER 1 AUTOMATION AND PARALLEL RESPONSE PLAYBOOKS ..... 29**

1. Curing Control Statement ..... 29
2. Purpose and threat relevance ..... 29
3. Organizational implementation (CISO perspective) ..... 29
4. Technical implementation (for IT professionals) ..... 30
5. Implementation examples ..... 30
6. Maturity path (cumulative) ..... 30
7. Measurement and audit evidence ..... 30
8. Typical mistakes ..... 31
9. Demarcation, residual risk and references ..... 31

**MHC-13 — AI AGENT GOVERNANCE AND HARNESS SECURITY ..... 31**

1. Curing Control Statement ..... 32
2. Purpose and threat relevance ..... 32
3. Organizational implementation (CISO perspective) ..... 33
4. Technical implementation (for IT professionals) ..... 33
5. Implementation examples ..... 34
6. Maturity path (cumulative) ..... 34

7. Measurement and audit evidence ..... 35
8. Typical mistakes ..... 35
9. Demarcation, residual risk and references ..... 36

**MHC-14 — INDEPENDENT VERIFICATION OF SUPPLIER EVIDENCE ..... 36**

1. Curing Control Statement ..... 37
2. Purpose and threat relevance ..... 37
3. Organizational implementation (CISO perspective) ..... 37
4. Technical implementation (for IT professionals) ..... 37
5. Implementation examples ..... 38
6. Maturity path (cumulative) ..... 38
7. Measurement and audit evidence ..... 38
8. Typical mistakes ..... 38
9. Demarcation, residual risk and references ..... 39

**MHC-15 — AI-RESISTANT IDENTITY VERIFICATION OF PERSONNEL AND ACCESS ..... 39**

1. Curing Control Statement ..... 39
2. Purpose and threat relevance ..... 39
3. Organizational implementation (CISO perspective) ..... 40
4. Technical implementation (for IT professionals) ..... 40
5. Implementation examples ..... 40
6. Maturity path (cumulative) ..... 41
7. Measurement and audit evidence ..... 41
8. Typical mistakes ..... 41
9. Demarcation, residual risk and references ..... 41

**MHC-16 — MULTI-AGENT INTEGRITY ..... 42**

1. Curing Control Statement ..... 42
2. Purpose and threat relevance ..... 42
3. Organizational implementation (CISO perspective) ..... 43
4. Technical implementation (for IT professionals) ..... 43
5. Implementation examples ..... 43
6. Maturity path (cumulative) ..... 44
7. Measurement and audit evidence ..... 44
8. Typical mistakes ..... 44
9. Demarcation, residual risk and references ..... 45

**SUPPLEMENTARY INDIVIDUAL MEASURES ..... 46**

**GLOSSARY ..... 47**

**MAPPING: MHC ↔ MRIS-TISAX® 2027, PART I ..... 48**

# Purpose and conventions

Part I diagnoses 72 individual requirements across 31 controls, bundled into eight recurring patterns (phase 2) and a gap analysis against six external frames of comparison (phase 3); this document provides the implementation guideline for each of the nine MHCs (Phase 4) synthesized from it in comparable depth — purpose, organizational and technical implementation, maturity level and evidence — so that a CISO function can understand it without additional specialized literature. It does not replace Part I, but carries out its diagnosis in practice. A tenth MHC (MHC-16) is added, which is not derived from a TISAX® requirement, but represents an original supplement that goes beyond TISAX® (see note on MHC numbering and Part I).

**Time reference:** Like Part I, this Implementation Guide also reflects the state of knowledge in July 2026. The MHC entries are calibrated against the attacker capabilities proven at that time; new AI capabilities or updated external frameworks (NIST, BSI, DORA, CRA, NIS2, ISA/IEC 62443) may shift individual maturity paths or prioritizations in the future.

**Note on MHC numbering:** Mythos Hardening Controls (MHC) are a cross-standard measure vocabulary — the same number denotes the same curing measure, regardless of the underlying standard. Seven of the ten MHCs documented here (MHC-03, MHC-05, MHC-08, MHC-09, MHC-10, MHC-11, MHC-13) deliberately have the same number and the same basic title as in the MRIS Implementation Guide 1.2 for ISO 27002, because the TISAX® diagnosis from Part I reveals the same structural gap. The entries here are calibrated to what TISAX® ISA 2027 specifically diagnoses; where IG 1.2 describes a technically deeper expansion of the same control (e.g. policy-as-code for MHC-10, pipeline security testing for MHC-09), reference is made to it. MHC-13 has been a special feature since this version: IG 1.2 already documents a much broader measure (including hardening against prompt injection, harness-as-code) under the same number than the core originally calibrated here, which was closely tailored to Control 1.3.3/1.3.4; this entry has therefore been extended to keep the same number consistent across both documents — with reference to IG 1.2 for the additional technical depth available there (see MHC-13, section 9). Three MHCs (MHC-14, MHC-15, MHC-16) have been newly assigned: For the underlying topics — supplier verification verification, identity verification of personnel/access and multi-agent integrity — there was no equivalent among the 13 MHCs of IG 1.2. According to Part I, Phase 3, MHC-14 and MHC-15 are original gaps that have not yet been solved throughout the industry, each of which is derived from a specific TISAX® requirement finding. MHC-16 occupies a special position: It is not derived from a TISAX® requirement finding, but an independent supplement that goes beyond the TISAX® diagnosis — for details and justification, see Part I, section "Original Additions beyond TISAX®", and MHC-16, Section 9. The other MHCs of IG 1.2 (including MHC-01 Post-Quantum Strategy, MHC-02 SBOM/Build Provenance, MHC-04 Workload Identity, MHC-06 Container Security, MHC-07 Multi-Tenancy Isolation, MHC-12 Threat-Led Penetration Testing) cover topics for which the TISAX® ISA-2027 diagnosis from Part I did not provide its own downgraded findings, and therefore do not appear here. The document affiliation always remains decisive for the assignment of an MHC.

## Fixed structure per MHC

Each MHC entry follows the same nine-part structure:

- Head »At a glance«
- 1. Curing Control Statement
- 2. Purpose and threat relevance
- 3. Organizational Implementation (CISO)
- 4. Technical implementation (concludes with efficacy test)
- 5. Implementation examples (example A/B)
- 6. Maturity Path
- 7. Measurement and audit evidence
- 8. Typical mistakes
- 9. Definition, Residual Risk and References.

The sections "Measurement and Audit Evidence" are deliberately kept lean — practicable evidence instead of excessive documentation burden. Organizations with a higher level of audit regulation or customer requirements can add an extended audit grid for each MHC:

- Control Owner,
- Scope,
- Applicability rationale,
- Implementation Status,
- Evidence Type,
- Test Frequency,
- Sampling Logic,
- Exceptions,
- Risk Acceptance,
- KPI/KRI,
- Last Effectiveness Review.

This grid is optional, not mandatory part of this guide.

The maturity path per MHC (Initial/Defined/Managed) is an independent, cumulative implementation measure and independent of the TISAX® maturity model (0–5, target value 3) from Part I — both are compatible with each other, but measure different things: **TISAX® evaluates the process of the organization, the MHC maturity path the implementation depth of the respective additional measures.**

# Prioritizing MHC — Threat Correlation and Roadmap

The ten MHCs documented here do not follow any internal ranking, and their numbers are deliberately not sequential — unlike in a self-contained list (see note on MHC numbering above). It is not the control itself that is prioritized, but its contribution to reducing the threats most relevant to the organization. This chapter combines two complementary views: the measurable range of impact of each MHC (how many of the 72 individual findings identified in Part I it carries) and the threat and effort-related order (threat correlation).

**At a glance**

- Purpose: comprehensible, threat-oriented implementation roadmap instead of linear MHC order.
- Basis: Coverage per MHC plus Threat Priority Score.
- Result: Standard tiering P0/P1/P2 as a customizable template, with dependency rules.

## 1. Scope of action: how many individual findings each MHC carries

The figures are checked against the cluster overview in Part I, Phase 2 (72 individual findings across 31 controls, of which 1 Pure Friction — the pure password factor in Control 4.1.2; classic MFA procedures within the same and other controls count as partially demoted, see Part I, Chapter 4). Each finding is supported by exactly one MHC or individual measure; Multiple afflictions arise at the control level (a control can contain findings from several clusters), not at the findings level. MHC-16 is a deliberate exception: it does not carry any of the 72 individual findings because it is not derived from the TISAX® diagnosis, but represents an original supplement that goes beyond TISAX® (see Part I, section "Original Supplements beyond TISAX®").

| MHC | Individual findings | Controls | Character |
|---|---|---|---|
| MHC-10 Continuous Control Monitoring and Policy-as-Code | 18 | 12 | Broad, cross-process |
| MHC-03 Phishing-Resistant Multi-Factor Authentication | 10 (of which 1 pure friction) | 6 | Identity/Access |
| MHC-05 Behavior-Based Detection and Kill Chain Correlation | 9 | 5 | Detection/Correlation |
| MHC-09 AI-Powered Security Testing in the Pipeline | 6 | 3 | Vulnerability/Patch Management |
| MHC-15 AI-Resistant Identity Verification Staff/Access | 6 | 3 | Personnel/Facility |
| MHC-14 Independent Supplier Verification | 5 | 2 | Supply chain/contract |
| MHC-11 SOAR-based Tier 1 Automation | 4 | 1 | Incident Response |

| MHC | Individual findings | Controls | Character |
|---|---|---|---|
| MHC-13 AI Agent Governance and Harness Security | 4 | 2 | AI Agent Governance |
| MHC-08 Immutable Backups and Recovery Validation | 3 | 1 | Backup/Recovery |
| Individual measures (6, see below) | 7 | 6 | without a recurring pattern |
| MHC-16 Multi-Agent Integrity | 0 (original) | 0 | AI Agent Governance, Forward-Looking |

MHC-10 is by far the broadest cross-sectional control — the strongest argument against the perception that every MHC is equally important.

## 2. Two Priority Views and Their Relationship

Coverage is the unconditional leverage effect — how many individual findings an MHC carries. Threat and effort reference is the actual order — depending on threat level, impact, exposure, control gap, dependencies, and effort. Both views complement each other: Coverage alone would put MHC-13 at the back because it only flanks two controls — that would be wrong, because MHC-13 closes the only gap that none of the six comparison frames from Phase 3 solves (see prioritization below). "Low hanging fruits" are controls with high leverage, large control gap and low effort, taking into account the dependencies.

## 3. Evaluation logic: Threat Priority Score

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

| Factor | Guiding question | Scale |
|---|---|---|
| Threat relevance | How realistic, up-to-date and relevant is the attack scenario for the organization? | 1 = low ... 5 = very high |
| Business Impact | How serious would the consequences be for TISAX® scope processes, customer data, supply chain, reputation? | 1 = low ... 5 = existential |
| Exposure | How exposed is the organization (internet-facing, remote access, privileged accounts, supplier network)? | 1 = low ... 5 = highly exposed |
| Control Gap | How big is the gap compared to the MHC target picture? | 1 = well covered ... 5 = barely covered |

Example: Real-time phishing relay (AITM) against privileged remote access with threat relevance 5, Business Impact 4, Exposure 5, and Control Gap 4 results in a score of 400 — very high priority, primarily for MHC-03, flanked by MHC-11.

The Threat Priority Score is a qualitative prioritization index, not a substitute for a formal risk analysis. Multiplication makes relative priorities for action visible; the value is ordinal, not metric — a score of 400 is not "twice as urgent" as 200, but is used for ranking and triage.

## 4. Threat-to-MHC matrix

| Threat/Attack Scenario | Primary MHC | Justification | Priority |
|---|---|---|---|
| Real-time phishing relay (AITM) against authentication | MHC-03, MHC-11 | MHC-03 makes intercepted factors worthless, MHC-11 mitigates if an account is compromised. | very high |
| Patch Gap Collapse / Ransomware After Exploit | MHC-09, MHC-08, MHC-05, MHC-11 | MHC-09 shortens the time window to remediation, MHC-08 ensures recoverability even after compromise, MHC-05 detects exploitation based on behavior, MHC-11 automatically contains. | very high |
| Deepfake identity spoofing (setting, access, calling) | MHC-15 | Directly addresses AI-supported document/identity forgery in the personnel and access process. | high |
| Fake/embellished supplier certificates | MHC-14 | Enforces independent verification instead of pure self-disclosure. | medium to high |
| Autonomous AI agent integrates unauthorized tool/package | MHC-13 | The only MHC against a threat type that none of the six comparison frames from Phase 3 has regulated so far. | high, rising |
| Prompt Injection hijacks agent action objective (Goal Hijack) | MHC-13 | According to OWASP, the most prevalent threat category for AI agents (ASI01); MHC-13 requires channel separation, input/output filtering, and a pen test before going live. | high, rising |
| Compromised or taken over agent cannot be isolated in a timely manner | MHC-13 | Without a kill switch, the withdrawal of an agent ties up a broad system intervention; MHC-13 sets a target value of less than 5 minutes. | high |
| Unnoticed changed behavior of an already released agent component (rug pull) | MHC-13 | One-time approval does not protect against subsequent changes; MHC-13 requires retesting with each update. | medium to high |

| Threat/Attack Scenario | Primary MHC | Justification | Priority |
|---|---|---|---|
| Fragmented, low-and-slow attack | MHC-05, MHC-10 | MHC-05 correlates across time and sources, MHC-10 prevents control drift from going undetected between periodic test cycles. | high |
| Outdated/non-event-related controls in general | MHC-10 | Widest range of impact; affects practically every other cluster indirectly. | high |
| Multiple communicating agents: memory/context poisoning, cascading effects | MHC-16 | Requires a multi-agent architecture; increasing relevance. | medium, future-oriented |

## 5. Dependencies as roadmap rules

| Dependency | Significance |
|---|---|
| MHC-10 → MHC-13 | An allow list for AI agent tools fizzles out if it is never sharpened on an ad hoc basis as soon as new attack patterns become known. |
| MHC-03 → MHC-11 | Phishing-resistant authentication prevents a large proportion of incidents that MHC-11 would otherwise have to contain automatically. |
| MHC-09 → MHC-11 | Automated containment is only as fast as the upstream vulnerability detection on which alarms are based. |
| MHC-09, MHC-08 complement each other | Both address ransomware resilience from different directions — MHC-09 shortens the time window to remediation, MHC-08 ensures recoverability if exploitation succeeds; can be implemented independently of each other. |
| MHC-14, MHC-15 independent | Different responsibilities (purchasing vs. HR/plant security) — can be implemented in parallel, no sequencing necessary. |
| MHC-13 → MHC-16 | Multi-agent integrity requires a functioning single-agent safeguard (identity, allow-list, kill switch) before communication between agents can be meaningfully hardened. |

## 6. Standard prioritization P0 / P1 / P2

The following tiering is a starting point, not a universal implementation plan — it needs to be adapted to each organization's exposure, architecture, asset criticality, and resource location.

| Priority | MHC | Justification |
|---|---|---|
| P0 — immediately | MHC-03 | Closes the only pure friction finding of the entire analysis. |
| P0 — immediately | MHC-08 | Technically clearly defined, established measure (Object Lock/WORM) with low implementation effort and high leverage |

| Priority | MHC | Justification |
|---|---|---|
| | | against ransomware; unlike MHC-09, C5:2026 does not formulate a binding immutability requirement here — the urgency results from the threat situation, not from an already existing external standard. |
| P0 — immediately | MHC-09 | Patch gap closure: C5:2026 already requires a defined, CVSS-bound deadline framework (OPS-25.01AC) — low implementation effort, high leverage. |
| P0 — immediately | MHC-13 | Largest independent gap — no external framework already provides a solution, urgency increases with every day of unregulated AI agent deployment. |
| P1 — after | MHC-10 | Widest range of impact, but higher change management effort across 12 controls. |
| P1 — after | MHC-14 | Contract clause addition comparatively simple, but only takes full effect with the next review cycle. |
| P1 — after | MHC-11 | Builds on MHC-09/MHC-03; more effective if they are already taking effect. |
| P2 — risk-based | MHC-15 | High effort (process change HR/plant security), original new development without external template. |
| P2 — risk-based | MHC-05 | The most technically demanding measure (product selection EDR/XDR, development of UEBA correlation) requires an existing SIEM basis in some cases. |
| P2 — risk-based | MHC-16 | Original and future-oriented — relevant as soon as several agents communicate productively with each other, MHC-13 presupposes. |

## 7. Four-step approach

- **Step 1 — Create a threat landscape:** identify relevant attacker types and attack scenarios for your own TISAX® scope environment; Supply chain, remote access, and AI agent exposure.
- **Step 2 — Map threats with MHCs:** Determine which MHCs have a preventive, detective, reactive or validating effect for each threat; clearly justify and defer non-applicable MHCs (e.g. MHC-13 without the use of AI agents).
- **Step 3 — Calculate score:** Evaluate threat relevance, business impact, exposure, and control gap from 1 to 5 each; Calculate score, take effort and dependencies into account.
- **Step 4 — Derive roadmap:** P0 (immediately against high threats and large gaps), P1 (controls with dependencies or basic work), P2 (contextual); reassess regularly as the threat situation or AI agent deployment changes.

### 8. Wording for governance documents

*For risk-based prioritization of the TISAX® curing controls, the MHCs are correlated with relevant threat scenarios. For each threat, it is assessed which controls have* ***a preventive, detective, reactive*** *or* ***validating*** *effect.* ***Prioritization is based on threat relevance, business impact, exposure, existing control gap, dependencies, and implementation effort.*** *This does not create a linear MHC sequence, but a threat-oriented roadmap that justifies why each control is implemented at its time.*

## MHC-03 — Phishing-Resistant Multi-Factor Authentication

**At a glance**

- Operational benefit: Makes intercepted or real-time credentials worthless, removing the foundation for automated, AI-scaled phishing.
- Affected Policy: Access Control and Authentication Policy.
- Dependencies: no precondition; works together with MHC-11.
- Effort drivers: Number of users/services/legacy systems, existing identity platform.
- Primary metric: Proportion of privileged and externally accessible access with phishing-resistant authentication.
- Affected Standards (without Claim for Fulfilment): NIS2 Art. 21(2)(j); NIST SP 800-63B (AAL3); TISAX® ISA 2027 4.1.1, 4.1.2, 4.1.3, 5.1.2, 5.2.7, 6.1.3.

### 1. Curing Control Statement

For authentication procedures that TISAX® requires as "state of the art" or as a second factor, a phishing-resistant procedure (FIDO2/WebAuthn, PKI-based certificates) is mandatory at least for privileged accounts and remote access. Pure password authentication without an additional factor is not sufficient for any protection requirement level. Access authorizations are automatically and immediately blocked in the event of loss or compromise of an ID card.

### 2. Purpose and threat relevance

Automated real-time phishing relays (adversary-in-the-middle, AITM) undermine OTP and push-based methods, which are still considered "state of the art" in many places: the attacker forwards the login in real time via a fake page and receives the valid factor directly from the user. With pure password authentication, even this detour is omitted — the attacker receives the correct password directly, not by guessing (Part I classifies this as the only pure friction finding of the entire analysis, Control 4.1.2). Phishing-resistant methods cryptographically bind the proof of registration to the real address of the service; on a fake page, the proof is worthless. The goal of protection is to ensure that intercepted or captured login data is of no use to the attacker. This cryptographic binding marks the difference between "weakened"

(classic OTP/push MFA — partially demoted in Part I) and "holds" (phishing-resistant practices — the steadfast target state that this MHC prescribes).

## 3. Organizational implementation (CISO perspective)

- **Policy:** The authentication guideline prescribes phishing-resistant procedures for privileged, externally accessible and particularly sensitive accesses; pure password authentication as well as SMS/simple push procedures are excluded for new accesses.
- **Responsibilities:** IAM team is responsible for rollout and replacement of old processes according to a fixed transition plan.
- **Risk analysis:** Target maturity level is derived from the need for protection — privileged and externally accessible access first.
- **Processes:** regulated issuance and redemption of tokens/passkeys; Blocking process in case of loss is automated, not only implemented "as far as possible"; Replacement procedure for loss cases without undermining the protection level.

## 4. Technical implementation (for IT professionals)

- **FIDO2/WebAuthn rollout:** Introduction for privileged accounts first, then remote access, then other externally accessible services.
- **Automated blocking:** technical connection so that a reported loss of an ID card blocks the authorization immediately and automatically, not only after manual processing.
- **Context/anomaly protection:** where phishing-resistant methods are not (yet) possible across the board, additional device binding and continuous access monitoring as compensation.
- **Legacy process shutdown:** defined end date for OTP/push-only procedures after migration phase.
- **Effectiveness test:** A simulated AITM (controlled environment) phishing test against an account with phishing-resistant authentication must fail, while the same test against an account with classic OTP would be proven to succeed — the difference makes the effectiveness visible.

## 5. Implementation examples

**Example A — Organization with its own identity platform.** A manufacturer with a central identity provider rolls out FIDO2 security keys first for all administrative and remote access accounts, then for all externally accessible applications. Blocking in the event of loss is automated via the identity platform and takes effect in real time.

**Example B — Organization with many legacy and third-party systems.** An organization with a heterogeneous, partly historically grown system landscape cannot introduce FIDO2 everywhere. It prioritizes the transition to privileged and externally accessible access and compensates for remaining legacy systems with additional context monitoring (device binding,

location/anomaly check) until a complete migration is possible — documented as a temporary compensation measure, not as a permanent state.

## 6. Maturity path (cumulative)

- **Initial:** phishing-resistant procedure for privileged accounts piloted.
- **Defined:** phishing-resistant procedure for all privileged and remote access accounts productive; automated blocking in case of loss.
- **Managed:** nationwide for all externally accessible services; Old process completely switched off.

## 7. Measurement and audit evidence

- **Metric:** Proportion of privileged and externally accessible access with phishing-resistant authentication (target value: 100% for privileged accounts).
- **Proof:** Authentication policy, IAM configuration excerpt, lockout logs, migration plan with end date for legacy procedures.
- **Audit logic:** documented? — guideline with procedural specification; responsible? — IAM team; Frequency? — Blocking in real time, migration progress checked quarterly; Tolerance? — no privileged access without a phishing-resistant factor without documented compensation.

## 8. Typical mistakes

- "Second factor" is equated with "phishing-resistant" — OTP/push is already mistakenly considered sufficient.
- Blocking in the event of loss is possible, but runs via a manual ticket process instead of automated.
- Rollout starts with standard users instead of privileged/exposed accounts, where the risk is greatest.
- Compensation measures for legacy systems are never provided with an end date and remain in place permanently.

## 9. Demarcation, residual risk and references

- **Definition:** concerns the authentication method; Authorization assignment and revocation is governed by the respective IAM process document, not this MHC.
- **Residual risk:** Session hijacking after successful authentication is not prevented by phishing-resistant login — additional session monitoring is required for this.
- **TISAX-Controls®:** 4.1.1, 4.1.2, 4.1.3, 5.1.2, 5.2.7, 6.1.3 (see Part I; 4.1.2 there as the only pure friction finding).
- **Framework:** NIS2 Art. 21(2)(j) (first EU legal standard that mentions MFA or Continuous Authentication by name as a suitable measure in the risk-based catalogue of measures, formulated there "where appropriate" instead of as an absolute obligation); NIST SP 800-63B (AAL3, phishing-resistant authenticators).

# MHC-05 — Behavior-Based Detection and Kill Chain Correlation

**At a glance**

- Operational benefit: Detects both unknown, signature-less malware based on its behavior and attacks that are deliberately broken down into many small, inconspicuous steps — in their sum instead of step by step.
- Affected policy: Malware Protection/Endpoint Security Policy; Reporting, log evaluation and crisis detection guidelines.
- Dependencies: no precondition; The endpoint and correlation levels share a central log aggregation base and reinforce each other.
- Effort drivers: Product selection (EDR/XDR), integration into existing endpoint and SIEM landscape, tuning to avoid false alarms, maturity of the existing log infrastructure.
- Primary metric: Proportion of systems with a need for protection high/very high that are covered by a behavior-based component (EDR/XDR); Proportion of security-relevant log sources in time-spanning correlation.
- Affected standards (without performance claim): ISA/IEC 62443-2-1:2024 (behavioral analytics/continuous monitoring mentioned as an extension); NIST CSF 2.0 (DE. AE); TISAX® ISA 2027 5.2.3, 1.6.1, 1.6.3, 5.2.4, 5.2.7.

## 1. Curing Control Statement

Malware protection is not based solely on signature-based detection: for systems with increased protection needs, a behavior- or anomaly-based detection component (EDR/XDR class) is added. At the same time, reporting and log evaluation processes are not based exclusively on individually conspicuous events: A correlation of events over time and across several sources (e.g. according to the UEBA approach) deliberately detects fragmented, inconspicuous attack steps in their totality. Both layers — endpoint behavior and cross-source correlation — converge on the same central log aggregation basis. Crisis detection and escalation procedures are explicitly supplemented by the scenario of a fast, AI-orchestrated and multi-pronged escalating situation.

## 2. Purpose and threat relevance

Classic, signature-based malware protection is structurally weaker against AI-generated or polymorphic malware, as signatures only take effect after a variant becomes known — with AI-supported variant generation, the next variant emerges faster than new signatures can be distributed. Encrypted content also structurally bypasses gateway inspection, a bypass that AI-powered attackers deliberately use. At the same time, a reporting system that depends on a person noticing something conspicuous does not recognize an attack that is deliberately broken down into many small steps that seem harmless in themselves — not a single step is noticeable. The GTG-1002 campaign uncovered by Anthropic in September 2025 (see Part I, Methodology) first carried out an unobtrusive, "low-and-slow" reconnaissance within normal

usage patterns before the remaining steps followed at machine speed — a pattern that neither pure signature recognition nor log checking designed for human case-by-case conspicuousness coherently captures. Classic crisis detection is also designed for gradually escalating situations, not for an AI-orchestrated, multi-pronged attack with rapid escalation. The aim of protection is to ensure that both unknown malware is detected as a coherent pattern based on its behavior and the sum of inconspicuous individual steps.

## 3. Organizational implementation (CISO perspective)

- **Policy:** The Malware Protection Policy requires a behavior-based component in addition to signature-based detection for high/very high protection needs; the Log Evaluation and Crisis Detection Policy requires cross-source, cross-time correlation in addition to single-event inspection.
- **Responsibilities:** IT operations is responsible for product selection, rollout and tuning of endpoint detection as well as the maintenance of correlation rules; a security operations department (internal or as a managed service) handles alarm processing; CISO function is responsible for expanding the crisis scenarios.
- **Risk analysis:** Prioritization according to protection requirements (systems high/very high first) and according to criticality of the log sources to be integrated.
- **Processes:** defined sequence for processing behavior-based alarms including tuning cycle; Crisis scenario catalogue is supplemented by at least one AI-orchestrated, multi-pronged escalation scenario and is practiced regularly.

## 4. Technical implementation (for IT professionals)

- **EDR/XDR rollout:** Select and roll out a behavior-based detection component on prioritized systems; Establishment of a normal behavior profile for each system class.
- **Encrypted content compensation:** where gateway inspection of encrypted traffic is not possible, endpoint-side control (process/file behavior) as compensation.
- **Centralized log aggregation:** Consolidate security-related logs from multiple sources (endpoint, network, identity, application) into a common reporting platform — the same basis on which EDR/XDR alarms run.
- **UEBA correlation rules and time window correlation:** Normal behavior baselines per user/system; Rules that evaluate not only individual events, but chains of events over defined time windows (hours to days) and condense them into an overall picture.
- **Efficacy test:** Two separate tests. First: a controlled, behaviorally conspicuous, but signature-less test script (comparable to an Atomic Red Team test) must be detected and alerted by behavior alone. Second: a simulated "low-and-slow" test chain (several inconspicuous individual actions, spread over a longer period of time) must be recognized as a coherent pattern and alarmed.

## 5. Implementation examples

**Example A — Organization with Security Operations Team.** A manufacturer with its own SOC rolls out an EDR solution on all systems with a need for protection high/very high and

feeds their alarms into the same central detection platform that also carries the cross-source UEBA correlation. A monthly tuning cycle reduces false positives on both levels. In addition, the company is supplementing its crisis scenario catalog with an AI-orchestrated, multi-pronged escalation scenario and practices it annually against the real chain of detection.

**Example B — Organization without its own SOC.** An organization without its own security operations team obtains both behavior-based endpoint detection and cross-source correlation as a managed detection service with contracted response time for critical alarms, and ensures that all of its own security-related log sources are fully connected to the service.

## 6. Maturity path (cumulative)

- **Initial:** behavior-based component piloted on the most critical systems; central log aggregation established for the most important sources.
- **Defined:** Rollout to all systems with protection requirements high/very high completed; Correlation rules active across multiple sources and time windows; Crisis scenario catalog supplemented by AI-orchestrated escalation.
- **Managed:** Baseline tuning established at both levels, false positive rate low, UEBA baseline tested regularly; Crisis scenario is practiced, not just documented.

## 7. Measurement and audit evidence

- **Key figure:** Proportion of systems with a need for protection high/very high with active behavior-based detection component (target value: full coverage); Proportion of security-relevant log sources in cross-time correlation (target value: all sources with protection requirements high/very high).
- **Evidence:** Rollout documentation, configuration excerpt, correlation rule documentation, results of both effectiveness tests, alarm statistics, updated crisis scenario catalog with exercise protocols.
- **Audit logic:** documented? — policy, rollout plan, correlation rules; responsible? — IT operations/SOC, CISO function for crisis scenarios; Frequency? — ongoing operations, efficacy tests and crisis exercises at least once a year, Tolerance? — no system with a need for protection high/very high and no critical log source without coverage without a documented exception.

## 8. Typical mistakes

- The behavior-based component is installed, but never tuned to its own environment — high false alarm rate causes alarms to be ignored.
- Log sources are collected centrally, but never actually correlated — pure data storage without evaluation logic.
- Both levels run technically separately instead of on a common basis, which means that an attack that remains below the threshold on one level is not compensated for on the other.
- The supplemented crisis scenario remains on paper and is never practiced.

### 9. Demarcation, residual risk and references

- **Demarcation:** concerns endpoint-level detection and cross-source correlation; automated response to a detected incident is governed by MHC-11 and preemptive closure of the underlying vulnerability MHC-09.
- **Residual risk:** behavior-based detection and correlation reduce, but do not eliminate the risk of completely new attacks tailored to one's own baseline and defined correlation thresholds.
- **TISAX-Controls®:** 5.2.3, 1.6.1, 1.6.3, 5.2.4, 5.2.7 (see Part I).
- **Framework:** ISA/IEC 62443-2-1:2024 mentions behavioral analytics and continuous monitoring as an extension, without formulating this as a strict obligation; NIST CSF 2.0 (DE. AE, Anomalies and Events) — predominantly original MHC concretization.

## MHC-08 — Immutable Backups and Recovery Validation

**At a glance**

- Operational benefit: Ensures that data backups remain recoverable even if admin rights are compromised, and that recovery actually works in the event of a disaster instead of just existing on paper.
- Affected policy: Backup concept and continuity planning.
- Dependencies: no precondition; complements MHC-09 as the reactive side of ransomware resilience.
- Expense drivers: existing backup infrastructure, conversion to Object-Lock/WORM-capable storage systems, separation of administrative access paths.
- Primary metric: Success rate of regular recovery tests from immutable backups.
- Affected standards (without performance claim): BSI C5:2026 (OPS-06 to OPS-09); DORA Art. 12; NIST SP 1800-11/1800-25 (Data Integrity, NCCoE); TISAX® ISA 2027 5.2.8.

### 1. Curing Control Statement

Data backups of critical IT services are kept immutable or physically/logically separate where the need for protection justifies it. Administrative deletion or modification rights to the backup repository are separated from the production administration area, so that a compromise of productive admin accounts does not include the backups. Recovery is fully tested on a regular basis, not just the backup itself.

### 2. Purpose and threat relevance

Ransomware with compromised backup administration rights can delete or encrypt existing backups if they are not technically immutable — a "sufficiently protected" backup in the sense of a pure access list does not withstand a sufficiently captivated, AI-powered accelerated attack. TISAX® ISA 2027 explicitly calls for protection against unauthorized

modification/deletion only from the need for protection, and even there without an immutability requirement in the wording ("sufficiently protected"). The goal of protection is to ensure that recovery remains possible even after privileged accounts have been compromised, and that this capability is proven to work in the event of an emergency.

## 3. Organizational implementation (CISO perspective)

- **Policy:** The backup policy prescribes immutability or physical/logical separation for protection requirements high/very high, regardless of whether the TISAX® wording already goes so far at the respective point.
- **Responsibilities:** IT operations is responsible for operating and testing the backup infrastructure; administrative rights to the immutable repository are assigned to a separately managed role.
- **Risk analysis:** prioritization according to criticality of IT services; Systems with a need for protection high/very high were first converted to immutable security.
- **Processes:** regular, complete recovery test with documented result; Escalation on failed test.

## 4. Technical implementation (for IT professionals)

- **Immutable backups:** technical implementation via Object Lock (S3 compatible), WORM storage or physically separate/offline copies.
- **Separate administration:** administrative deletion/modification rights to the backup repository are technically separated from the production administration area (own identity, no penetration of compromised production admin accounts).
- **Restore testing:** regular, complete recovery test (not just sample) with logging of result and time required against a defined RTO.
- **Effectiveness test:** A simulated deletion attempt on a backup marked as immutable after assumed admin compromise must technically fail; in addition, a complete restore test is performed and measured against the defined RTO.

## 5. Implementation examples

**Example A — Organization with its own backup operation.** A manufacturer operates an Object Lock-capable storage system for systems with high/very high protection requirements, whose deletion and modification rights are managed separately from the productive administration accounts. A quarterly, complete recovery test is part of regular operations.

**Example B — Organization with outsourced IT operations.** An organization that has outsourced its IT operations to a service provider requires the service provider to provide proof of the Object Lock (WORM) technology used, as well as regular, documented recovery test results as part of the contract.

## 6. Maturity path (cumulative)

- **Initial:** Backups available; Recovery tests annually.
- **Defined:** 3-2-1-1-0 principle implemented; Recovery tests quarterly.

- **Managed:** immutable backups with separate access paths; automated integrity check.

## 7. Measurement and audit evidence

- **Metric:** Success rate of quarterly recovery tests from immutable backups (target: 100%).
- **Proof:** Logs of recovery tests, proof of immutability (retention and lock settings), separate assignment of rights to the backup infrastructure.
- **Audit logic:** documented? — backup policy and test logs; responsible? — IT operations; Frequency? — quarterly tests, continuous integrity checks, Tolerance? — No business-critical data stock without an immutable or separate copy and without a passed recovery test.

## 8. Typical mistakes

- There are backups, but they can be accessed via the same access points as the productive systems — an attacker encrypts both.
- Backups are created, but the restore is never fully tested; in an emergency, parts are missing or the process takes too long.
- Immutable is configured, but not validated; a blocking period that is too short makes the copy vulnerable.
- Immutability is only implemented for some of the systems, without the need for protection justifying this.

## 9. Demarcation, residual risk and references

- **Definition:** concerns backup and recovery; MHC-09 regulates the speed of remediation of identified vulnerabilities, and MHC-05 regulates the detection of the attack itself. TISAX® ISA 2027 5.2.9 already independently assesses the backup *medium* question as stable in the case of very high protection requirements (see Part I) — MHC-08 closes the independent gap 5.2.8 (continuity planning level, already high from protection requirement).
- **Residual risk:** a copy that is disconnected from the network is naturally less up-to-date; a very slow or rarely tested recovery can still lead to downtime in the event of an emergency.
- **TISAX-Controls®:** 5.2.8 (see Part I; cf. 5.2.9 for the backup medium question, which has already been independently assessed as stable, in the case of very high protection requirements).
- **Framework:** BSI C5:2026 OPS-06 to OPS-09 (requires encryption, RTO/RPO alignment, monitoring and regular restore tests, but no immutability — this gap remains open compared to C5:2026); DORA Art. 12 (sector-specific reference framework financial sector: backup policies, segregated recovery systems, data integrity check during recovery); NIS2 Implementing Regulation No. 4.2 (Backup and Redundancy Management, Annex Section to Art. 21(2)(c) NIS2 Directive); NIST SP

800-34 Rev. 1; NIST SP 1800-11 (ransomware recovery) and 1800-25 (ransomware identification/protection), NCCoE; Industrial practice 3-2-1-1-0.

# MHC-09 — AI-Powered Security Testing in the Pipeline

**At a glance**

- Operational Value: Reduces the time between known vulnerability and remediation to a level that rivals AI-accelerated exploit development.
- Policy concerned: Vulnerability and patch management.
- Dependencies: no precondition; increases the effectiveness of MHC-11; complements MHC-08 as a preventative side of ransomware resilience.
- Effort drivers: Number of systems to be patched, maturity of vulnerability scanning.
- Primary metric: Percentage of critical vulnerabilities that are fixed within the CVSS-bound deadline; in the case of existing software development, additional time from the discovery of an actively exploited vulnerability (KEV) to the rolled out patch.
- Standards affected (without performance claim): BSI C5:2026 (OPS-25, Vulnerability Management Basic Criteria); CRA Art. 14 (reporting deadlines for manufacturers); TISAX® ISA 2027 1.3.4, 5.2.5, 5.2.6.

## 1. Curing Control Statement

Actions to remediate identified vulnerabilities are tied to fixed deadlines staggered according to criticality (e.g. CVSS score), not to vague terms such as "appropriate" or "timely". For protection requirements high/very high, an upper limit applies to critical vulnerabilities in the hourly to daily range. System and service audits that verify the success of the remediation are carried out as frequently as technically justifiable instead of in rigid, long cycles. Where in-house software development takes place, the following applies: security checks run automatically in the development pipeline (code, dependencies, running application), serious findings block the takeover, and AI-generated code is additionally checked as a separate risk category.

## 2. Purpose and threat relevance

A 2026 analysis of 69,159 CVE by Cogent Security shows that the average time from vulnerability disclosure to the first working exploit has decreased from 125.3 days (January 2025) to 0.5 days (April 2026) (see Part I, Methodology). An assessment methodology that calculates that it takes weeks for an exploit to work systematically underestimates the urgency — pure knowledge of the version and patch status says nothing about the speed of response. Where software is developed in-house, AI programming aids regularly generate insecure patterns (built-in access data, missing input checking, outdated building blocks) and thus new vulnerabilities arise faster than they would be found classically before delivery. The goal of protection is to ensure that the speed of remediation — from discovery to rolled out patch, even before delivery in the case of in-house development — keeps up with the speed of the threat.

## 3. Organizational implementation (CISO perspective)

- **Policy:** The Vulnerability Management Policy prescribes CVSS-bound, specific deadlines; for in-house software development, the development policy prescribes automatic security checks in the pipeline as well as separate testing of AI-generated code.
- **Responsibilities:** IT operations are responsible for meeting deadlines and escalating if they are exceeded; in the case of in-house development, AI-generated code is listed as a separate risk category in the Statement of Applicability.
- **Risk analysis:** deadlines are staggered according to criticality and exposure (externally accessible systems stricter than internal ones); Thresholds for blocking pipeline detections and the handling of actively exploited vulnerabilities (KEV) are defined.
- **Processes:** Escalation chain in the event of imminent exceeding of the deadline; documented exemption with risk acceptance where a patch is technically not possible on time (e.g. legacy systems); in the case of in-house development, an additional accelerated path for KEV cases.

## 4. Technical implementation (for IT professionals)

- **CVSS integration:** Vulnerability scanner automatically delivers the CVSS score per find; coupled with an automatic deadline assignment in the ticket system, with automatic escalation run in the event of imminent or actual deadline violation.
- **Regular system/service audits:** Vulnerability scans run as frequently as technically justifiable, not just on fixed, infrequent dates — especially for externally accessible systems.
- **In the case of in-house software development, additionally:** SAST/DAST/SCA test classes automatically for every change in the pipeline; Pre-merge block for serious detections from defined reliability; actively exploited vulnerabilities (KEV) block immediately; additional check rules against typical weaknesses of AI-generated code; daily scans of the systems already running, not just the new code.
- **Efficacy test:** A test vulnerability with a defined CVSS value is injected; the time until automatic ticket creation with the correct deadline is measured. In the case of in-house development, additionally: In a test branch, a known unsafe spot is deliberately included; the pipeline must report the discovery and block the takeover.
- **Limit of effectiveness:** automated testing does not replace technical evaluation and in particular overlooks logic, authorization and architecture weaknesses; critical findings and accepted exceptions require technical triage and documented approval.

## 5. Implementation examples

**Example A — Organization with its own software development.** A software vendor automatically links the CVSS score of each scan fund to a deadline matrix in the ticket system and also builds security checks directly into the development pipeline: code, dependencies and running application are automatically checked for every change, serious detections prevent the takeover, and there is an accelerated path for actively exploited vulnerabilities. Since the development uses AI programming aids, the company adds test rules against their typical errors and evaluates these findings separately.

**Example B — Organization with outsourced IT operations, without in-house development.** An organization without its own software development includes the CVSS-bound deadlines as a contractual SLA in the service agreement with its IT service provider and has compliance reported on a monthly basis, including proof of regular vulnerability scans.

## 6. Maturity path (cumulative)

- **Initial:** CVSS score is recorded, deadlines are documented, but manually tracked.
- **Defined:** Deadline assignment automated in the ticket system; escalation chain defined and active; in the case of in-house development: SAST/DAST/SCA in the pipeline with blocking in case of serious finds.
- **Managed:** Compliance with deadlines is continuously measured and reported; in the case of in-house development: separate evaluation of AI-generated code, daily scans of the running systems, accelerated path for KEV cases.

## 7. Measurement and audit evidence

- **Key figure:** Proportion of critical vulnerabilities that were fixed within the defined deadline (target value: close to 100% with no documented exception); in the case of in-house development, additional time from the time an actively exploited vulnerability (KEV) became known to the patch was rolled out (guideline: less than 24 hours).
- **Proof:** Deadline matrix, scan reports with CVSS score and remediation date, escalation logs; if developed in-house, also pipeline configuration and blocking logs.
- **Test logic:** documented? — deadline matrix according to CVSS; responsible? — IT operations, in the case of in-house development additionally development/product managers; Frequency? — ongoing, with its own development at each change, Tolerance? — Exceeding the deadline only with documented risk acceptance, no delivery with open serious finds.

## 8. Typical mistakes

- The deadline is calculated from the date of the scan instead of when the vulnerability is actually known, which means that delayed scans artificially extend the deadline.
- Exceptions to the deadline are granted informally instead of with documented risk acceptance.
- If developed in-house: the pipeline checks are running, but do not block — serious detections are reported and still delivered.
- In the case of in-house development: only the new code is checked, not the systems already running.

## 9. Demarcation, residual risk and references

- **Differentiation:** refers to the speed of remediation, in the case of in-house development also the automated check during development; the detection of an attack or exploitation itself is regulated by MHC-05, and the recoverability in the MHC-08 ransomware case.
- **Residual risk:** a deadline shortens the time window, but does not close it to zero; Zero-day vulnerabilities without an available patch remain a residual risk regardless of the deadline; automated pipeline checks find known patterns, not every novel or logical vulnerability.
- **Expansion stage:** This entry covers the deadline and scan cadence gap specifically diagnosed by TISAX® ISA 2027. Full pipeline integration (SAST/DAST/SCA as a pre-merge block, AI code as a separate check category) is the decisive expansion for organizations with their own software development; Details can be found in the MRIS Implementation Guide 1.2 for ISO 27002, MHC-09.
- **TISAX-Controls®:** 1.3.4, 5.2.5, 5.2.6 (see Part I).
- **Framework:** BSI C5:2026 OPS-25.01AC (requires a defined, monitored, CVSS-bound deadline framework; cited as an example in the criteria catalog: critical 9.0–10.0 = 24–48 hours, high 7.0–8.9 = 48–72 hours, medium 4.0–6.9 = 5 days, low 0.1–3.9 = 1 month) and OPS-25.01AS (tightening to daily scans); CRA Art. 14 (24-hour notification of actively exploited vulnerabilities to ENISA for manufacturers, binding from 11 September 2026); NIST SP 800-218 (SSDF) and SP 800-218A (KI) for the pipeline expansion stage.

# MHC-10 — Continuous Control Monitoring and Policy-as-Code

**At a glance**

- Operational benefit: Closes the window between two periodic audit cycles in which new attack patterns, incidents or system changes would otherwise go undetected until the next cycle.
- Policy concerned: ISMS framework (Review, Audit, and Review Processes organization-wide).
- Dependencies: no precondition; increases the effectiveness of MHC-13.
- Effort drivers: Number of affected review processes (12 controls) and maturity of existing GRC/ticketing tools.
- Primary KPI: Proportion of safety-relevant test processes with a defined event-related trigger mechanism.
- Affected standards (without performance claim): NIST CSF 2.0 (DE. CM); NIS2 Art. 21(2)(f); TISAX® ISA 2027 1.1.1, 1.2.1, 1.3.1, 1.3.3, 1.3.4, 1.5.1, 1.6.3, 4.1.3, 4.2.1, 5.2.9, 6.1.2, 6.1.3.

## 1. Curing Control Statement

For safety-relevant testing, review and audit processes, an event-related triggering mechanism will be introduced in addition to the regular test cycle. Triggers include at least: newly disclosed, relevant attack patterns or threat intelligence indications; security-related incidents in one's own or comparable environment; significant changes to the affected system or process landscape. For test objects requiring high/very high protection, a fixed maximum cycle length also applies instead of "regular"/"periodic".

## 2. Purpose and threat relevance

TISAX® requirements with a pure time cycle ("regular", "periodic") assume that the threat situation does not change significantly within this cycle. This assumption no longer holds true against a Gen AI-accelerated attacker: new attack patterns, tools, and vulnerability exploits spread within days to weeks, not the next annual or quarterly cycle. A control that is only followed up at the next cycle remains at the old level in the meantime — exactly the time window that an automated attacker exploits. The aim of protection is to ensure that safety-relevant testing processes react to new findings as soon as they are available, not only in the next cycle.

## 3. Organizational implementation (CISO perspective)

- **Guideline:** The ISMS framework will be supplemented by a clause that prescribes an event-related trigger mechanism in addition to the control cycle for all testing, review and auditing processes with safety-relevant relevance.
- **Responsibilities:** A central office (usually CISO function) is designated for maintaining the trigger sources (threat intel feed, incident register, change

management); the operational execution of the triggered check remains with the respective responsible department.

- **Risk analysis:** The cycle length per test object is derived from the protection requirement — critical areas (privileged access rights, externally accessible systems) are given the shortest fixed upper limit.
- **Processes:** Defined sequence of how a trigger (threat intel hit, incident, system change) automatically or manually triggers an unscheduled check of the affected controls, including setting a deadline until implementation.

## 4. Technical implementation (for IT professionals)

- **Threat-Intel connection:** Subscription to a relevant feed (BSI situation report/CERT-Bund, industry ISAC, commercial Threat-Intel service); automated keyword/category filtering to your own system landscape.
- **Incident trigger:** Link the ticket system so that an incident classified as security-relevant automatically generates a review task for the affected policies/controls.
- **Change trigger:** Connection to CMDB/asset inventory, so that a significant change in the system landscape (new externally accessible system, new cloud service) automatically triggers an audit obligation.
- **Deadline clock:** each triggered check receives a due date with escalation if it is exceeded.
- **Effectiveness test:** A test trigger is simulated (e.g. a test entry in the Threat Intel feed or a test incident ticket) and checked whether a review task for the affected control is automatically created within the defined period (e.g. 5 business days) — regardless of the next regular cycle.

## 5. Implementation examples

**Example A — Organization with GRC tooling.** A manufacturer with an established GRC system connects a Threat Intel feed via an API, and when a new, relevant threat message arrives, the system automatically generates a review task for all controls linked to the affected topic area with a fixed processing period. The regular annual cycle will remain in place, but will be supplemented by these event-related intermediate audits.

**Example B — Organization without GRC tooling.** A smaller organization without its own GRC system appoints a permanent responsible person who reviews the BSI situation reports and relevant CERT reports on a weekly basis. A hit with reference to your own system landscape is entered with the date in a simple, divided table; from there, the relevant specialist office is informed and the unscheduled inspection is documented with a deadline. No degree of automation, but a traceable, auditable process.

## 6. Maturity path (cumulative)

- **Initial:** Trigger categories defined; manual review of at least one source (e.g. BSI management report).
- **Defined:** All three trigger types (Threat-Intel, Incident, Change) linked to a defined process and deadline.
- **Managed:** triggered automatically (feed connection, ticket trigger, CMDB webhook); Compliance with the processing deadline is continuously measured.

## 7. Measurement and audit evidence

- **Key figure:** Proportion of safety-relevant inspection processes with documented event-related triggering mechanism (target value: all 12 controls affected).
- **Proof:** Supplemented policy texts, list of trigger sources, example tickets of triggered unscheduled checks with processing date.
- **Test logic:** documented? — definition of triggers per control; responsible? — notified body for each type of trigger; Frequency? — ongoing/occasion-related in addition to the regular cycle, Tolerance? — no critical requirement without a defined trigger mechanism.

## 8. Typical mistakes

- The trigger mechanism is documented, but no source is actually observed — the clause only exists on paper.
- All 12 controls receive the same flat cycle length, regardless of the actual protection requirement.
- A trigger creates a task, but without a deadline — the exam fizzles out.
- The regular cycle is replaced by the occasion-related mechanism without replacement instead of supplementing it — if there is no trigger, there is no longer any check at all.

## 9. Demarcation, residual risk and references

- **Differentiation:** concerns the triggering of audit processes; the depth of the content of the respective audit itself remains regulated by the underlying TISAX-Control® or the other MHCs.
- **Residual risk:** an event-related mechanism is only as good as its sources — an attack pattern that doesn't (yet) show up in threat Intel feeds doesn't trigger a trigger.
- **Expansion level:** This entry realizes the organizational-process-related basic level of MHC-10 (event-related triggers on existing test processes). The technically lower level of expansion — Policy-as-Code (e.g., Open Policy Agent/Rego), runtime enforcement in pipeline and operation, and continuous monitoring coverage as a guided metric — is described in detail in the MRIS Implementation Guide 1.2 for ISO 27002, MHC-10.
- **TISAX® controls:** 1.1.1, 1.2.1, 1.3.1, 1.3.3, 1.3.4, 1.5.1, 1.6.3, 4.1.3, 4.2.1, 5.2.9, 6.1.2, 6.1.3 (see Part I for original wording and individual justification).

- **Framework:** NIST CSF 2.0, Detect function (DE. CM); NIS2 Art. 21(2)(f) in conjunction with NIS2-DVO No. 7 (Effectiveness assessment: measurement methodology, accountability, analysis cycle — minimum frequency annually or on an ad hoc basis after material incident).

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

**At a glance**

- Operational benefit: Reduces the response time to a safety-related event to a level that no longer depends solely on human processing speed.
- Affected Policy: Incident Response/Incident Management Policy.
- Dependencies: works most strongly in conjunction with MHC-09 and MHC-03.
- Effort drivers: maturity of the existing automation tools (SOAR or similar), number of defined severity playbooks.
- Primary metric: Mean Time to Containment (MTTC) for defined severity levels.
- Affected standards (without performance claim): DORA (four-stage reporting chain 4h/24h/72h/1 month as de facto automation pressure); TISAX® ISA 2027 1.6.2.

## 1. Curing Control Statement

Fixed, short response periods are provided for the processing of reported safety-relevant events, which are not based solely on human processing speed. For defined severity levels, at least one semi-automated initial containment action (e.g., automated account lockout, network segment isolation) is provided that triggers before a manual processing is completed.

## 2. Purpose and threat relevance

The processing of reported security events classically requires that humans manage to process them at the necessary speed. When an attacker performs most of their operations automatically and without human intervention — as in the GTG-1002 campaign, which executed attack steps with thousands of requests per second (see Part I, Methodology) — human-based processing cannot keep up structurally. Defining a target also does not mean that it can be met against an automated and fast-acting attacker, as long as there is no obligation to automate. The goal of protection is to trigger an initial containment before an attacker can fully exploit his automated speed.

## 3. Organizational implementation (CISO perspective)

- **Policy:** The incident response policy defines severity levels with fixed response deadlines and specifies the severity levels for which automated initial containment is intended.
- **Responsibilities:** IT operations/SOC is responsible for playbook maintenance; a notified body decides on the release and escalation of automated measures.

- **Risk analysis:** Severity rating is based on the potential impact and speed of propagation of the respective event type.
- **Processes:** defined sequence that distinguishes between automated initial containment and subsequent, more detailed manual processing — automation does not replace human processing, but precedes it in time.

## 4. Technical implementation (for IT professionals)

- **Playbook automation:** predefined, automatically triggering initial actions per severity (account lockout, network segment isolation, session termination).
- **Severity trigger:** Connects to the detection platform (see MHC-05) so that an alarm classified as critical automatically triggers the corresponding playbook.
- **Safety net against false triggering:** defined undo option if an automated action was incorrectly triggered.
- **Effectiveness test:** A simulated alarm of a defined severity level is triggered; the time until the automated initial action is measured, checked against the defined deadline.

## 5. Implementation examples

**Example A — Organization with SOAR platform.** A manufacturer with SOAR integration defines an automated playbook for critical alarms (e.g., notification of compromised privileged account) that immediately schedules the affected session and locks the account while opening a human processing ticket in parallel.

**Example B — Organization without a SOAR platform.** An organization without its own automation platform starts with a single, technically simple automation — such as a script that automatically triggers an account lockout from the existing identity platform when a critical alert type is defined — rather than waiting for a full SOAR rollout.

## 6. Maturity path (cumulative)

- **Initial:** Severity levels and reaction times defined, initial action still manual.
- **Defined:** at least one automated initial action for the highest severity productive.
- **Managed:** automated playbooks for multiple severity levels, Mean Time to Containment is continuously measured and optimized.

## 7. Measurement and audit evidence

- **Key figure:** Mean Time to Containment (MTTC) for defined severity levels (target value: according to defined deadline per severity level).
- **Evidence:** Playbook documentation, alarm-to-containment timing measurements, efficacy test result.
- **Audit logic:** documented? — Severity definition with deadlines; responsible? — IT operations/SOC; Frequency? — continuous measurement, efficacy test at least annually, Tolerance? — no critical alarm type without automated initial action without documented justification.

### 8. Typical mistakes

- Automated measures are implemented without the possibility of undoing, which means that false triggers cause greater damage than the original alarm.
- Automation completely replaces human post-processing instead of just preceding it in time — there is no in-depth analysis.
- Playbooks are created once and never updated against new attack patterns.
- Response deadlines are defined but never actually measured, which means that it goes unnoticed whether they are being met.

### 9. Demarcation, residual risk and references

- **Demarcation:** refers to the speed of initial containment; the upstream detection itself is regulated by MHC-05, the preventive closure of vulnerabilities is regulated by MHC-09.
- **Residual risk:** an automated initial measure contains, but does not solve the underlying cause — without subsequent human processing, the risk of re-escalation remains.
- **Expansion Level:** This entry covers Tier 1 containment. The hardening of the automation layer itself — authenticated/signed triggers, full proof of execution, rate limits per playbook, human-in-the-loop release for actions with a high damaging effect — is described in detail in the MRIS Implementation Guide 1.2 for ISO 27002, MHC-11, and applies equally here as soon as several parallel playbooks are productive.
- **TISAX-Controls®:** 1.6.2 (see Part I).
- **Framework:** DORA principle (sector-specific reference framework for the financial sector) — the four-stage reporting chain (initial report within 4 hours after classification or no later than 24 hours after taking note, interim report within 72 hours, final report no later than 1 month after the interim report) effectively enforces a high degree of automation without prescribing it literally.

## MHC-13 — AI Agent Governance and Harness Security

**At a glance**

- Operational benefit: Prevents an AI-supported assistant from independently integrating an untested external tool, service, or software package as part of a task, and additionally limits identity, manipulability through prompt injection, and response time of productively deployed agents.
- Policy concerned: Policy on the release of external IT services and software; AI Agent Usage Policy.
- Dependencies: works more strongly in conjunction with MHC-10 (event-related tightening of the allow list); MHC-16 regulates threats from multiple communicating agents.
- Effort drivers: Number and variety of AI assistants/agents deployed with tool access, maturity of existing software release processes.

- Primary metric: Proportion of AI agent environments with a technically enforced allow list, complete usage log, own agent identity and functional kill switch.
- Affected standards (without performance claim): no comparison framework from phase 3 fully relevant (original); OWASP Top 10 for Agentic Applications (ASI01, ASI02, ASI03, ASI04, ASI05); NIST AI Agent Standards Initiative (under construction, since February 2026); NIST IR 8596 Cyber AI Profile (draft stage, status "Reviewing Comments", not yet final — see Section 2); TISAX® ISA 2027 1.3.3, 1.3.4.

## 1. Curing Control Statement

The use of AI-supported assistants or agents with access to external tools, services, APIs or software package sources requires technical, not only organizational, release control: a binding allow list of permitted tools/sources, complete logging of every tool use including a basis for decision-making, and a release mechanism for new or unknown destinations that cannot be circumvented by the agent himself. Cryptographic signature verification is mandatory for automatically reloaded software packages.

In addition to this TISAX-derived® core, this MHC requires four other original minimum standards that are not derived from a single TISAX® requirement finding: (a) a technical, capability-scoped (limited to the permissions required for the task at hand), and time-limited agent identity rather than inherited or persistent credentials; (b) a basic hardening against prompt injection, in particular the treatment of tool outputs as untrusted input; (c) a kill switch that disables a single agent or integration within a few minutes; (d) a recheck each time a component is updated instead of a one-time release.

## 2. Purpose and threat relevance

The obligation to release external services and software classically requires that a person consciously selects a certain service or package and submits it for review. However, AI assistants with tool access can independently call up or reload external services and software packages as part of a task without a human ever having submitted this specific target for approval. A pure access list without cryptographic signature verification does not withstand a sufficiently capable automated attack. This vulnerability is not a TISAX® peculiarity: Neither NIST AI RMF nor ISO/IEC 42001 nor the EU AI Act contained a reference to "agent" or autonomous AI systems at the time of this analysis; NIST has only just begun to work on a comparable vulnerability with the AI Agent Standards Initiative (launched in February 2026). The goal of protection is that an AI agent cannot integrate tools or packages that no human has ever tested.

The extension described in Section 1(a)–(d) is based on the OWASP Top 10 for Agentic Applications (ASI01–ASI10, as of June 2026): The TISAX-derived® core of this MHC covers ASI02 (Tool Misuse) as well as parts of ASI04 (Agentic Supply Chain) and ASI05 (Unexpected Code Execution). The extension also includes ASI01 (Agent Goal Hijack/Prompt Injection — according to OWASP the most widespread threat category for AI agents, related to six of the ten ASI categories) as well as the identity-related core of ASI03 (Identity & Privilege Abuse — according to OWASP the category with the largest gap between risk severity and

organizational preparation). Threats that require an architecture with multiple communicating agents (ASI06 Memory/Context Poisoning, ASI07 Insecure Inter-Agent Communication, ASI08 Cascading Failures) are addressed separately in MHC-16, as such an architecture is not yet the norm in the TISAX® context at the time of this analysis.

A regulatory reference point is only just emerging and is not yet reliably citable: The NIST Cyber AI Profile (NIST IR 8596, Cybersecurity Framework Profile for Artificial Intelligence) published a first ("preliminary") draft on December 16, 2025; the comment period ended on January 30, 2026, and according to NIST's own project page (NCCoE), the processing status is "Reviewing Comments" at the time of this analysis — a final or even consolidated interim draft was not available. Secondary sources (including OWASP) describe the draft as the first NIST document with an explicit reference to agentic AI security, including requirements for unique agent identities and a limit on the code execution capability of agents; this content-related reference could not be verified against the NIST primary text in the context of this analysis (the available NIST press release itself does not use the terms "agent"/"agentic") and is reproduced here accordingly cautiously, not adopted as a substantiated requirement.

## 3. Organizational implementation (CISO perspective)

- **Policy:** The release policy for external services and software will be supplemented by an explicit regulation for AI agent tool access — technically enforced, not just organizationally ordered.
- **Responsibilities:** Development/engineering and IT operations are jointly responsible for maintaining the allow list; a notified body decides on release requests for new targets; an additional human responsible person is appointed for each productive agent who can trigger the kill switch.
- **Risk analysis:** Prioritization according to the criticality of the systems on which AI agents with tool access are deployed.
- **Processes:** defined, documented release process for new tools/packages with clear deadlines so that productive use is not blocked by sluggish processes; a procedure for rapid deactivation of a single agent (target value less than 5 minutes); a recheck every time an already released component is updated instead of a one-time, permanently valid release.

## 4. Technical implementation (for IT professionals)

- **Allow-listing:** technical enforcement at the network/system level of which external services, APIs, and package sources an AI agent is allowed to reach — not just a documented but unaudited list.
- **Seamless logging:** every tool used by an AI agent is logged, including the basis for the decision (which task, which goal, which result).
- **Unknown target release mechanism:** a new, non-allow-listed target triggers a check task for a human instance instead of being automatically allowed; the agent cannot bypass this mechanism itself.
- **Signature verification for packages:** automatically downloaded software packages are checked against cryptographic signatures before they are executed.

- **Agent identity (NHI governance):** each agent receives a technical, capability-scoped identity with time-limited (ephemeral) credentials per tool call instead of permanent, inherited, or personal credentials; Tool outputs are generally treated as untrusted input (protection against abused permission transfer, "confused deputy").
- **Basic hardening against prompt injection:** separation of instruction and data channels, as far as technically possible; Inbound and outbound filtering for externally supplied content (e-mails, tickets, documents, web content); a test attempt with a prepared input (injection sample) before going live checks whether the agent executes an illegal action contained therein.
- **Kill switch:** a technical way to disable a single agent or integration in a targeted manner without affecting other agents or systems; Target time to deactivation: Less than 5 minutes.
- **Continuous re-evaluation (rug-pull protection):** the release of a component (MCP server, extension, package) does not apply across the board to future updates — every update of an already released component goes through a new signature, authorization and behavior check before it goes into production; automatic, unchecked updates are technically prevented.
- **Efficacy test:** A test attempt in which an AI agent specifically tries to use a tool or package that is not on the allow list must be technically blocked and logged in full — not only prohibited by the directive.

## 5. Implementation examples

**Example A — Organization with in-house development and AI coding assistants.** A software manufacturer whose development teams use AI-supported programming assistants with package reload capability sets up an internal package repository with signature verification: the wizard can only reload packages from this verified repository, an attempt outside is technically blocked and ends up as a release request at a notified authority. Each assistant also receives its own, time-limited tool identity instead of a shared API key; a conspicuous or compromised assistant can be isolated within minutes without affecting the other teams.

**Example B — Organization with AI assistants in the specialist department, without in-house development.** An organization whose departments use AI assistants with access to external web tools (research, data matching) limits tool access to a validated, curated list of external services via a central proxy/gateway configuration and logs each usage centrally instead of allowing each assistant unrestricted Internet access. If a provider updates a connection that has already been released, the new version undergoes the same check as for the initial release before it goes into production; in the event of suspicious behavior, the specialist department can block the access of the assistant concerned centrally and immediately.

## 6. Maturity path (cumulative)

- **Initial:** Allow list documented, enforcement still organizational instead of technical; Agents run predominantly under personal or shared credentials, no kill switch.

- **Defined:** technical enforcement of the allow list active; seamless logging established; Agents run under their own, capability-scoped identity; Basic hardening against prompt injection (channel separation, input/output filtering).
- **Managed:** Release mechanism for new targets productive, signature verification for packages mandatory, regular effectiveness test; Kill switch productive with time to deactivate under 5 minutes; each update to a shared component goes through retesting before it takes effect.

## 7. Measurement and audit evidence

- **Metric:** Proportion of AI agent environments with technically enforced allow list and full usage log (target value: 100% of AI agents with tool access in production); Time to disable a compromised or suspicious agent (target value: less than 5 minutes); Percentage of updates to released components that were re-checked before they take effect (target: 100%).
- **Evidence:** Allow-list configuration, log excerpts, efficacy test and prompt injection test results, documented release decisions, kill switch test log, proof of retesting for component updates.
- **Audit logic:** documented? — Allow list and release process; responsible? — Development/IT operations, designated assignee per agent for the kill switch; Frequency? — ongoing enforcement, efficacy and prompt injection testing at least every six months in view of the rapid development in this area, kill switch testing at least annually, Tolerance? — No productive AI agent deployment with tool access without a technical allow list, own agent identity and functional kill switch.

## 8. Typical mistakes

- The allow list exists only as a document, with no technical enforcement — the agent can effectively bypass it.
- Logging records that a tool was used, but not the basis for the decision — in retrospect, it is not possible to reconstruct why.
- The approval process for new targets is so sluggish that users systematically bypass it to stay productive.
- Packet signature verification is considered unnecessary for internal, "trusted" sources — that's where supply chain attacks often start.
- The release of a component also implicitly applies to all future updates (rug pull risk) — an unnoticed changed tool description or a manipulated update receives the same rights as the originally checked version.
- No quick, targeted takedown path for individual agents — Retiring a compromised agent requires broad system intervention that takes days instead of minutes.
- Agents continue to run under personal or shared credentials because a separate agent identity is deferred as "additional effort without immediate benefit".

### 9. Demarcation, residual risk and references

- **Differentiation:** concerns the technical control over tool/service integration by AI agents, the identity and access basis of individual agents as well as the basic hardening against prompt injection; the content check of a newly released service itself is regulated by the regular release check (Control 1.3.3/1.3.4, Part I); MHC-16 regulates threats that require an architecture with several communicating agents (memory/context poisoning, insecure inter-agent communication, cascading failures).
- **Residual risk:** a sufficiently qualified agent could try to rephrase the task in such a way that an already released tool is misused — this is outside of pure access control and requires additional behavioral monitoring (see MHC-05); Prompt injection is not completely solvable as long as the instruction and data channels cannot be clearly separated from a technical point of view — the hardening prescribed here reduces the probability of success, but does not replace complete prevention; Detection is not prevention.
- **Expansion stage and origin:** The core of this entry (allow list, logging, release mechanism, signature check) is derived from Control 1.3.3/1.3.4 (Part I, Cluster 7). The extension — (a) Agent Identity, (b) Prompt Injection Hardening, (c) Kill Switch, (d) Continuous Reassessment — is original and not derived from a single TISAX® Request Finding (see Part I, Section "Original Additions Beyond TISAX®"). For the full technical depth — in particular harness-as-code with code review obligation and version management, circuit-breaker mechanisms and full maturity levels for software development environments — see MRIS Implementation Guide 1.2 for ISO 27002, MHC-13 (identical numbering; there with a stronger focus on development environments, here on the TISAX® core plus the minimum standards a–d applicable to each form of deployment).
- **TISAX-Controls®:** 1.3.3, 1.3.4 (Core Scope, Part I); the extensions (a)–(d) are original and not assigned to any individual TISAX® requirement findings.
- **Framework:** original — universal blind spot that none of the six frames of comparison from Phase 3 has closed so far; NIST AI Agent Standards Initiative (February 2026, Interoperability Profile Q4 2026 at the earliest) and NIST IR 8596 Cyber AI Profile (draft stage, status "Reviewing Comments") as the only known, not yet finalized approaches; OWASP Top 10 for Agentic Applications as a threat taxonomy, not a compliance framework.

## MHC-14 — Independent Verification of Supplier Evidence

**At a glance**

- Operational benefit: Prevents an AI-supported convincingly falsified or embellished proof of supplier from being accepted as sufficient undetected.
- Policy concerned: Supplier Management and Procurement Policy.
- Dependencies: no precondition; can be implemented independently of MHC-15.

- Expense drivers: Number of critical suppliers, contract terms (new contracts vs. existing contracts).
- Primary indicator: Proportion of critical suppliers with exercised audit/audit rights in the current cycle, not only contractually provided for.
- Affected standards (without performance claim): DORA Art. 30(3); TISAX® ISA 2027 6.1.1, 6.1.3.

## 1. Curing Control Statement

Safety certificates, certifications and self-disclosures submitted by suppliers or cooperation partners do not replace the company's own examination. For suppliers with access to information requiring protection, an independent right of inspection or auditing is contractually anchored at least for protection needs high/very high — directly or via a commissioned third party — and is actually exercised in the agreed audit cycle.

## 2. Purpose and threat relevance

The audit of suppliers is mainly based on documents submitted by the supplier itself. AI tools make it easier to create convincingly falsified or embellished reports and evidence without requiring independent cross-checking — pure self-disclosure or attestation is sufficient in the TISAX® standard formulation. The aim of protection is to ensure that falsified or embellished evidence is not accepted as sufficient without being discovered.

## 3. Organizational implementation (CISO perspective)

- **Policy:** The Supplier Management Directive prescribes an independent right of inspection/audit for critical suppliers that goes beyond the receipt of certificates.
- **Responsibilities:** Purchasing/Supplier Management is responsible for the contractual clause and scheduling; a designated inspection body conducts the audit or commissions a third party.
- **Risk analysis:** Suppliers are prioritized according to the depth of access to information in need of protection; the most intensive checks are for the most critical suppliers.
- **Processes:** fixed audit cycle with scheduling, escalation in the event of non-compliance with the audit right by the supplier.
- **Procurement:** the audit clause is included as a standard part of new supplier contracts.

## 4. Technical implementation (for IT professionals)

- **Contract clause template:** standardised formulation of the auditing law for inclusion in new existing contracts and existing contracts to be negotiated.
- **Sampling approach:** in the case of a large number of suppliers, sampling logic according to criticality instead of full testing of all suppliers at the same time.
- **Evidence storage:** central, versioned documentation of audits carried out, including date, scope and result.

- **Effectiveness test:** For a sample of critical supplier contracts, it is checked whether the audit clause exists AND whether an audit or review has actually taken place in the current audit cycle — not just contractually envisaged.

## 5. Implementation examples

**Example A — Organization with its own supplier audit team.** A manufacturer with a dedicated third-party risk team conducts its own on-site or remote audit of the most critical suppliers on an annual basis, in addition to certificates submitted. Deviations between the self-disclosure and the audit result are documented and are included in the supplier's next risk classification.

**Example B — Organization without its own audit capacity.** A smaller organization without its own audit team commissions an external audit service provider for the most important suppliers with a focused review instead of a full audit, focusing on the most relevant control areas for cooperation. The result flows into the same supplier file as the certificates submitted.

## 6. Maturity path (cumulative)

- **Initial:** Audit clause included in contract template for new contracts.
- **Defined:** Audit clause for all critical existing contracts renegotiated; Test cycle scheduled.
- **Managed:** Audits are demonstrably carried out in the agreed cycle; Results flow back into the supplier risk classification.

## 7. Measurement and audit evidence

- **Metric:** Percentage of critical suppliers with actual audit in the current cycle (target: 100% of suppliers classified as critical).
- **Evidence:** Contract clauses, audit/review reports with date, escalation logs in case of denied audit performance.
- **Audit logic:** documented? — clause in the contract; responsible? — purchasing/supplier management; Frequency? — in accordance with the agreed cycle, at least once a year in the case of critical suppliers, Tolerance? — No critical supplier without an ongoing cycle audit.

## 8. Typical mistakes

- The audit clause is in the contract, but is never actually exercised — a mere formality.
- All suppliers are treated equally, regardless of their actual depth of access to sensitive information.
- A certificate submitted will be accepted uncritically as a complete substitute for one's own examination.
- Existing contracts without an audit clause are never renegotiated because "new contracts are enough".

### 9. Demarcation, residual risk and references

- **Differentiation:** concerns verification of evidence in supplier management; MHC-13 regulates the technical connection of external services itself where AI agents are involved.
- **Residual risk:** even an audit carried out is a sample at a time — a deterioration occurring after the audit remains undetected until the next cycle (see MHC-10 for event-related addition).
- **TISAX-Controls®:** 6.1.1, 6.1.3 (see Part I).
- **Framework:** DORA Art. 30(3) (Sector-Specific Reference Framework Financial Sector) — Certifications submitted "support but do not replace" one's own audit; unrestricted audit/inspection right for critical functions.

## MHC-15 — AI-Resistant Identity Verification of Personnel and Access

**At a glance**

- Operational benefit: Prevents a convincing deepfake identity or a forged document from being accepted undetected in the recruitment or access process.
- Policy concerned: Personnel policy (recruitment process), access policy.
- Dependencies: no precondition; affects HR and Facility/Plant Security jointly; can be implemented independently of MHC-14.
- Expense driver: Process change in two different departments, procurement of liveness detection tools if necessary.
- Primary metric: Proportion of security-related hiring/access cases with an additional independently verified channel.
- Affected standards (without performance claim): none of the six benchmarks from phase 3 are relevant (original); TISAX® ISA 2027 2.1.1, 2.1.3, 3.1.1.

### 1. Curing Control Statement

For identity verification in the recruitment process and for physical access, verification is not based solely on a single procedure conducted by video, telephone or document presented. For roles with access to sensitive information, at least one additional, independently verified channel is provided. Awareness measures on social engineering and phishing are supplemented by AI-generated, error-free and personalized attack patterns.

### 2. Purpose and threat relevance

Convincing fake identity documents and, in combination with video interviews, deepfake identities have so far been considered specialist knowledge. AI-generated document forgery and real-time video/audio deepfakes are making deceptively real impersonation widely available in the hiring process today — a well-known current pattern is fake remote employees.

The same skill decoupling applies to physical access: fake IDs and convincing legends on the phone or during visitor registration. Classic awareness content trains the recognition of spelling mistakes and impersonal salutations — characteristics that AI-generated attacks no longer show. The goal of protection is to ensure that a convincing but fake identity is not accepted on the basis of a single channel alone.

## 3. Organizational implementation (CISO perspective)

- **Policy:** The HR policy and access policy will be supplemented by the obligation to provide a second, independently verified channel for roles with access to sensitive information.
- **Responsibilities:** Human Resources for the recruitment process, Facility/Plant Security for the access process; both areas align on a common verification principle.
- **Risk analysis:** the additional verification depth is staggered according to the access sensitivity of the respective role, not the same for all positions.
- **Processes:** defined procedure for the callback via an independently determined number or personal appearance with a liveness-verified identity document; Escalation in the event of abnormalities.
- **Training:** Awareness content for personnel and reception functions is expanded to include AI-generated deception patterns.

## 4. Technical implementation (for IT professionals)

- **Liveness detection:** where available, use of commercial anti-deepfake detection for video interviews and video identification procedures.
- **Independent call-back verification:** Contact via a self-researched number not provided by the applicant/visitor himself.
- **Document verification:** where technically possible, NFC chip reading for chip-capable ID documents instead of pure visual inspection.
- **Efficacy test:** A simulated test case (test user without the additional verification channel) goes through the process — it must technically or organizationally recognize and escalate the missing second channel, not silently let it through.

## 5. Implementation examples

**Example A — Organization with a structured remote hiring process.** A manufacturer with many remote settings supplements the video interview with a liveness check and, for safety-relevant roles, also requires a short callback via an independently researched telephone number before the final commitment. Reference checks are carried out exclusively via self-determined contact channels not specified by the applicant.

**Example B — Organization with classic on-site reception.** A physical reception location trains receptionists on AI-powered deception patterns (convincing legends, fake supplier IDs) and introduces mandatory appointment confirmation via a second, independent channel (callback instead of just email confirmation) for visits with access to vulnerable areas.

## 6. Maturity path (cumulative)

- **Initial:** additional verification channel for the most sensitive roles/accesses piloted.
- **Defined:** Process mandatory for all roles with access to sensitive information; Awareness training updated.
- **Managed:** Liveness detection or technical document verification in productive use; regular test cases for effectiveness testing.

## 7. Measurement and audit evidence

- **Metric:** Proportion of security-related hiring/access cases with a documented second verification channel (target value: 100% for roles with access to sensitive information).
- **Evidence:** updated policy texts, logs of recall verifications performed, training evidence on AI deception patterns.
- **Audit logic:** documented? — process description for both departments; responsible? — Human Resources/Plant Security; Frequency? — in any security-related case, Tolerance? — No security-relevant setting/access without a second channel without a documented exception.

## 8. Typical mistakes

- The second channel is envisaged, but it is conducted via the same contact channel indicated by the applicant/visitor himself — no real gain in independence.
- Awareness training stops at classic phishing traits without dealing with AI-generated deception.
- The additional verification applies across the board to all positions, which means that the process is perceived as a burden and is gradually circumvented.
- Liveness detection tools are procured, but their results are not included in the decision.

## 9. Demarcation, residual risk and references

- **Differentiation:** concerns personnel recruitment and physical access; the verification of supplier certificates is regulated by MHC-14.
- **Residual risk:** Liveness detection tools are no guarantee against every future deepfake generation; the organizational second-channel obligation remains the more robust compensation.
- **TISAX-Controls®:** 2.1.1, 2.1.3, 3.1.1 (see Part I).
- **Framework:** original — no benchmark from Phase 3 addresses this scenario.

# MHC-16 — Multi-Agent Integrity

**At a glance**

- Operational Benefit: Limits how far a tampered agent, faulty, or compromised agent can operate within a chain or orchestration of multiple AI agents before damage is done or spreads.
- Affected policy: AI agent usage policy (addendum for multi-agent architectures).
- Dependencies: requires MHC-13 (single-agent identity, tool access, and kill switch as a basis).
- Effort drivers: Number and complexity of agent-to-agent connections, level of orchestration, amount of shared storage/context systems.
- Primary metric: Percentage of agent-to-agent connections with authenticated, integrity-protected communication and functional circuit breaker.
- Affected standards (without performance claim): no comparative framework from phase 3 relevant (original); OWASP Top 10 for Agentic Applications (ASI06, ASI07, ASI08).

## 1. Curing Control Statement

In addition to single-agent protection (MHC-13), organizations that use multiple AI agents that communicate with each other productively (multi-agent systems, orchestrator/sub-agent architectures, agent chains) technically secure communication between agents: authenticated and integrity-protected messages between agents, a separation of shared storage and context systems along clear trust boundaries, and a circuit breaker that enables the Propagation of a faulty or manipulated agent is limited to downstream agents.

## 2. Purpose and threat relevance

The single-agent protection from MHC-13 assumes that an agent can be considered in isolation. As soon as multiple agents communicate with each other — for example, a scheduling agent that delegates subtasks to specialized execution agents, or several agents with access to a shared repository — additional, emergent risks arise: An agent can implicitly trust messages from another agent without verifying the identity of another agent or the integrity of the message (insecure inter-agent communication, including via MCP/A2A protocols). A shared memory or context store can be poisoned by a single manipulated interaction (memory/context poisoning) and then affects all subsequent consumers of that memory, possibly across sessions and agents. And an erroneous or manipulated result of an agent, if it is passed on unchecked as a trusted input to the next agent, can strengthen within the chain instead of being dampened (cascading failures) — with effects in the range of seconds, outside direct human supervision. Remarkably, even if no single agent on its own meets the well-known "lethal trifecta" pattern (simultaneous access to sensitive data, exposure to untrusted content, and ability to communicate externally), a chain of multiple agents can meet this pattern in total if each agent carries only a portion of it — a blind spot when risk is assessed only on a per-agent basis rather than system-wide. These three threats correspond

to ASI06, ASI07, and ASI08 of the OWASP Top 10 for Agentic Applications. The goal of protection is to ensure that a single compromised or faulty agent cannot affect other agents or shared storage without hindrance.

## 3. Organizational implementation (CISO perspective)

- **Policy:** The policy on the use of AI agents will be supplemented by a regulation for multi-agent architectures: documented trust boundaries between agents, mandatory release for new agent-to-agent connections.
- **Responsibilities:** In addition to the assignees appointed per single agent (MHC-13), overall responsibility is designated for each productive multi-agent architecture.
- **Risk analysis:** Trust boundary card per multi-agent system — which agent can influence which other agent or which shared storage, and with what effect in the event of an error.
- **Processes:** Release process for new agent-to-agent connections, analogous to the release mechanism from MHC-13 for tools; documented response to triggered circuit breakers.

## 4. Technical implementation (for IT professionals)

- **Authenticated inter-agent communication:** each agent cryptographically verifies the identity of a sending agent (connecting to the agent identity from MHC-13) before a received message is treated as trusted; Protection against spoofing and agent-in-the-middle.
- **Isolation of memory and context systems:** shared memories (memory, vector/context stores) are partitioned along trust boundaries; Entries are given an origin label (which agent, which source) so that a consuming agent can assess the trustworthiness of an entry.
- **Circuit breaker (blast radius limitation):** automated interruption of an agent chain or orchestration in the event of unusual patterns (repeated contradictory results, error accumulation, unexpected action sequences) before the effect continues to downstream agents.
- **Validation of intermediate results:** Results that are passed from one agent to another go through a sanity/schema check instead of being accepted as unchecked as trusted input.
- **Efficacy test:** a deliberately manipulated message or a poisoned memory entry is imported at one point in the chain; the spread to downstream agents must be technically prevented or the circuit breaker must be triggered, and the process must be logged in full.

## 5. Implementation examples

**Example A — Orchestrated development pipeline.** An organization with its own software development uses a planning agent that delegates subtasks to several specialized execution agents (code creation, testing, documentation). If an execution agent repeatedly delivers

contradictory or conspicuous results, a circuit breaker automatically interrupts the affected subchain instead of passing on the result unchecked; a shared context store is partitioned in such a way that a manipulated entry by an execution agent does not directly affect the planning agent's decisions.

**Example B — Customer service with an internal inquiry agent.** A customer-side support agent can consult a second, internally connected agent with access to order and customer data for queries. The customer-side agent does not meet a complete lethal trifecta pattern on its own (it does not have direct access to internal systems); only the chain of both agents does it. Therefore, every request to the internal agent is authenticated and the content it contains, originally from the customer, is treated as untrusted input, not as an internal instruction that has already been checked.

## 6. Maturity path (cumulative)

- **Initial:** Inventory of agent-to-agent connections available, communication not yet technically secured.
- **Defined:** authenticated, integrity-protected inter-agent communication active; Storage/context systems separated along trust boundaries.
- **Managed:** Circuit-breaker productive with defined threshold values; regular test against context poisoning and cascade effects; Trust Limits map kept up to date.

## 7. Measurement and audit evidence

- **Metric:** Percentage of agent-to-agent connections with authenticated, integrity-protected communication (target: 100%); Proportion of productive multi-agent systems with documented trust boundary map and functional circuit breaker (target value: 100%).
- **Proof:** Trust Boundary Map, Authentication Configuration Proof, Context Poisoning/Cascade Test Test Log, Circuit Breaker Trigger History.
- **Audit logic:** documented? — trust boundary map per multi-agent system; responsible? — named overall responsibility per system, in addition to the single-agent managers from MHC-13; Frequency? — test at least once a year in view of the rapid development in this field, Tolerance? — No productive multi-agent system without authenticated inter-agent communication and a functional circuit breaker.

## 8. Typical mistakes

- Agents implicitly trust messages from other agents because they come "from their own system" — no distinction between internally verified and externally influenced origin.
- Shared storage or context store without origin labeling — not possible to reconstruct which agent or external source caused a particular entry.
- Not a circuit breaker — a single faulty or manipulated agent continues to supply an entire chain or orchestration with erroneous results unchecked.

- Risk assessment is only carried out per individual agent, not for the chain as a whole — a chain that satisfies a lethal trifecta pattern remains undetected because no individual agent is conspicuous in it.

## 9. Demarcation, residual risk and references

- **Demarcation:** requires an architecture with several agents communicating with each other; the protection of a single agent (identity, tool access, kill switch, basic hardening against prompt injection) is regulated by MHC-13, on which this entry is based.
- **Residual risk:** emergent misbehavior in complex multi-agent systems is not fully predictable; a circuit breaker limits the impact, but does not prevent every wrong decision in individual cases.
- **TISAX-Controls®:** none. This MHC is completely original — no TISAX® requirement finding from Part I is based on it (see Part I, section "Original Supplements beyond TISAX®").
- **Framework:** original — no comparison framework from phase 3 relevant; OWASP Top 10 for Agentic Applications (ASI06 Memory & Context Poisoning, ASI07 Insecure Inter-Agent Communication, ASI08 Cascading Failures) as a threat taxonomy, not as a compliance framework; Priority P2, forward-looking — relevant as soon as several agents communicate productively with each other (see prioritization chapter).

## Supplementary individual measures

The seven individual findings from Part I without a recurring pattern (spread over six controls) do not receive a complete MHC treatment — a six-frame origin test would not be meaningful for individual cases. They are incorporated as independent, compactly formulated measures:

| Control | Findings (short version) | Recommended Supplement |
|---|---|---|
| 1.2.2 | Role separation can be circumvented by AI double role | Enforce critical role separation in addition to system technology (separate system rights, four-eyes clearance in the system itself), not only organizationally |
| 1.4.1 (2 requirements: must + should) | Risk assessment methodology assumes a narrowly defined group of offenders — both at the level of the assessment itself and the underlying assessment criteria | Supplement the methodology with a reassessment obligation in the event of a significantly expanded group of offenders (AI accessibility of a capability), not only in the case of new individual vulnerabilities — applies equally to assessment results and criteria |
| 1.6.3 (high) | AI-orchestrated scenarios not provided for in crisis planning | Supplement the crisis scenario catalog with at least one AI-orchestrated, multi-pronged escalation scenario (see also MHC-05) |
| 3.1.4 | PIN/password as the sole fallback option without encryption obligation | Encryption of mobile devices/data carriers as a minimum requirement, regardless of the level of protection required |
| 5.1.2 (must) | Non-specific measures for remote access without a minimum standard | Encryption as a binding minimum standard already at the must-level, not only from high |
| 5.3.1 | Re-identification of pseudonymized data through AI correlation | Complement pseudonymization methodology with re-identification risk assessment taking into account AI-supported correlation methods |

# Glossary

- **A2A (Agent-to-Agent): Protocol** class for direct communication between AI agents, including Google's A2A protocol.
- **AITM (Adversary-in-the-Middle):** Real-time phishing relay that redirects a login through a fake page and grabs the valid authentication factor directly from the victim.
- **Allow-Listing:** Whitelist of allowed targets (tools, services, packages); anything not listed is locked by default.
- **Confused Deputy:** Attack pattern in which a component with higher privilege is tricked into abusing that privilege on behalf of a less privileged or malicious actor.
- **CVSS (Common Vulnerability Scoring System):** Standardized rating scale (0–10) for the criticality of a vulnerability.
- **EDR / XDR (Endpoint/Extended Detection and Response):** Behavior-based detection and response tools at endpoint or multiple levels at the same time.
- **FIDO2 / WebAuthn:** Open standard for phishing-resistant login with cryptographic binding to the address of the service.
- **Immutable / WORM:** Storage in which data cannot be changed or deleted for a specified period of time.
- **KEV (Known Exploited Vulnerabilities):** List of vulnerabilities maintained by an authority (e.g. CISA) that is demonstrably already actively exploited.
- **Kill switch:** technical possibility to deactivate a single agent or integration in a targeted and rapid manner without affecting other systems.
- **Liveness-Detection:** Technical procedure for checking whether a person is present in a video/image stream in real and live, not deepfake-generated.
- **MCP (Model Context Protocol):** Open standard through which AI agents access external tools, data, and services.
- **MTTC (Mean Time to Containment):** Average time to contain a safety-related event.
- **NHI (Non-Human Identity):** technical identity of a non-human actor (service account, workload, AI agent) with its own manageable credentials.
- **Policy-as-Code:** Security specifications that are formulated as executable, versioned rules (instead of text) and enforced automatically.
- **Prompt injection:** Attack technique in which covert instructions are injected into an AI system via seemingly harmless content (e-mail, document, tool output).
- **Rug pull (component context):** Subsequent, unnoticed change of an already released component (e.g. MCP server, extension) during an update, whereby behavior or scope of authorization changes without going through the original check.
- **SAST / DAST / SCA:** Static Code Analysis, Dynamic Application Testing and Dependency Analysis (Software Composition Analysis) — three complementary automated test classes in the development pipeline.
- **SOAR (Security Orchestration, Automation and Response): Platform** that automatically executes defined response sequences (playbooks).
- **UEBA (User and Entity Behavior Analytics):** Behavior-based recognition based on a learned normal line for users and systems.
- **Playbook:** Predefined response flow for a specific incident type.

# Mapping: MHC ↔ MRIS-TISAX® 2027, Part I

| MHC | Title | Part I: Controls | Part I: Clusters (Phase 2) | Key figure |
|---|---|---|---|---|
| MHC-03 | Phishing-resistant multi-factor authentication | 4.1.1, 4.1.2, 4.1.3, 5.1.2, 5.2.7, 6.1.3 | Cluster 3 | Share of phishing-resistant accesses |
| MHC-05 | Behavior-based detection and kill chain correlation | 5.2.3, 1.6.1, 1.6.3, 5.2.4, 5.2.7 | Cluster 5 + 6 | EDR/XDR coverage; Proportion of correlated log sources |
| MHC-08 | Immutable backups and recovery validation | 5.2.8 | Cluster 2 (Backup Part) | Recovery Test Success Rate |
| MHC-09 | AI-powered security testing in the pipeline | 1.3.4, 5.2.5, 5.2.6 | Cluster 2 (Patch Part) | Deadline Compliance with Critical Vulnerabilities |
| MHC-10 | Continuous Control Monitoring and Policy-as-Code | 1.1.1, 1.2.1, 1.3.1, 1.3.3, 1.3.4, 1.5.1, 1.6.3, 4.1.3, 4.2.1, 5.2.9, 6.1.2, 6.1.3 | Cluster 1 | Share of testing processes with trigger mechanism |
| MHC-11 | SOAR-based Tier 1 automation and parallel response playbooks | 1.6.2 | Cluster 8 | Mean Time to Containment |
| MHC-13 | AI Agent Governance and Harness Security | 1.3.3, 1.3.4 | Cluster 7 | Percentage of technically enforced allow lists |
| MHC-14 | Independent Supplier Verification | 6.1.1, 6.1.3 | Cluster 4 (Supplier Part) | Percentage of audited critical suppliers |
| MHC-15 | AI-Resistant Identity Verification Staff/Access | 2.1.1, 2.1.3, 3.1.1 | Cluster 4 (Staff/Access Part) | Proportion of cases with a second channel |
| MHC-16 | Multi-Agent Integrity | none (original) | none (original, outside TISAX® diagnosis) | Percentage of authenticated agent-to-agent connections |