TID-CMM

Threat-Informed Detection Capability Maturity Model

A framework for measuring whether your detection capability is driven by adversary behaviour — and whether it has been proven to work.

Version 1.5.0 · Released 2026-08-17

Aligned to MITRE ATT&CK Enterprise v19.2 (697 techniques, 109 data components)

Crosswalked to NIST CSF 2.0, SOC-CMM, ISO/IEC 27001:2022 and MITRE D3FEND

8 domains · 58 sub-capabilities · 348 level descriptors

https://tid-cmm.com · hello@tid-cmm.com

Licence: CC-BY-4.0

Contents

Executive summary

1. Why detection programmes plateau

1.1 The measurement problem
1.2 The three failure modes
1.3 What the plateau looks like

2. What already exists, and what is missing

2.1 The gap, stated precisely

3. Design principles

3.1 Evidence over assertion
3.2 Behaviour over inventory
3.3 Whole loop, not one function
3.4 Constraints over exhortation
3.5 Scope honestly, then measure
3.6 Two audiences, one assessment

4. The model

4.1 Structure
4.2 The maturity scale
4.3 The three integrity constraints
4.4 The Validated Coverage Score

5. The domains in detail

5.1 TI — Threat Intelligence & Adversary Prioritisation
5.2 TM — Threat Modeling & Attack Path Analysis
5.3 DC — Telemetry & Detection Coverage
5.4 DE — Detection Engineering
5.5 AV — Adversarial Validation & Emulation
5.6 AA — Analytics, Automation & Hunting
5.7 IR — Incident Response & Recovery
5.8 GV — Governance, Metrics & Continuous Improvement

6. Running an assessment

6.1 Choose the assessment type
6.2 Who takes part
6.3 Sequence
6.4 Common failure modes in the assessment itself

7. Scoring mechanics

7.1 Rollup
7.2 Order of operations
7.3 Bands
7.4 Prioritisation arithmetic

8. The toolkit

8.1 Privacy
8.2 Consistency between implementations

9. Worked example

9.1 Scope

9.2 Results
9.3 Coverage
9.4 The roadmap
9.5 What the CISO is told

10. Adoption

10.1 A first ninety days
10.2 Setting a target
10.3 Re-assessment
10.4 Contributing
10.5 Licence

Appendix A — Full sub-capability register

Appendix B — Framework crosswalk

Appendix C — ATT&CK alignment and scoping

Appendix D — Glossary

Executive summary

Most organisations cannot answer a simple question about their own security operations: against the adversaries most likely to attack us, what proportion of their behaviour would we actually see — and how do we know?

The question is harder than it sounds, and the reason is structural. Detection capability is usually described by the things that are easy to count: the number of log sources onboarded, the number of rules enabled, the number of alerts triaged, the number of techniques shaded green on an ATT&CK Navigator layer. None of these measures capability. A rule that has never fired on a true positive, running on a log source that stopped reporting six weeks ago, mapped to a technique the organisation's adversaries do not use, counts identically to a rule that has been proven by emulation to catch a real intrusion within minutes.

TID-CMM exists to close that gap. It is a maturity model in the conventional sense — eight domains, fifty-three sub-capabilities, each scored from 0 to 5 with explicit descriptors — but it differs from its peers in three deliberate ways.

- **It treats adversarial validation as a first-class domain, not an afterthought.** Atomic testing, breach and attack simulation, threat-actor emulation, purple teaming, penetration testing and red teaming are scored together as the evidence engine of the whole model.
- **It treats threat modeling and attack path analysis as a first-class domain.** Threat intelligence tells you what an adversary does in general; attack trees and computed attack paths tell you what that behaviour looks like against your architecture, your identities and your crown jewels. Without that step, ATT&CK coverage is a generic checklist.
- **It refuses to let the assessment flatter itself.** Three integrity constraints are applied mechanically at scoring time: no domain may exceed the validation domain by more than one level; detection engineering may not exceed the telemetry it runs on by more than one level; and a score of 4 or 5 without a named artefact is counted as a 3.

Those constraints are the model's distinguishing feature, and they are not decorative. In the worked example in Chapter 9, an organisation with genuinely strong security operations self-assesses at 2.44. After the constraints are applied it scores 2.34, and the reason is visible in one line of output: four separate domains were capped because the organisation had never emulated an adversary against the capability it was claiming. The model does not merely report a lower number; it identifies the single investment that would raise every other domain.

> **The central claim.** An untested detection capability is an assumed detection capability. A maturity model that scores assumptions the same way it scores evidence will systematically overstate readiness, and will do so most severely in the organisations that have spent the most money.

TID-CMM is open. The model is published as machine-readable YAML with a JSON schema, a Python scoring engine, an Excel self-assessment workbook and a single-file browser tool that requires no installation and transmits nothing. Model content is licensed CC-BY-4.0 and the tooling Apache-2.0, so organisations can embed it in internal assurance processes and vendors can implement it without negotiation.

The intended outcome of an assessment is not a number. It is a ranked list of the things that would most improve your ability to see an adversary, expressed in terms your detection engineers can act on and your board can fund.

1. Why detection programmes plateau

1.1 The measurement problem

A security operations function that has been running for five years will typically have accumulated a large detection estate: several thousand rules across a SIEM, an EDR platform, an identity provider, a cloud-native detection service and whatever the email gateway contributes. It will have onboarded most of the log sources anyone thought to ask for. It will have an incident response plan, a case management platform, a threat intelligence subscription and a set of dashboards.

It will also, usually, be unable to say which of those rules work.

This is not incompetence. It is the predictable result of measuring the wrong things for a long time. Consider what the standard metrics actually tell you:

Common metric	What it appears to measure	What it actually measures
Number of detection rules	Detection breadth	Historical accumulation, including duplicates, vendor defaults and content nobody dares disable
ATT&CK techniques ‘covered’	Adversary coverage	How many rules have had a technique ID typed into a metadata field
Alerts triaged per analyst	Operational throughput	Noise volume, which usually correlates negatively with precision
Mean time to detect	Speed of detection	Speed of detecting the things you already detect. It is silent on everything you miss
Log sources onboarded	Visibility	Ingestion, not coverage. It says nothing about parsing, completeness, timeliness or whether anything queries the data
Threats blocked	Protection	Perimeter noise. It is dominated by commodity scanning and tells you nothing about targeted intrusion

Table 1.1 — Standard security operations metrics and what they are actually measuring.

Each metric has a common property: it can be improved without improving the organisation's ability to detect an adversary, and in several cases it improves fastest when capability degrades. Alert volume rises when tuning is neglected. Rule count rises when nothing is retired. Mean time to detect improves when the noisy, easy, low-value detections dominate the sample.

1.2 The three failure modes

Across detection programmes, the same three failures recur, and they are the failures the model is built to expose.

Failure one: coverage claimed without visibility

A detection rule is written against a data source that is only deployed on 60% of the estate, or that stopped parsing correctly after a vendor agent upgrade, or whose ingestion pipeline silently drops events above a volume threshold. The rule exists. It is enabled. It is mapped to a technique. It cannot fire. Nobody knows, because nothing tests it and nothing monitors the dependency.

This failure is invisible to every metric in Table 1.1. It is why TID-CMM makes telemetry quality a scored sub-capability with its own ceiling rule, and why detection engineering cannot score more than one level above the telemetry domain.

Failure two: detection built without a threat model

An organisation buys a detection content subscription and enables everything. Coverage of the ATT&CK matrix rises impressively. But the content was written for a generic enterprise, and the organisation's actual exposure — a bespoke payment application, a specific cloud entitlement path to the customer database, a third-party integration with standing privileged access — is not in the matrix in any actionable form. The adversary who matters does not need to use a technique the content covers. They need to use the path nobody modelled.

Generic ATT&CK coverage is a necessary foundation and an insufficient one. TID-CMM addresses this by scoring attack tree construction and attack path analysis as distinct capabilities, and by requiring traceability from a modelled node to a deployed detection to a validation result.

Failure three: capability asserted without evidence

The most consequential failure. Someone is asked whether the organisation would detect credential dumping. They think about it, recall that there is a rule for it, and say yes. That answer then propagates into a risk register, a board report, a regulatory submission and a budget decision. It is never tested. When it is tested — usually by an adversary — the answer turns out to be no, for a reason that would have taken twenty minutes of emulation to discover.

> **Why this failure dominates.** Assertion is free and testing is not. In any assessment process where an unevidenced claim scores the same as an evidenced one, assertion will drive out evidence, because the incentives are unambiguous. The only durable fix is structural: make the unevidenced claim score lower.

1.3 What the plateau looks like

The observable symptom is a programme that spends steadily and improves little. Tooling is refreshed, headcount grows modestly, dashboards proliferate, and yet the same categories of intrusion succeed in the same way. Post-incident reviews produce lessons that are identified but not engineered. Penetration test reports are read by the risk function and never reach the detection engineers. Purple team exercises happen once, produce a long finding list, and the findings age quietly in a tracker.

The plateau is not a resourcing problem. It is a feedback problem. The loop that should connect intelligence to modelling to telemetry to detection to validation to improvement is broken in several places at once, and no single team can see the whole break. A maturity model is useful here precisely because it forces the whole loop into one picture.

2. What already exists, and what is missing

TID-CMM is not built in a vacuum, and it deliberately does not duplicate work that is already good. This chapter states what each adjacent framework does well, and the specific gap that justifies another model.

Framework	What it does well	Where it leaves a gap for TID-CMM
MITRE ATT&CK	A shared, evidenced vocabulary for adversary behaviour, maintained and versioned. The single most valuable contribution to defensive practice in a decade.	It is a knowledge base, not a maturity model. It describes behaviour; it does not tell you whether your organisation's process for turning that behaviour into detection is any good, nor whether your claimed coverage is real.
SOC-CMM	Rigorous, well-established assessment of the SOC as an organisational function across business, people, process, technology and services. Excellent for structuring a security operations capability.	It assesses the SOC as an operating unit. It is comparatively light on technique-level detection coverage, on adversarial validation as a discipline, and on attack path analysis. It answers ‘is this a well-run SOC?’ more than ‘would this SOC see the adversary?’
NIST CSF 2.0	The common language for expressing cybersecurity posture to executives, regulators and insurers. Broad, governance-aware, and now with a Govern function that materially improves it.	Deliberately outcome-level and technology-neutral. ‘DE.CM-09: computing hardware and software are monitored to find potentially adverse events’ is correct and unarguable, and provides no way to distinguish a programme that does this well from one that does it nominally.
Elastic DEBMM	A focused, practical model for detection engineering behaviour specifically, and a genuine advance in treating detection as an engineering discipline.	Scoped, by design, to detection engineering. It does not extend to threat modeling, attack path analysis, adversarial validation, response or governance, and so cannot produce a whole-loop picture.
Hunting Maturity Model	A clear, widely understood scale for hunting capability that has usefully shaped practice.	Single-capability scope. Hunting is one sub-capability within one domain of a detection programme.
Gartner CTEM	The right cycle for exposure management — scoping, discovery, prioritisation, validation, mobilisation — and it correctly elevates validation.	Exposure-centric rather than detection-centric. It asks whether an exposure can be exploited; TID-CMM asks whether the exploitation would be seen. The two are complementary and TID-CMM's attack path sub-capability is explicitly crosswalked to CTEM.
ISO/IEC 27001:2022	Auditable management-system discipline and international recognition.	Control-existence oriented. An organisation can be certified while detecting almost nothing, because the standard asks whether a control is implemented and reviewed, not whether it works against an adversary.
C2M2, CMMC and sector models	Strong domain structure and, in CMMC's case, assessed rather than self-declared.	Compliance-anchored and slow-moving relative to adversary tradecraft. Neither is designed to be re-run quarterly against a shifting threat profile.

Table 2.1 — Adjacent frameworks and the gap TID-CMM addresses.

2.1 The gap, stated precisely

There is no widely adopted, open model that does all four of the following at once:

1. Scores the complete loop from intelligence, through threat modeling and attack path analysis, through telemetry and detection engineering, through adversarial validation, to response and governance.
2. Anchors coverage claims to ATT&CK at technique and sub-technique level, while explicitly rejecting whole-matrix coverage as a vanity metric.
3. Requires adversarial validation as the evidence for capability claims, and structurally prevents a high score without it.

4. Is machine-readable, freely licensed, and shipped with working tooling rather than a PDF and an invitation to build your own spreadsheet.

TID-CMM is designed to occupy that space, and to sit alongside rather than replace the frameworks above. An organisation running SOC-CMM should keep running it; the crosswalk in Appendix B lets a TID-CMM assessment feed the same reporting. An organisation reporting in NIST CSF 2.0 terms can map every sub-capability to a CSF outcome and use TID-CMM as the evidence layer underneath.

3. Design principles

Six principles governed the design, and each one has a visible consequence in the model.

3.1 Evidence over assertion

Every sub-capability defines what evidence would justify each level, and the scoring engine enforces it at the top of the scale. In strict mode, a 4 or a 5 without a named artefact is recorded as a 3, with the reason logged. This is not an accusation of dishonesty; it is a recognition that self-assessment without an evidence rule reliably drifts upward, and that the drift is largest in exactly the areas where nobody has looked.

3.2 Behaviour over inventory

The model scores what the organisation does, not what it owns. No sub-capability asks whether a particular class of product has been purchased. Several ask whether the outcome that product is meant to deliver has been achieved and demonstrated. An organisation can reach Level 4 in analytics and automation with modest tooling and excellent process; it cannot reach it with excellent tooling and no process.

3.3 Whole loop, not one function

Detection capability is a loop, and loops fail at their weakest joint. Scoring detection engineering in isolation produces a locally optimised function attached to broken inputs and outputs. The eight domains cover the full circuit, and the weighting reflects that detection engineering is the largest single contributor but nowhere near a majority.

3.4 Constraints over exhortation

Most maturity models handle the risk of over-scoring by telling assessors to be honest. TID-CMM handles it arithmetically. If the validation domain scores 1.6, every other domain is capped at 2.6 regardless of what the assessor believes, and the cap is reported explicitly rather than hidden in the total. Exhortation does not survive a budget cycle; arithmetic does.

3.5 Scope honestly, then measure

The model rejects coverage measured against the full ATT&CK matrix. An organisation with no macOS estate and no container platform gains nothing from reporting that it does not detect techniques it cannot experience. It must instead define an in-scope technique set from its platforms and its prioritised threat profile, record the rationale, and measure against that. This makes the number smaller, more difficult to game and considerably more useful.

> **A warning that is built into the tooling.** A high coverage score over a small, conveniently chosen in-scope set is worse than a low score over an honest one, because it is persuasive. Both the workbook and the browser tool display the in-scope count beside the score for exactly this reason.

3.6 Two audiences, one assessment

A detection engineer needs technique-level gaps, data component dependencies and the specific descriptor of the next level up. A CISO needs a defensible statement of what the organisation can and cannot detect, what that costs, and what changes if it is funded. These are different reports from the same data, and the tooling produces both without a second assessment.

4. The model

4.1 Structure

TID-CMM v1.5.0 comprises 8 domains and 58 sub-capabilities. Each sub-capability is scored 0 to 5 against explicit descriptors, or marked not applicable with a recorded rationale. Sub-capability scores roll up to a weighted domain score; domain scores roll up to a weighted overall score.

Domain	Name	Weight	Sub-capabilities	Question it answers
TI	Threat Intelligence & Adversary Prioritisation	12%	6	Who are we defending against, and how do we know?
TM	Threat Modeling & Attack Path Analysis	12%	7	What do their behaviours look like against our architecture?
DC	Telemetry & Detection Coverage	14%	6	Can we see the activity at all?
DE	Detection Engineering	16%	10	Do we build, test and maintain detection like engineers?
AV	Adversarial Validation & Emulation	14%	8	Have we proven any of it works?
AA	Analytics, Automation & Hunting	12%	8	Does detection output become a decision at operational tempo?
IR	Incident Response & Recovery	10%	6	Can we act on what we detect?
GV	Governance, Metrics & Continuous Improvement	10%	7	Is this directed, measured and sustainable?

Table 4.1 — The eight domains, their weights and the question each answers.

The weighting is a judgement, and it is exposed as an editable parameter in every tool so an organisation can defend a different one. Detection engineering carries the largest weight (16%) because it is where capability is manufactured. Telemetry and adversarial validation follow at 14% each because they are, respectively, the precondition and the proof. Governance and incident response carry 10% each — not because they matter less, but because both are extensively covered by other frameworks the organisation is likely already running, and TID-CMM's marginal contribution there is smaller.

4.2 The maturity scale

The same six-point scale applies to every sub-capability. The scale is deliberately not the classic CMMI wording; the third level is named Threat-Informed rather than Defined, because in this model the step from 2 to 3 is precisely the step from doing something consistently to doing it because a specific adversary made it necessary.

Level	Name	What it means	Evidence bar
0	Absent	The capability does not exist in any recognisable form.	Nothing to show.
1	Ad hoc	Happens occasionally, driven by individual initiative or an incident. Undocumented, unrepeatable, lost when the individual leaves.	Anecdote, a person who "knows how".
2	Repeatable	Documented and consistently performed, but driven by compliance, tooling defaults or vendor content rather than by adversary behaviour.	A written procedure and proof it was followed more than once.

Level	Name	What it means	Evidence bar
3	Threat-Informed	Driven by a prioritised adversary profile. Work is explicitly mapped to ATT&CK techniques and traceable back to an intelligence requirement or a threat model.	ATT&CK-mapped artefacts with traceability to a threat driver.
4	Measured & Validated	Quantitatively managed and independently proven. Claims are tested by emulation, results are measured over time, and gaps enter a managed backlog.	Test results, trended metrics, closed-loop backlog records.
5	Adaptive	A self-correcting closed loop. Change in the threat landscape automatically produces changes in telemetry, detection and validation, with measured cycle time. The organisation contributes findings back to the community.	Automated pipeline metrics, cycle-time trends, external contributions.

Table 4.2 — The TID-CMM maturity scale.

Two properties of the scale deserve comment. First, the step from 3 to 4 is the largest in the model, because it is the step from doing threat-informed work to proving it. Most organisations that consider themselves mature sit between 2.5 and 3.5; the population above 4 is small and, in the authors' experience, always contains a substantial validation programme. Second, Level 5 requires contribution back to the community. This is a deliberate statement about what an optimising security function looks like: it is one whose learning does not stop at its own perimeter.

4.3 The three integrity constraints

These are the mechanism that distinguishes TID-CMM from a self-assessment questionnaire. They are applied by the scoring engine after the raw scores are collected, and every adjustment is reported.

C1 — Validation ceiling

No domain may be scored above the AV domain score + 1. You cannot claim measured, validated detection engineering if you have never emulated an adversary against it.

C2 — Visibility ceiling

DE (Detection Engineering) may not exceed DC (Telemetry & Detection Coverage) + 1. Detection logic cannot be more mature than the telemetry it runs on.

C3 — Evidence rule

Any score of 4 or 5 requires a named artefact recorded in the evidence field. Unevidenced 4s and 5s are downgraded to 3 by the scoring engine in strict mode.

C4 — Intent ceiling

DC and DE may not exceed max(TI, TM) + 1. Telemetry and detection content cannot be more mature than the strategy directing them. Sensors without architectural intent produce noise, not defence.

C5 — Inherited intent ceiling

TI.2 may not exceed level 2 where the adversary set was accepted from the tool's suggested threat profile without modification. Accepting it unchanged means the question TI.2 asks — who are you defending against, and how do you know? — was answered by the tool, not by you. Reviewing the set lifts the ceiling: add an actor, remove one, or record why one was excluded.

Constraint C1 encodes the model's central claim: an untested capability is an assumed capability. It permits a one-level margin, which acknowledges that a capability can be genuinely well-built before it has been validated — but only just. Constraint C2 encodes the physical reality that detection logic cannot outperform its inputs. Constraint C3 makes evidence the price of a high score.

These constraints are visible, not hidden. Every tool reports the unadjusted self-assessed score beside the adjusted one, together with a line stating which constraint was applied and why. The gap between the two numbers is itself a finding, and often the most useful one in the assessment.

4.4 The Validated Coverage Score

Alongside the maturity score, TID-CMM defines a coverage metric that is deliberately harder to game than a Navigator layer. Each in-scope ATT&CK technique is scored on a four-point scale:

Status	Meaning	What it requires
0	No telemetry	The activity would generate no record you collect. You are blind.
1	Telemetry only	The data exists and is queryable. Nothing alerts. Useful for hunting and investigation, not for detection.
2	Detection logic exists	A rule or analytic is deployed and healthy. It has not been proven to fire on the real behaviour.
3	Validated by emulation	The behaviour has been executed and the detection observed to fire, within the defined recency window.

Table 4.3 — The coverage status scale underlying the Validated Coverage Score.

The Validated Coverage Score is the achieved points divided by the maximum available across the in-scope set, expressed as a percentage. It is reported alongside the domain scores, never instead of them, and always beside the in-scope technique count.

> **Recency matters.** A status of 3 expires. A validation result older than the organisation's defined review window is downgraded to 2, because a detection proven to work eighteen months ago, across two platform migrations and a data model change, is not proven to work now.

5. The domains in detail

5.1 TI — Threat Intelligence & Adversary Prioritisation

Weight 12% · 6 sub-capabilities

Establishes *who* you are defending against and *why*. Without this domain the rest of the model has no input signal — detection becomes vendor-content-driven rather than threat-informed. This domain is scored on the quality of the prioritisation decision, not on the number of feeds consumed.

> **Anti-pattern.** A large IOC feed volume with no documented threat profile. Feed count is not intelligence; it is procurement.

ID	Sub-capability	Weight	The question
TI.1	Intelligence requirements and PIRs	18%	Are there documented, prioritised intelligence requirements that state what the organisation needs to know, tied to business risk and to named decisions?
TI.2	Threat profile and adversary prioritisation	20%	Is there a maintained, evidenced threat profile that names the actors, campaigns and behaviours most relevant to this organisation, and ranks them?
TI.3	Technical CTI ingestion and indicator lifecycle	14%	Are technical indicators ingested, scored, deployed, aged and retired under a defined lifecycle, with measured operational value?
TI.4	TTP extraction and ATT&CK mapping discipline	18%	Is finished reporting systematically decomposed into ATT&CK-mapped adversary behaviours with enough procedural detail to build a detection from?
TI.5	Intelligence-to-detection tasking	18%	Does intelligence reliably and measurably produce detection, hunting and emulation work — and can you prove the linkage?
TI.6	Dissemination, sharing and community contribution	12%	Is intelligence delivered in the form each consumer can act on, and does the organisation contribute back to sector and community sharing?

Table 5.1 — TI sub-capabilities. Full 0–5 descriptors are in Appendix A.

Where this domain turns

The pivotal step in this domain is TI.4, extraction to procedure level. An organisation that tags reports with parent technique IDs has a searchable archive; an organisation that extracts the specific command line, API call or protocol behaviour has something a detection engineer can build from this afternoon. The difference between those two states is usually the difference between a threat intelligence function that is respected by engineering and one that is politely ignored.

5.2 TM — Threat Modeling & Attack Path Analysis

Weight 12% · 7 sub-capabilities

Converts "which adversary" into "which path through *our* estate". Threat intelligence tells you what an adversary does in general; threat modeling and attack path analysis tell you what that behaviour looks like against your specific architecture, identities and crown jewels. This is the domain that stops ATT&CK coverage from being a generic checklist.

> **Anti-pattern.** A threat model produced once for an audit, in a diagram tool, never revisited, and never referenced by a single detection rule.

ID	Sub-capability	Weight	The question
TM.1	Asset, identity and crown-jewel identification	13%	Do you know what you are actually protecting — the systems, data, identities and business processes whose compromise would matter most?
TM.2	System and data-flow threat modeling	16%	Are systems threat-modelled using a recognised structured method, and does that modelling happen at the right point in the delivery lifecycle?
TM.7	Attack surface enumeration	14%	Do you continuously know every way in — external exposure, APIs, SaaS tenants, cloud services, identity federation, shadow IT and third-party ingress — so attack trees are rooted in reality rather than in the architecture diagram?
TM.3	Attack tree construction	16%	Are attack trees built for the objectives that matter — decomposing an adversary goal into the alternative branches by which it can be achieved — and are all branches carried through to a detection decision?
TM.4	Attack path and exposure analysis	15%	Do you analyse real, computed attack paths through identity, network and cloud relationships in the live estate — not only hypothetical ones?
TM.5	Abuse cases to detection requirements traceability	14%	Can you trace a specific detection rule back to the threat model or attack tree node that justified it — and identify model nodes with no detection?
TM.6	Model maintenance and change triggers	12%	Are threat models, attack trees and path analyses kept alive by defined triggers, or do they decay silently after first publication?

Table 5.2 — TM sub-capabilities. Full 0–5 descriptors are in Appendix A.

Where this domain turns

The pivotal step is TM.3 and TM.4 taken together: moving from threat models as documents to attack trees and computed attack paths as structured, queryable data. Once trees are data, choke-point analysis becomes possible — identifying the nodes that appear across many trees and are therefore worth disproportionate detection investment. This is where a detection strategy stops being a list and starts being an argument.

5.3 DC — Telemetry & Detection Coverage

Weight 14% · 6 sub-capabilities

Detection is bounded by visibility. This domain measures whether the right telemetry exists, whether it is trustworthy, and whether ATT&CK coverage claims are grounded in data rather than in a spreadsheet of intentions.

Anti-pattern. A green ATT&CK Navigator layer produced by counting rule names, with no check that the underlying data components are actually collected and healthy.

ID	Sub-capability	Weight	The question
DC.1	Log source inventory and ownership	14%	Is there a complete, owned inventory of telemetry sources with their scope, coverage percentage and criticality?
DC.2	Telemetry quality, completeness and timeliness	18%	Do you measure whether the data arriving is complete, correctly parsed, timely and unaltered — and do you alert when it is not?
DC.3	Normalisation and data model discipline	14%	Is telemetry normalised to a documented, versioned data model so detections are portable and analysts are not re-learning field names per source?
DC.4	ATT&CK technique coverage measurement	20%	Is coverage measured honestly at technique and sub-technique level, grounded in data-component availability and detection validity — not in rule counts?
DC.5	Coverage breadth across attack surfaces	20%	Does visibility extend across every surface the prioritised adversaries use — endpoint, identity, cloud control plane, SaaS, network, email, application, container, OT/IoT and third-party — rather than concentrating on endpoint?
DC.6	Visibility gap management	14%	Are known blind spots recorded, prioritised, costed and driven to closure — or quietly tolerated?

Table 5.3 — DC sub-capabilities. Full 0–5 descriptors are in Appendix A.

Where this domain turns

The pivotal step is DC.2, telemetry quality. Almost every organisation believes its telemetry is fine, and almost none measures completeness, parsing success, ingestion latency and field-level fill rate per source. Silent telemetry failure is the most common cause of a detection that exists and cannot fire, and it is invisible until either someone measures it or an adversary benefits from it.

5.4 DE — Detection Engineering

Weight 16% · 10 sub-capabilities

The production engine of the model. Measures whether detection content is built, tested, released, monitored and retired with the discipline of software engineering — and whether the portfolio is deliberately composed rather than accumulated.

Anti-pattern. Thousands of enabled vendor rules, no version control, no test, no owner, and a "we can't turn it off in case we need it" culture.

Ceiling rule. DE may not score more than one level above DC. Detection logic cannot be more mature than the telemetry it runs on.

ID	Sub-capability	Weight	The question
DE.1	Detection lifecycle and intake	10%	Is there a defined lifecycle from requirement through design, build, test, release, monitor and retire — with a controlled intake?
DE.2	Detection-as-code	12%	Is detection content managed as code — versioned, peer-reviewed, and deployed through an automated pipeline?
DE.3	Detection standards, metadata and documentation	10%	Does every detection carry the metadata needed to operate, audit and improve it — and is a shared standard enforced?
DE.4	Testing and pre-deployment validation	13%	Is every detection proven to fire on true positive input and stay quiet on benign input, before it reaches production?
DE.5	Tuning, precision and false-positive management	10%	Is alert precision measured per detection and improved deliberately, rather than by disabling noisy rules?
DE.6	Detection health and silent-failure monitoring	10%	Would you know if a detection stopped working — not because it was deleted, but because its data stopped arriving or its schema changed?
DE.7	Versioning, deprecation and retirement	8%	Is content retired deliberately when it no longer earns its place, with a record of why?
DE.8	Portfolio composition and detection strategy	12%	Is the detection portfolio deliberately balanced across the pyramid of pain — or is it a pile of indicator matches with a few behavioural rules on top?
DE.9	Detection content sourcing and provenance	8%	Do you have a deliberate strategy for where detection ideas come from, and is the provenance of every deployed detection recorded?
DE.10	Detection modality breadth	7%	Does detection span the modalities the adversary can be caught in — event analytics, file and memory content, network, identity behaviour, integrity and deception — rather than relying on one?

Table 5.4 — DE sub-capabilities. Full 0–5 descriptors are in Appendix A.

Where this domain turns

The pivotal step is DE.4, testing. It is the boundary between detection writing and detection engineering. Every other practice in this domain — version control, peer review, metadata, health monitoring — is more valuable once tests exist and considerably less valuable without them, because without a test there is no definition of the rule working.

5.5 AV — Adversarial Validation & Emulation

Weight 14% · 8 sub-capabilities

The evidence engine. Every other domain makes claims; this domain tests them. It spans atomic testing, automated breach and attack simulation, threat-actor emulation, purple teaming, penetration testing and red teaming — and, critically, the loop that turns findings into closed detection gaps.

Anti-pattern. An annual penetration test whose report is read by the risk team, filed, and never converted into a single detection or a single re-test.

Ceiling rule. No other domain may score more than one level above AV. An untested capability is an assumed capability.

ID	Sub-capability	Weight	The question
AV.1	Atomic testing and control verification	13%	Are individual techniques executed safely and repeatably to verify that telemetry, detection and alerting actually fire?
AV.2	Breach and attack simulation automation	12%	Is there automated, scheduled simulation providing continuous assurance across prevention and detection layers?
AV.3	Threat-actor emulation plans	15%	Do you emulate the full behaviour chains of the specific adversaries in your threat profile, in sequence, rather than isolated techniques?
AV.4	Purple team programme	13%	Is there a structured, recurring collaboration in which offensive execution and defensive engineering work the same exercise together and fix gaps live?
AV.5	Penetration testing integration	12%	Are penetration test findings systematically converted into detection requirements — not only into vulnerability remediation tickets?
AV.6	Red teaming and independent assurance	12%	Is the detection and response capability tested by objective-based, intelligence-led adversarial engagements under realistic constraints?
AV.7	Findings-to-closure loop	13%	Do validation findings reliably become closed detection or control changes, confirmed by re-test — and is the loop's speed measured?
AV.8	Control efficacy scoring	10%	Are validation results expressed as a defensible efficacy score per technique across the prevent / detect / alert / respond chain, and used in reporting?

Table 5.5 — AV sub-capabilities. Full 0–5 descriptors are in Appendix A.

Where this domain turns

There is no single pivotal step here; the domain is the pivot. But if one sub-capability predicts the rest, it is AV.7, the findings-to-closure loop. Organisations frequently run impressive exercises whose findings are never closed and never re-tested. An exercise whose output is a report is a spending event; an exercise whose output is a closed, re-tested gap is a capability improvement.

5.6 AA — Analytics, Automation & Hunting

Weight 12% · 8 sub-capabilities

Determines whether detection output becomes usable decisions at operational tempo. Covers enrichment, correlation into attack narratives, automation, advanced analytics governance, and hypothesis-driven hunting as a producer of new detection.

Anti-pattern. A SOAR platform used exclusively to close tickets faster, and a "hunting" function that is really unstructured dashboard browsing.

ID	Sub-capability	Weight	The question
AA.1	Triage enrichment and context automation	13%	Does an alert arrive with the context an analyst needs, automatically, or does triage begin with twenty minutes of manual lookups?
AA.2	Correlation and attack-chain assembly	14%	Are related alerts assembled into a single narrative of adversary progress, or triaged as isolated events?
AA.3	Response automation and orchestration	12%	Is automation applied to response actions with appropriate safeguards, and is its coverage and value measured?
AA.4	Advanced analytics governance	11%	Where machine learning, UEBA or statistical analytics are used, are they governed, explainable and validated like any other detection?
AA.5	Threat hunting programme	15%	Is hunting hypothesis-driven, threat-informed and productive of new detection — or is it browsing?
AA.7	Deception and adversary engagement	13%	Are deception assets — honeytokens, canary credentials, decoy hosts and services, tripwire data — deliberately placed at attack-tree choke points, so that touching them is a high-fidelity signal an adversary cannot avoid without abandoning the objective?
AA.6	Case management and knowledge capture	11%	Is the knowledge generated by every investigation captured in a form that improves the next one?
AA.8	Agentic and AI-assisted operations	11%	Where AI assistants or autonomous agents take part in detection, triage, hunting or response, are they governed, bounded, auditable and measured?

Table 5.6 — AA sub-capabilities. Full 0–5 descriptors are in Appendix A.

Where this domain turns

The pivotal step is AA.5, hunting, because hunting is the only sub-capability in the model that reliably produces new detection requirements from within the organisation rather than from external reporting. A hunting programme whose hypotheses come from the threat profile, threat models and validation gaps — and whose required output is a detection, a tuning change or an evidenced negative result — is a compounding asset.

5.7 IR — Incident Response & Recovery

Weight 10% · 6 sub-capabilities

Detection without response is an expensive alarm. This domain measures the organisation's ability to act on what it detects, and — uniquely in this model — the strength of the feedback path from incidents back into detection.

Anti-pattern. A polished incident response plan that has never been exercised, and post-incident reviews that produce lessons "identified" but never engineered.

ID	Sub-capability	Weight	The question
IR.1	Response plan, playbooks and readiness	18%	Are there current, tested response procedures covering the scenarios the threat profile says are most likely?
IR.2	Detection-to-response handoff and SLAs	17%	Is the path from alert to responder defined, measured and reliable at all hours?
IR.3	Forensic readiness and evidence handling	15%	Can you acquire and defensibly preserve the evidence needed to answer what happened — including in cloud and SaaS?
IR.4	Containment, eradication and recovery	17%	Can you contain and recover at the speed and scale a real intrusion demands, and have you proven it?
IR.5	Exercising and crisis management	16%	Are response and crisis capabilities exercised realistically, including at executive level, with findings tracked?
IR.6	Post-incident review to detection backlog	17%	Does every incident measurably improve detection — and is the "why did we not see this sooner?" question answered structurally every time?

Table 5.7 — IR sub-capabilities. Full 0–5 descriptors are in Appendix A.

Where this domain turns

The pivotal step is IR.6, the post-incident review, and specifically its hardest question: at what point in the timeline could we first have detected this, and why did we not? Answering that structurally every time, and then proving the answer has changed by re-emulating the incident, is the single most reliable route from operational experience to improved detection.

5.8 GV — Governance, Metrics & Continuous Improvement

Weight 10% · 7 sub-capabilities

Determines whether the capability is deliberately directed and sustainable, or dependent on a few motivated individuals. Also covers the honesty of the metrics used to report it upward.

> **Anti-pattern.** Reporting alert volume and "threats blocked" to the board as evidence of security performance, while nobody can state how much of the threat profile is actually detectable.

ID	Sub-capability	Weight	The question
GV.1	Strategy, mandate and funding	15%	Does the detection capability have an articulated strategy, an explicit mandate and funding tied to the risks it is meant to reduce?
GV.2	Roles, skills and capability development	15%	Are the roles the model requires actually defined and staffed, with a deliberate path to build the skills the work needs?
GV.3	Metrics and performance measurement	18%	Are the metrics used to run and report the capability meaningful measures of detection effectiveness, or activity counts?
GV.4	Risk and compliance alignment	14%	Is detection capability expressed in the organisation's risk language and mapped to the obligations it must satisfy — without letting compliance drive the design?
GV.5	Executive and board reporting	13%	Do decision-makers receive an honest, comparable picture of what the organisation can and cannot detect?
GV.6	Continuous improvement cadence	13%	Is improvement a managed, funded, scheduled activity with measured outcomes?
GV.7	Third-party and supply-chain detection	12%	Does detection extend to the third parties, managed services and software supply chain through which adversaries reach you?

Table 5.8 — GV sub-capabilities. Full 0–5 descriptors are in Appendix A.

Where this domain turns

The pivotal step is GV.3, metrics. The metrics an organisation reports determine what it improves, and most report activity. Replacing alert counts with validated coverage, detection precision and time from adversary action to detection changes behaviour across every other domain — which is precisely why it is usually resisted.

6. Running an assessment

6.1 Choose the assessment type

Type	Effort	Confidence	Use it when
Rapid self-assessment	Half a day, 2–3 people	Low. Directional only.	You need a first baseline and a sense of where to look. Do not report the number outside the team.
Structured self-assessment	2–3 days across 6–10 people, evidence gathered	Moderate. Defensible internally.	Annual planning, budget cases, board reporting with the assumptions stated.
Evidence-based assessment	1–2 weeks, artefacts collected and reviewed against each claimed level	High. Withstands challenge.	Regulatory or contractual assurance, post-incident review, pre- and post-investment measurement.
Independent assessment	2–4 weeks including validation sampling	Highest. Includes spot-testing claims by emulation.	Third-party assurance, acquisition due diligence, or when internal assessments have plateaued suspiciously.

Table 6.1 — Assessment types. Confidence rises with the cost of the evidence, not with the seniority of the assessor.

6.2 Who takes part

An assessment scored by one person is an opinion. The minimum credible participation set is:

- Threat intelligence lead — domains TI, and contributes to TM and AV.
- Security architect or threat modeller — domain TM.
- Detection platform or data engineering lead — domain DC.
- Detection engineering lead — domain DE.
- Purple team, red team or offensive security lead — domain AV. If no such role exists, that is itself a finding, and AV is unlikely to score above 1.
- SOC manager and senior analyst — domains AA and IR.
- Security leadership — domain GV, and sign-off on scope and exclusions.

Scores should be proposed by the accountable owner and challenged by at least one other participant. The challenge question is always the same and always the same three words: show me the artefact.

6.3 Sequence

5. Agree and record scope, exclusions and the threat profile reference. An assessment without a scope statement cannot be compared with anything, including itself six months later.
6. Define the in-scope ATT&CK technique set from platforms and threat profile. Record the rationale. This is the step most often skipped and most often regretted.
7. Score domains in order TI, TM, DC, DE, AV, AA, IR, GV. The order matters: scoring AV before the domains it caps tends to produce defensive scoring in the earlier domains.
8. Collect evidence references as you go, not afterwards. A claim you cannot evidence in the session will not become evidenceable later.
9. Run the scoring engine or the workbook, and read the constraint log before the scores. The constraint log tells you where the organisation's self-image and its evidence diverge.

10. Produce the two reports: the engineering roadmap ranked by weighted impact, and the executive summary stating what is and is not proven.
11. Set the re-assessment date. Annual is the minimum; semi-annual is better for a programme actively investing.

6.4 Common failure modes in the assessment itself

Failure	How it shows up	Counter
Scoring the intent	'We are doing that next quarter' scored as if it were done.	Score the present tense only. Put the intent in the target column, where it belongs.
Scoring the tool	'We have a BAS platform' scored as Level 4 in AV.2.	The descriptors ask about cadence, scope and drift alerting, not licences. Read them aloud.
Best-case scoring	The best-instrumented business unit is described as if it were the estate.	Score the scope as stated. If coverage is uneven, either narrow the scope or score to the weakest material part and note it.
Vocabulary drift	'Threat modeling' meaning a risk workshop; 'hunting' meaning dashboard review.	Use the level descriptors as the definition. They exist to settle exactly this argument.
Consensus averaging	Disagreement resolved by splitting the difference.	Disagreement is information. Record both scores and the evidence each side cites, then let the evidence decide.
Assessment as performance review	Owners defending scores as if they were being appraised.	State explicitly that a low score is a funding argument, not a failure. If that is not true in your organisation, the assessment will not be honest and you should commission an independent one.

Table 6.2 — Assessment failure modes and their counters.

7. Scoring mechanics

7.1 Rollup

Sub-capability scores roll up to a domain score as a weighted mean over the in-scope sub-capabilities. Sub-capabilities marked not applicable are excluded from both numerator and denominator, so scoping something out neither helps nor harms the score — it simply removes it. Domain scores roll up to the overall score as a weighted mean using the domain weights.

The arithmetic is intentionally simple and fully transparent. A weighted mean is easy to explain to a board, easy to audit, and easy to reproduce independently. The three implementations shipped with the model — Python, Excel and JavaScript — produce identical results to two decimal places on the same input, and the test suite asserts this.

7.2 Order of operations

12. Apply C3 at sub-capability level: any score of 4 or 5 without a named evidence artefact becomes 3.
13. Compute raw domain scores as weighted means.
14. Apply C2: cap DE at DC + 1.
15. Apply C1: cap every other domain at the adjusted AV score + 1.
16. Compute the overall score from the adjusted domain scores.
17. Report the unadjusted score, the adjusted score and every adjustment made.

7.3 Bands

Overall score	Band	What it means in practice
0.00 – 0.99	Level 0 — Absent	There is no detection capability to speak of. Treat this as a build project, not an improvement project.
1.00 – 1.99	Level 1 — Ad hoc	Capability exists in individuals. It will not survive their departure, and it cannot be relied upon in a report.
2.00 – 2.99	Level 2 — Repeatable	Consistent operations driven by tooling defaults and compliance. The most populated band, and the one where spending most often outruns capability.
3.00 – 3.99	Level 3 — Threat-Informed	Work is driven by a prioritised adversary profile and traceable to it. A credible target for most organisations.
4.00 – 4.99	Level 4 — Measured & Validated	Claims are proven by emulation and trended over time. Realistic only with a standing validation function.
5.00	Level 5 — Adaptive	A closed, automated, measured loop that also contributes back. Rare, and should be treated with scepticism unless the evidence is exceptional.

Table 7.1 — Overall maturity bands.

7.4 Prioritisation arithmetic

The roadmap ranks improvement actions by weighted impact: domain weight multiplied by sub-capability weight multiplied by the gap to target, scaled for readability. This ranking is intentionally mechanical. It prevents the roadmap from being driven by whichever capability its owner argued for most forcefully, and it produces the counter-intuitive but usually correct answer that a two-level gap in a heavily weighted sub-capability outranks a four-level gap in a lightly weighted one.

The ranking is a starting point, not an instruction. Dependencies matter — improving detection engineering before fixing telemetry quality wastes effort, which is why C2 exists — and the tooling surfaces the constraint log next to the roadmap so the sequencing argument is visible.

8. The toolkit

The model ships with working tools, because a maturity model distributed as a PDF becomes a spreadsheet somebody rebuilds badly.

Artefact	What it is	Use it for
model/*.yaml	The model itself: eight domain files plus metadata, fully machine-readable, with a JSON schema.	Integration, automation, building your own tooling, proposing changes by pull request.
tidcmm (Python package)	Loader, validator, scoring engine and CLI. Implements C1, C2, C3 and the Validated Coverage Score.	Scoring in a pipeline, regression-testing your own assessments, generating reports as JSON.
TID-CMM-Self-Assessment.xlsx	Eleven-tab workbook: setup, eight domain tabs with the full descriptors as cell comments, the complete ATT&CK technique list, a dashboard with radar chart, a ranked roadmap and a crosswalk.	Running an assessment offline, in a workshop, with people who will not install anything.
tid-cmm-assessment.html	A single self-contained file — no server, no CDN, no network traffic — with the full questionnaire, live scoring, radar chart, ATT&CK coverage tab, JSON import and export, CSV export and a print view.	Distributed assessment, embedding in an intranet or the project site, sharing with a client without sending them a macro-enabled workbook.
data/attack_techniques.csv	The normalised ATT&CK Enterprise v19.2 technique set with tactics, platforms, required data components and detection guidance.	Scoping the in-scope technique set, building Navigator layers, telemetry gap analysis.
assessments/	A blank template and a fully worked example.	Starting an assessment; understanding what a completed one looks like.

Table 8.1 — What ships with the model.

8.1 Privacy

The browser tool holds everything in the page. It makes no network requests, loads no external resources and stores nothing outside the session. Export to JSON is the only way data leaves it. This is a deliberate design decision: assessment data is a detailed map of an organisation's blind spots, and it should not traverse a third party's infrastructure to be scored.

8.2 Consistency between implementations

The Python engine, the Excel workbook and the browser tool implement the same arithmetic independently. The test suite asserts agreement between them on the worked example to two decimal places, including the constraint log. This matters because the three will be used by different people in the same organisation, and a discrepancy would undermine the assessment more effectively than any methodological criticism.

9. Worked example

This chapter works through a complete assessment for an illustrative organisation, Northgate Financial Services — a mid-sized financial services group with a mature, well-funded security operations function. The figures are constructed, but the shape is drawn from a pattern that recurs constantly: strong operations, weak validation, and a self-assessment that does not survive the constraints.

9.1 Scope

Group SOC covering the UK and EU entities. The recently acquired payments subsidiary and the OT estate at two manufacturing sites are explicitly excluded, with the residual risk accepted by the group risk committee. The in-scope ATT&CK technique set is 581 of 697 techniques, derived from the platform mix (Windows, Linux, macOS, IaaS, SaaS, identity provider, office suite, containers).

9.2 Results

Domain	Self-assessed	Adjusted	Band	Constraint applied
TI — Threat Intelligence & Adversary Prioritisation	2.62	2.62	L2 Repeatable	—
TM — Threat Modeling & Attack Path Analysis	1.69	1.69	L1 Ad hoc	—
DC — Telemetry & Detection Coverage	2.84	2.63	L2 Repeatable	C1 validation ceiling
DE — Detection Engineering	2.70	2.63	L2 Repeatable	C1 validation ceiling
AV — Adversarial Validation & Emulation	1.63	1.63	L1 Ad hoc	—
AA — Analytics, Automation & Hunting	2.63	2.63	L2 Repeatable	—
IR — Incident Response & Recovery	2.86	2.63	L2 Repeatable	C1 validation ceiling
GV — Governance, Metrics & Continuous Improvement	2.57	2.57	L2 Repeatable	—
OVERALL	2.43	2.37	L2 Repeatable	

Table 9.1 — Domain results. Four domains were capped by the validation ceiling.

The organisation self-assesses at 2.43. After constraints it scores 2.37. The ten-hundredths difference is not the interesting part; the constraint log is:

> C1: DC 2.84 exceeds AV 1.63 + 1; capped to 2.63.
>
> C1: DE 2.70 exceeds AV 1.63 + 1; capped to 2.63.
>
> C1: IR 2.86 exceeds AV 1.63 + 1; capped to 2.63.

Four domains — telemetry, detection engineering, analytics and incident response — were all capped at the same value, 2.63, by the same cause: an adversarial validation score of 1.63. The organisation has built a great deal and proven very little of it. This is the finding. Everything else in the assessment is detail.

9.3 Coverage

Against the 581 in-scope techniques, the Validated Coverage Score is 47.4%. Decomposed: 48.9% of in-scope techniques have detection logic deployed, but only 12.7% have been proven to fire by emulation, and 19.3% have no telemetry at all.

> **Read that decomposition carefully.** The organisation would, on the standard industry metric, report roughly 49% ATT&CK coverage — a respectable number that would pass without challenge in most board packs. The proportion it has actually proven is 12.7%. The gap between those two figures is the entire argument for this model.

9.4 The roadmap

Impact	ID	Sub-capability	Now	Target
67.0	AV.3	Threat-actor emulation plans	1	4
58.1	AV.4	Purple team programme	1	4
57.6	TM.3	Attack tree construction	1	4
56.0	DC.5	Coverage breadth across attack surfaces	2	4
54.0	TM.4	Attack path and exposure analysis	1	4
53.6	AV.6	Red teaming and independent assurance	1	4
50.4	TM.5	Abuse cases to detection requirements traceability	1	4
48.0	TI.2	Threat profile and adversary prioritisation	2	4
44.7	AV.8	Control efficacy scoring	1	4
43.2	TI.5	Intelligence-to-detection tasking	2	4

Table 9.2 — Top ten improvement actions by weighted impact.

The ranking is unambiguous and it is not what the organisation expected. Six of the top ten items sit in threat modeling and adversarial validation — the two domains that did not exist in the organisation's previous framework and had therefore never been assessed, resourced or reported. Not one of the top ten is a tooling purchase.

9.5 What the CISO is told

> "We are at 2.34 on a five-point scale, in the Repeatable band. Our operations are sound: telemetry, detection engineering, analytics and incident response all sit close to Level 3 on their own merits. The number is held down by one thing. We have never systematically tested whether our detection works against the adversaries we say we care about, so four of our eight domains are capped by our validation score. On the coverage measure, we have detection logic for about half the techniques relevant to our estate, and we have proven about one in eight. The single highest-return investment available to us is a standing purple team and emulation capability. It will raise four domains at once, it will tell us which of the detection we already own is real, and it costs materially less than the platform renewal we are being asked to approve this quarter."

That paragraph is defensible, specific, and leads to a decision. It is also, by construction, impossible to write from alert volumes.

10. Adoption

10.1 A first ninety days

18. Week 1 — Run a rapid self-assessment with the browser tool to get a baseline. Do not report it. Its purpose is to show you which conversations you need to have.
19. Weeks 2–3 — Define the in-scope technique set properly. This is the highest-value fortnight in the programme, because everything downstream is measured against it.
20. Weeks 4–6 — Run a structured assessment with evidence, using the workbook, with the participation set from Chapter 6. Record every artefact reference.
21. Weeks 7–8 — Publish the two reports. Take the constraint log to leadership before the score; it is the part that changes decisions.
22. Weeks 9–12 — Start the top three roadmap items. If AV is your binding constraint, start with atomic testing (AV.1) — it is the cheapest sub-capability in the domain and it will immediately tell you which of your existing detection is real.

10.2 Setting a target

Level 5 is not a target for most organisations, and treating it as one produces theatre. A defensible target for a well-resourced enterprise is 3.5 to 4.0 overall, with adversarial validation at 4.0 or above so that it stops being the binding constraint. For a smaller organisation, 2.5 to 3.0 with an honest, narrow in-scope technique set is a stronger position than 3.5 over a scope chosen to flatter.

Set the target per domain, not globally. The model's tooling supports this, and it forces the useful conversation: which of these eight things do we actually need to be excellent at, given who is attacking us?

10.3 Re-assessment

Annually at minimum. Semi-annually if you are actively investing, because you need to be able to attribute movement to the investment. Keep the scope and the in-scope technique set stable between assessments; if either changes, say so explicitly and report both the like-for-like and the new-basis figures. A maturity score that moves because the scope moved is worse than no score.

10.4 Contributing

The model is open and versioned at https://github.com/ReZaAdineH/tid-cmm. Level descriptors, weights, crosswalks and the in-scope scoping guidance are all open to challenge by pull request. Anonymised assessment data, sector benchmarks and additional crosswalks are all welcome. The model will move to v1.1 on the first substantive change to any descriptor, and every version is tagged so an assessment can state which version produced it.

10.5 Licence

Model content is licensed CC-BY-4.0; code and tooling are Apache-2.0. Commercial use is permitted, including by vendors implementing the model in products, provided attribution is retained. Assessments are self-declared: there is no certification scheme, and any claim of a certified TID-CMM level should be treated as marketing.

Appendix A — Full sub-capability register

Generated from the machine-readable model. Every sub-capability with its weight, assessment question, complete 0–5 descriptors and the evidence that would justify a high score.

TI — Threat Intelligence & Adversary Prioritisation (12%)

TI.1 Intelligence requirements and PIRs · weight 18%

Question. Are there documented, prioritised intelligence requirements that state what the organisation needs to know, tied to business risk and to named decisions?

L	Descriptor
0	No intelligence requirements exist. Collection is undirected.
1	Informal requirements held by one analyst; expressed as topics of interest rather than questions.
2	A written PIR list exists and is reviewed annually, but is generic and not tied to specific decisions.
3	PIRs are decomposed into specific intelligence requirements (SIRs) and essential elements of information (EEIs), each tied to a named decision-maker and a business risk.
4	PIR satisfaction is measured — each requirement has a coverage and confidence rating, gaps are tracked, and collection is retasked on a defined cadence.
5	PIRs are dynamically re-prioritised from operational signal (incidents, hunts, emulation results, sector reporting) with measured time-to-retask.

Evidence. Signed PIR/SIR register with named decision owners · Requirement satisfaction and collection-gap tracker · Retasking log with dates

TI.2 Threat profile and adversary prioritisation · weight 20%

Question. Is there a maintained, evidenced threat profile that names the actors, campaigns and behaviours most relevant to this organisation, and ranks them?

L	Descriptor
0	No threat profile. "Everyone is a target" is the operating assumption.
1	An informal list of headline actors, largely drawn from vendor marketing and news cycles.
2	A documented threat profile exists, refreshed annually, based mainly on sector reporting.
3	Threat profile is built from sector, geography, technology stack, crown-jewel exposure and observed activity; actors are ranked by a documented relevance methodology and mapped to ATT&CK Groups and Campaigns.
4	Profile is re-scored at least quarterly against new reporting and internal telemetry; changes in ranking produce recorded downstream tasking to threat modeling, detection engineering and emulation.
5	Profile is continuously maintained, includes emerging and unattributed behaviour clusters, incorporates supply-chain and insider actors, and drives automated re-prioritisation of the detection backlog.

Evidence. Threat profile document with ranking methodology and ATT&CK Group/Campaign IDs · Quarterly re-score records · Tasking records showing profile change to backlog change

TI.3 Technical CTI ingestion and indicator lifecycle · weight 14%

Question. Are technical indicators ingested, scored, deployed, aged and retired under a defined lifecycle, with measured operational value?

L	Descriptor
0	No indicator ingestion, or manual copy-paste from emails.
1	Indicators are loaded ad hoc during incidents; nothing is retired.
2	A TIP or equivalent ingests feeds automatically; deduplication and basic scoring exist; retirement is manual and inconsistent.
3	Indicators carry confidence, source, ATT&CK context and expiry; deployment target (block, alert, enrich, hunt) is decided by score, not by default.
4	Indicator hit rates, false-positive rates and feed value are measured per source; low-value feeds are cancelled on the evidence.

L	Descriptor
5	Lifecycle is fully automated including sunset, with feedback from detection outcomes re-scoring source reliability, and internally derived indicators promoted back to the TIP.

Evidence. TIP configuration and lifecycle policy · Per-feed hit-rate/FP report · Feed decommissioning decision record

TI.4 TTP extraction and ATT&CK mapping discipline · weight 18%

Question. Is finished reporting systematically decomposed into ATT&CK-mapped adversary behaviours with enough procedural detail to build a detection from?

L	Descriptor
0	Reports are read and filed. No structured extraction.
1	Analysts occasionally note technique IDs in prose.
2	Reports are tagged with ATT&CK technique IDs, mostly at parent-technique level, stored in a searchable repository.
3	Extraction reaches sub-technique and *procedure* level — the specific command line, API call, registry path or protocol behaviour — recorded in a structured schema with source citation and confidence.
4	Extraction quality is reviewed; coverage of the prioritised threat profile by extracted procedures is measured; ambiguous mappings are arbitrated and the rationale recorded.
5	Extraction is partly automated (NLP-assisted with human validation), feeds a behaviour library reused by threat modeling, detection engineering and emulation, and is contributed to community knowledge bases.

Evidence. Structured TTP/procedure library with citations · Mapping quality-review records · Behaviour library referenced by detection tickets

TI.5 Intelligence-to-detection tasking · weight 18%

Question. Does intelligence reliably and measurably produce detection, hunting and emulation work — and can you prove the linkage?

L	Descriptor
0	No route from intelligence to engineering. The two functions do not interact.
1	Occasional informal requests, typically during a live incident.
2	A defined handoff exists (ticket or email) but is used inconsistently and without SLA.
3	Every prioritised behaviour produces a tracked work item routed to detection engineering, hunting or emulation, with a documented disposition even when the answer is "no action".
4	Time from publication to deployed-and-validated detection is measured per priority tier, trended, and reported; the backlog is triaged against the threat profile ranking.
5	Tasking is automated from the behaviour library, with measured cycle time under an agreed target and automatic escalation when a top-tier behaviour has no validated detection.

Evidence. Intel-to-detection ticket trail with dispositions · Publication-to-validated-detection cycle-time trend · Escalation records for uncovered top-tier behaviours

TI.6 Dissemination, sharing and community contribution · weight 12%

Question. Is intelligence delivered in the form each consumer can act on, and does the organisation contribute back to sector and community sharing?

L	Descriptor
0	No dissemination. Intelligence stays with the person who produced it.
1	Ad hoc emails and chat messages, one format for all audiences.
2	Regular scheduled reporting exists, but is a single product pushed to everyone.
3	Differentiated products by audience — executive risk narrative, SOC-actionable behaviour briefs, engineering-ready procedure detail — with a defined cadence.

L	Descriptor
4	Consumer feedback is collected and acted on; usefulness is measured; participation in ISAC/ISAO or sector sharing is active and reciprocal.
5	Bidirectional automated sharing (STIX/TAXII or equivalent), original research published, and community detection content contributed under an agreed disclosure policy.

Evidence. Product catalogue by audience with cadence · Consumer feedback and usefulness scores · Sharing-community membership and contribution records

TM — Threat Modeling & Attack Path Analysis (12%)

TM.1 Asset, identity and crown-jewel identification · weight 13%

Question. Do you know what you are actually protecting — the systems, data, identities and business processes whose compromise would matter most?

L	Descriptor
0	No asset inventory beyond what infrastructure teams happen to hold.
1	Partial inventories in spreadsheets, stale, no criticality rating.
2	A CMDB or asset inventory exists with ownership and basic criticality, refreshed periodically; identities and cloud resources are covered inconsistently.
3	Crown jewels are formally identified through business impact analysis and include data stores, privileged identity paths, build/CI systems and trust relationships; each has a named owner and an impact statement.
4	Inventory completeness is measured against independent discovery (network, cloud API, identity provider, EDR) and the delta is tracked and closed; criticality is reviewed on change.
5	Asset, identity and exposure inventories are continuously reconciled and automatically feed threat modeling, detection scoping and validation targeting.

Evidence. Crown-jewel register with business impact statements · Discovery-versus-inventory reconciliation report · Owner attestation records

TM.2 System and data-flow threat modeling · weight 16%

Question. Are systems threat-modelled using a recognised structured method, and does that modelling happen at the right point in the delivery lifecycle?

L	Descriptor
0	No threat modeling.
1	Occasional whiteboard sessions for high-profile projects, no method, no record.
2	A method is nominated (STRIDE, PASTA, LINDDUN, or equivalent) and used for major projects at design review; outputs are documents.
3	Threat modeling is mandatory for crown-jewel systems and material changes, uses data-flow diagrams with trust boundaries, and outputs are recorded as structured, queryable threats rather than prose.
4	Coverage of the crown-jewel estate by current threat models is measured; model quality is peer-reviewed; findings are tracked to closure with owners and dates.
5	Threat modeling is embedded in the delivery pipeline (threat-model-as-code, diagrams generated from IaC), automatically re-triggered by architectural change, and its output is machine-consumable by the detection backlog.

Evidence. Threat model repository with trust-boundary DFDs · Crown-jewel threat-model coverage metric · Threat-model-as-code artefacts and pipeline hooks

TM.7 Attack surface enumeration · weight 14%

Question. Do you continuously know every way in — external exposure, APIs, SaaS tenants, cloud services, identity federation, shadow IT and third-party ingress — so attack trees are rooted in reality rather than in the architecture diagram?

L	Descriptor
0	No attack surface enumeration. Exposure is whatever the last audit happened to notice.
1	An informal, partial picture held by individuals; discovered assets surprise the team regularly.
2	Periodic external scanning of known ranges and domains; cloud, SaaS and API exposure tracked inconsistently; shadow IT invisible.
3	Enumeration is continuous and deliberate across external services, APIs, cloud resources and identity federation, reconciled against the asset inventory; deltas are triaged and every internet-reachable path to a crown jewel is known

L	Descriptor
	and appears as an attack tree root.
4	Independent discovery (external ASM, cloud API inventory, certificate and DNS monitoring, SaaS discovery) is diffed against the declared surface; unknown-asset rate is measured and trended; new exposure automatically triggers threat model review and telemetry onboarding.
5	Attack surface change is handled as a live event — new exposure raises detection and modeling work items in near real time, pre-approved telemetry and baseline detections deploy with the asset, and surface reduction is a reported, incentivised metric.

Evidence. Attack surface register reconciled against independent discovery · Unknown-asset rate metric and trend · Change-triggered modeling and onboarding records

TM.3 Attack tree construction · weight 16%

Question. Are attack trees built for the objectives that matter — decomposing an adversary goal into the alternative branches by which it can be achieved — and are all branches carried through to a detection decision?

L	Descriptor
0	No attack trees. Threats are described as single-step statements.
1	Occasional informal "how would I break in" discussions, not recorded.
2	Attack trees are drawn for selected scenarios but stay as diagrams; leaves are not mapped to techniques or to controls.
3	Trees are built for prioritised adversary objectives (e.g. "obtain domain dominance", "exfiltrate the customer database", "tamper with the payment file"); every node is mapped to ATT&CK techniques, and every leaf carries a prevent/detect/accept decision.
4	Trees are annotated with feasibility and cost-to-adversary, are reviewed against real incident and emulation outcomes, and drive an explicitly prioritised choke-point strategy — nodes that appear in many trees are treated as high-value detection targets.
5	Attack trees are maintained as structured data (not pictures), versioned, automatically re-evaluated when the estate or the threat profile changes, and used to compute residual risk per objective.

Evidence. Attack tree library in structured form (YAML/JSON/graph DB) · Node-to-ATT&CK mapping and per-leaf control decisions · Choke-point analysis showing nodes shared across trees

TM.4 Attack path and exposure analysis · weight 15%

Question. Do you analyse real, computed attack paths through identity, network and cloud relationships in the live estate — not only hypothetical ones?

L	Descriptor
0	No attack path analysis. Exposure is understood only as a vulnerability list.
1	Path thinking happens only after a red team or pentest report describes one.
2	Point-in-time path analysis is run occasionally with a tool (identity graph, cloud permission analysis, AD path tooling) for specific reviews.
3	Path analysis runs on a defined cadence across identity, cloud entitlement, network reachability and trust relationships; results are ranked by proximity to crown jewels and issued as remediation and detection work.
4	Path exposure is trended as a metric (number and shortest length of viable paths to each crown jewel); choke points are instrumented for detection where remediation is not feasible; reduction is reported.
5	Continuous path computation is integrated with change management and CTEM cycles; new paths raise alerts in near real time and automatically create both a remediation and a detection work item.

Evidence. Attack path analysis output with paths to crown jewels · Trend of viable path count and shortest path length · Choke-point instrumentation records

TM.5 Abuse cases to detection requirements traceability · weight 14%

Question. Can you trace a specific detection rule back to the threat model or attack tree node that justified it — and identify model nodes with no detection?

L	Descriptor
0	No traceability. Detections exist for reasons nobody records.
1	Traceability exists in individuals' memory only.
2	Some detection tickets reference a threat model informally in free text.
3	A maintained traceability matrix links threat model / attack tree nodes to detection requirements, to deployed detections, and to validation results, with a unique identifier at each step.
4	Orphaned nodes (modelled but undetected and un-prevented) and orphaned detections (deployed but justified by nothing) are both reported as defects and worked down; coverage of model nodes is a reported metric.
5	Traceability is automated end to end — model node, requirement, rule, test, validation outcome and incident are linked in one queryable graph used for assurance reporting.

Evidence. Traceability matrix or graph query output · Orphaned-node and orphaned-detection reports · Assurance report tracing an incident back to a model node

TM.6 Model maintenance and change triggers · weight 12%

Question. Are threat models, attack trees and path analyses kept alive by defined triggers, or do they decay silently after first publication?

L	Descriptor
0	Models, where they exist, are never updated.
1	Updated only when someone remembers or an auditor asks.
2	A calendar-based review cycle exists (typically annual) and is partially honoured.
3	Defined triggers force review — architecture change, new crown jewel, new prioritised actor, significant incident, major ATT&CK release — and review completion is tracked.
4	Model freshness is measured (age distribution, percentage overdue), overdue models are escalated to owners, and drift between model and reality is sampled and reported.
5	Change detection is automated from CI/CD, cloud control plane and identity change events; affected models are flagged and re-validated with measured turnaround.

Evidence. Documented change triggers and review completion log · Model freshness metric and overdue escalations · Automated change-trigger integration

DC — Telemetry & Detection Coverage (14%)

DC.1 Log source inventory and ownership · weight 14%

Question. Is there a complete, owned inventory of telemetry sources with their scope, coverage percentage and criticality?

L	Descriptor
0	No inventory. Nobody can list what is being collected.
1	A partial list held by the platform team, out of date.
2	An inventory exists with source names and destinations, reviewed periodically; deployment coverage per source is estimated.
3	Each source has a named business and technical owner, defined scope (which estate it covers), measured deployment coverage, criticality rating and mapping to ATT&CK data components.
4	Inventory completeness is validated against independent asset discovery; unmonitored assets are reported as a defect class with an owner and a target date.
5	Inventory is continuously reconciled and automatically drives onboarding, alerting on unmonitored crown-jewel assets within a defined time window.

Evidence. Log source inventory with owners, coverage % and data-component mapping · Unmonitored asset report and closure trend

DC.2 Telemetry quality, completeness and timeliness · weight 18%

Question. Do you measure whether the data arriving is complete, correctly parsed, timely and unaltered — and do you alert when it is not?

L	Descriptor
0	Data quality is unknown. Gaps are discovered during investigations.
1	Occasional manual checks; problems found reactively when a search returns nothing.
2	Basic volume monitoring exists with static thresholds; parsing errors are noticed when someone reports them.
3	Quality is measured on defined dimensions — completeness, field-level fill rate, parsing success, ingestion latency, time-source accuracy, retention conformance — per source, with alerting on deviation.
4	Quality SLOs are agreed with source owners, breaches are ticketed and trended, and known-gap periods are recorded so investigations and coverage claims are adjusted for them.
5	Quality is enforced automatically — schema validation at ingest, synthetic canary events per source to prove the path end to end, self-healing pipelines, and measured mean time to detect a telemetry outage.

Evidence. Per-source data quality dashboard with SLOs · Canary event configuration and results · Telemetry outage MTTD metric

DC.3 Normalisation and data model discipline · weight 14%

Question. Is telemetry normalised to a documented, versioned data model so detections are portable and analysts are not re-learning field names per source?

L	Descriptor
0	Raw, source-specific fields only. Every search is bespoke.
1	Inconsistent ad hoc field extractions built by whoever needed them.
2	A data model is used for the main sources (OCSF, ECS, CIM, ASIM or in-house) but coverage is partial and undocumented.
3	A documented, versioned schema is mandated for onboarding; conformance is checked at onboarding; entity resolution (user, host, process, identity) is defined.
4	Schema conformance is measured continuously across all sources; non-conformant sources are tracked as debt; schema changes go through change control with impact analysis on affected detections.

L	Descriptor
5	Normalisation is automated and tested, detections are written against the model rather than the source, and the same detection content runs across multiple platforms with proven equivalence.

Evidence. Versioned schema document and onboarding conformance gate · Schema conformance metric per source · Cross-platform detection portability proof

DC.4 ATT&CK technique coverage measurement · weight 20%

Question. Is coverage measured honestly at technique and sub-technique level, grounded in data-component availability and detection validity — not in rule counts?

L	Descriptor
0	Coverage is not measured.
1	A hand-drawn Navigator layer produced once, based on opinion.
2	Coverage is mapped by tagging existing rules with technique IDs; parent-technique level only; no distinction between "we have a rule" and "it works".
3	Coverage is scored on a defined scale that separates telemetry availability, detection logic presence and detection quality; measured at sub-technique level for the in-scope set defined by the threat profile and platform mix.
4	Coverage scoring requires evidence from validation (see AV domain); the Validated Coverage Score is trended over time; regressions are investigated.
5	Coverage is computed automatically from the detection repository, telemetry health and the latest validation results, refreshed on every ATT&CK release, with automated diff reporting on new and deprecated techniques.

Evidence. Coverage model definition and scoring scale · Navigator layer generated from the detection repository · Validated Coverage Score trend and ATT&CK release diff report

DC.5 Coverage breadth across attack surfaces · weight 20%

Question. Does visibility extend across every surface the prioritised adversaries use — endpoint, identity, cloud control plane, SaaS, network, email, application, container, OT/IoT and third-party — rather than concentrating on endpoint?

L	Descriptor
0	One or two surfaces instrumented, typically endpoint and perimeter.
1	Additional sources exist but are unmonitored or only searched during incidents.
2	Most traditional surfaces are covered; cloud control plane, SaaS and identity are partial; OT/IoT and CI/CD are absent.
3	Coverage is deliberately scoped per surface against the threat profile, with documented decisions on surfaces deliberately not covered and the risk accepted.
4	Per-surface coverage is measured and reported separately, preventing a strong endpoint programme from masking a blind cloud or identity plane; gaps carry owners and dates.
5	New surfaces are onboarded as part of technology adoption governance — no material new platform reaches production without a telemetry plan and baseline detections.

Evidence. Per-surface coverage report · Accepted-risk records for uncovered surfaces · Technology-adoption gate requiring a telemetry plan

DC.6 Visibility gap management · weight 14%

Question. Are known blind spots recorded, prioritised, costed and driven to closure — or quietly tolerated?

L	Descriptor
0	Blind spots are not recorded.
1	Known informally; raised in conversation, not tracked.
2	A gap list exists but has no owners, priorities or dates.
3	Gaps are registered with the technique(s) they blind, the crown jewels affected, an owner, a priority derived from the

L	Descriptor
	threat profile, and a target date.
4	Gap closure rate and ageing are reported; gaps that cannot be closed are compensated with alternative detection or explicit risk acceptance at the right level.
5	Gaps are generated automatically from coverage and validation results, costed, and fed into budget planning with demonstrated closure of the highest-risk items each cycle.

Evidence. Visibility gap register with owners and dates · Gap ageing and closure-rate trend · Risk acceptance records for tolerated gaps

DE — Detection Engineering (16%)

DE.1 Detection lifecycle and intake · weight 10%

Question. Is there a defined lifecycle from requirement through design, build, test, release, monitor and retire — with a controlled intake?

L	Descriptor
0	No lifecycle. Rules appear when someone has an idea or a vendor ships content.
1	Informal build-and-deploy by individuals; no stages, no record.
2	A documented process exists covering build and deploy; test and retire stages are weak or skipped under pressure.
3	A full lifecycle is defined and enforced, with a single intake queue accepting requests from intelligence, threat modeling, hunting, incidents, emulation and audit — each item carrying a source and a priority.
4	Stage transition criteria are explicit and gated; lifecycle metrics (queue depth, lead time, stage ageing, rejection reasons) are measured and reviewed.
5	The lifecycle is automated end to end with policy-as-code gates; lead time from intake to validated production is measured against a target and continuously reduced.

Evidence. Documented lifecycle with stage gates · Intake queue showing source attribution per item · Lead-time and queue-depth trends

DE.2 Detection-as-code · weight 12%

Question. Is detection content managed as code — versioned, peer-reviewed, and deployed through an automated pipeline?

L	Descriptor
0	Rules are edited directly in the console. No history beyond the tool's audit log.
1	Occasional manual exports kept in a shared folder as backup.
2	Content is stored in version control but deployed manually; commits are not reviewed.
3	All detection content lives in version control with mandatory peer review, branch protection, meaningful commit history and a documented release process.
4	CI validates syntax, schema, metadata completeness and test results before merge; deployment is automated with rollback; production drift from the repository is detected and alerted.
5	Full GitOps — the repository is the single source of truth, environments are reproducible, deployments are automated with progressive rollout, and drift is auto-remediated.

Evidence. Repository with branch protection and review history · CI pipeline definition and passing runs · Drift detection alerts and rollback records

DE.3 Detection standards, metadata and documentation · weight 10%

Question. Does every detection carry the metadata needed to operate, audit and improve it — and is a shared standard enforced?

L	Descriptor
0	No standard. Rule names are the only documentation.
1	Some rules have descriptions; quality varies by author.
2	A template exists; completion is voluntary and partial.
3	A mandatory schema is enforced — unique ID, author, owner, ATT&CK technique and sub-technique, data sources required, logic rationale, known false positives, severity and risk score, triage guidance, response playbook link, validation reference, and version.
4	Metadata completeness and accuracy are measured; incomplete content cannot pass CI; analyst-facing triage guidance is reviewed for usability by the people who use it at 3am.

L	Descriptor
5	Metadata is machine-consumable and powers automated coverage reporting, response routing, validation targeting and impact analysis when a log source degrades.

Evidence. Enforced detection schema (e.g. Sigma-compatible) and CI validation rule · Metadata completeness metric · Automated report built from detection metadata

DE.4 Testing and pre-deployment validation · weight 13%

Question. Is every detection proven to fire on true positive input and stay quiet on benign input, before it reaches production?

L	Descriptor
0	No testing. Rules are enabled and observed.
1	Author eyeballs a historical search and calls it tested.
2	Manual testing against sample events for some rules; results not retained.
3	Every detection has a positive test (a reproducible execution or synthetic event that must trigger it) and a negative test set of benign activity that must not; results are recorded against the rule version.
4	Tests run automatically in CI against a representative dataset or lab range; regression tests re-run on every change and on data model changes; test coverage of the detection portfolio is measured.
5	Tests are generated from the emulation library, run continuously against production-like telemetry, and any detection without a passing test in the current period is automatically flagged as unverified in coverage reporting.

Evidence. Test definitions stored with the detection content · CI test run history and coverage-of-portfolio metric · Unverified-detection report

DE.5 Tuning, precision and false-positive management · weight 10%

Question. Is alert precision measured per detection and improved deliberately, rather than by disabling noisy rules?

L	Descriptor
0	No feedback loop. Noisy rules are muted or ignored by analysts.
1	Tuning happens reactively when analysts complain loudly enough.
2	Tuning requests are logged and actioned; changes are made in the console without a record of rationale.
3	Every detection has measured true-positive/false-positive outcomes captured at case closure; precision is calculated per rule; tuning changes are version-controlled with a stated rationale and expected effect.
4	Precision and volume thresholds are agreed; rules breaching them enter a formal remediation path with a deadline, ending in fix, demote-to-hunt, or retire; the effect of each change is measured after the fact.
5	Tuning is partly automated with statistical baselining and allow-list governance; suppression is time-boxed and expires by default; precision is trended per rule and per domain with alerting on degradation.

Evidence. Per-rule precision and volume report · Tuning change records with rationale and post-change effect · Expiring suppression policy and audit

DE.6 Detection health and silent-failure monitoring · weight 10%

Question. Would you know if a detection stopped working — not because it was deleted, but because its data stopped arriving or its schema changed?

L	Descriptor
0	No health monitoring. Silent failure is discovered during an incident, or never.
1	Occasional manual review of whether rules have fired recently.
2	Basic "rule has not fired in N days" reporting exists, treated as informational.
3	Health is monitored on multiple signals — data source availability for each rule's required components, execution errors, schema drift, scheduling failures, and unexpected volume change — with defined thresholds.

L	Descriptor
4	Health failures raise operational tickets with SLAs; the percentage of the portfolio in a healthy state is a reported KPI; dependency mapping shows which detections a given log source outage disables.
5	Health monitoring is closed-loop with canary events proving the full path from generation to alert; failures auto-open incidents, and coverage reporting automatically discounts unhealthy detections.

Evidence. Detection health dashboard with dependency mapping · Portfolio-health KPI trend · Canary-to-alert proof and auto-ticketing configuration

DE.7 Versioning, deprecation and retirement · weight 8%

Question. Is content retired deliberately when it no longer earns its place, with a record of why?

L	Descriptor
0	Nothing is ever retired. The rule set only grows.
1	Rules are occasionally disabled without record.
2	Retirement happens during periodic clean-ups, driven by performance not by value.
3	Retirement criteria are defined (superseded, permanently unsupported telemetry, technique no longer relevant, unfixable precision) and each retirement is recorded with rationale and approval.
4	The portfolio is reviewed on a cadence against the threat profile; retirement volume and reasons are reported; retired content is archived and recoverable with its history.
5	Deprecation is automated against ATT&CK changes, telemetry decommissioning and threat profile shifts, with impact analysis run before removal and coverage recomputed after.

Evidence. Retirement criteria and decision log · Portfolio review records and retirement statistics · Automated deprecation impact analysis

DE.8 Portfolio composition and detection strategy · weight 12%

Question. Is the detection portfolio deliberately balanced across the pyramid of pain — or is it a pile of indicator matches with a few behavioural rules on top?

L	Descriptor
0	No concept of portfolio. Content is whatever the tool shipped with.
1	Predominantly signature and indicator matching; behavioural detection is incidental.
2	A mix exists but is unplanned; nobody can state the balance or defend it.
3	The portfolio is explicitly classified — atomic indicator, tool artefact, behavioural/TTP, anomaly, correlation, deception — with a documented target mix, and behavioural detection is prioritised for the top-tier threat profile.
4	Composition is measured and reported against target; resilience to adversary evasion is assessed (would this survive a renamed binary, a new C2 domain, a different LOLBIN?); brittle detections are identified and reworked.
5	Detection strategy is derived from attack-tree choke points and cost-to-adversary analysis; the portfolio is optimised to raise adversary cost, and this is demonstrated through emulation results rather than asserted.

Evidence. Portfolio classification with target and actual mix · Evasion-resilience assessment · Choke-point-driven detection strategy document

DE.9 Detection content sourcing and provenance · weight 8%

Question. Do you have a deliberate strategy for where detection ideas come from, and is the provenance of every deployed detection recorded?

L	Descriptor
0	Detection content is whatever the platform shipped with. Nobody can say where a rule came from.
1	Content is copied ad hoc from blog posts and vendor reports when someone happens to read one.
2	Named sources are used routinely — vendor content subscriptions, community rule repositories — but adoption is

L	Descriptor
	uncritical and provenance is not recorded.
3	A documented sourcing strategy spans annual threat reports (for example Red Canary, CrowdStrike, Mandiant, Verizon DBIR), vendor and government advisories, community rule repositories (SigmaHQ, Elastic, Splunk Security Content, YARA collections), intelligence platforms (MISP, OpenCTI) and in-house research; every deployed detection records its source, licence and adoption date.
4	Content is evaluated before adoption against the organisation's own threat profile and telemetry — not enabled wholesale — and the value of each source is measured by the true positives and validated coverage it actually produced.
5	Sourcing is automated and bidirectional: upstream repositories are tracked for updates and deprecations with impact analysis, sector campaign reporting triggers targeted content review within a defined window, and internally developed detections are contributed back.

Evidence. Documented sourcing strategy naming the report, repository and intel-platform sources in use · Provenance, licence and adoption date recorded per detection · Per-source value report (true positives, validated coverage contributed) · Upstream change tracking with impact analysis

DE.10 Detection modality breadth · weight 7%

Question. Does detection span the modalities the adversary can be caught in — event analytics, file and memory content, network, identity behaviour, integrity and deception — rather than relying on one?

L	Descriptor
0	A single modality, almost always log or event analytics in a SIEM.
1	A second modality exists incidentally because a product provides it, but nobody plans across them.
2	Two or three modalities are in use and configured, but chosen by tooling rather than by what the prioritised behaviours require.
3	Modalities are selected deliberately per behaviour — event and process analytics, file and memory content matching (for example YARA), network signature and protocol analysis, identity and entitlement behaviour, configuration and integrity drift, and deception — with the choice recorded and justified against the threat profile.
4	Modality coverage is measured per prioritised scenario, and gaps are closed with the modality that fits rather than the tool already owned; where commercial endpoint tooling is absent, open-source equivalents are deliberately deployed to reach the same behaviours.
5	Modalities are composed rather than parallel — a single scenario is detected across several modalities that corroborate each other, raising both confidence and the cost of evasion, and the composition is validated end to end.

Evidence. Modality map per prioritised scenario with justification · Deployed content in more than one modality (for example Sigma plus YARA plus network signatures) · Evidence of corroboration across modalities in a real or emulated case

AV — Adversarial Validation & Emulation (14%)

AV.1 Atomic testing and control verification · weight 13%

Question. Are individual techniques executed safely and repeatably to verify that telemetry, detection and alerting actually fire?

L	Descriptor
0	No technique-level testing.
1	Occasional manual tests by a curious engineer, undocumented.
2	A test library (e.g. Atomic Red Team or in-house) is used sporadically against a lab; results are informal.
3	Atomic tests are run on a defined cadence against a representative production-like environment, mapped to ATT&CK sub-techniques, with recorded outcomes at each stage — telemetry generated, event ingested, detection fired, alert raised.
4	Test coverage of the in-scope technique set is measured; failures create tracked defects; re-test after fix is mandatory; results feed coverage scoring directly.
5	Atomic testing is continuous and automated with safe-execution guardrails and change control, results stream into coverage dashboards in near real time, and untested techniques are automatically reported as unproven.

Evidence. Test library mapped to sub-technique IDs · Per-stage outcome records (telemetry/ingest/detect/alert) · Defect and re-test records

AV.2 Breach and attack simulation automation · weight 12%

Question. Is there automated, scheduled simulation providing continuous assurance across prevention and detection layers?

L	Descriptor
0	No automated simulation capability.
1	A trial or proof of concept was run once.
2	A BAS tool is deployed against a limited scope, run manually and irregularly.
3	Simulation runs on a defined schedule across representative segments, covering prevention, detection and alerting layers, with scenarios selected from the prioritised threat profile rather than the vendor default set.
4	Results are trended, control drift (a previously passing test that now fails) is alerted on, and simulation scope covers all critical segments and cloud/identity planes as well as endpoint.
5	Simulation is integrated into change management — infrastructure or control changes trigger targeted re-simulation — and results are an input to control investment decisions.

Evidence. Simulation schedule, scope and scenario provenance · Control drift alerts and trend · Change-triggered simulation records

AV.3 Threat-actor emulation plans · weight 15%

Question. Do you emulate the full behaviour chains of the specific adversaries in your threat profile, in sequence, rather than isolated techniques?

L	Descriptor
0	No emulation. Testing, where it exists, is technique-by-technique only.
1	A single generic scenario borrowed from a public plan, run once.
2	Public emulation plans are executed occasionally with limited tailoring to the environment.
3	Emulation plans are authored for the top-ranked actors in the threat profile, sequencing techniques into realistic operations against realistic objectives, tailored to the organisation's platforms and crown jewels.
4	Plans are refreshed as actor tradecraft evolves; coverage of the prioritised actor set by current emulation plans is measured; detection outcomes are recorded per step in the chain, showing where in the kill chain detection actually occurs.

L	Descriptor
5	Emulation is derived automatically from the behaviour library and attack trees, includes evasion variants of previously detected behaviours to test resilience, and produces a measured "adversary dwell time before detection" per scenario.

Evidence. Emulation plan library referencing ATT&CK Group/Campaign IDs · Per-step detection outcome records showing earliest detection point · Evasion-variant test results

AV.4 Purple team programme · weight 13%

Question. Is there a structured, recurring collaboration in which offensive execution and defensive engineering work the same exercise together and fix gaps live?

L	Descriptor
0	No purple teaming. Offence and defence do not work together.
1	Occasional informal collaboration after a red team engagement.
2	Purple team exercises happen once or twice a year, ad hoc in scope, with a report at the end.
3	A defined programme with a regular cadence, scenarios drawn from the threat profile, agreed rules of engagement, and detection engineers present during execution making fixes in the session.
4	Every exercise produces measured outcomes per technique (prevented / detected-and-alerted / detected-not-alerted / logged-only / invisible), a tracked backlog, and a mandatory re-test that confirms closure.
5	Purple teaming is continuous rather than episodic, integrated with the detection pipeline so improvements are shipped within the exercise window, and its findings measurably improve time-to-detect over successive cycles.

Evidence. Programme charter, cadence and rules of engagement · Per-technique outcome matrix and re-test confirmations · Time-to-detect improvement trend across cycles

AV.5 Penetration testing integration · weight 12%

Question. Are penetration test findings systematically converted into detection requirements — not only into vulnerability remediation tickets?

L	Descriptor
0	Penetration testing is not performed, or reports never reach the detection team.
1	Tests are run for compliance; the detection team occasionally hears about the results.
2	Reports are shared with the SOC after the fact; a few detections may be built informally.
3	Every engagement has a defined detection-feedback stage — the tester's activity timeline is reconciled against SOC telemetry and alerts to determine what was seen, and each unseen action becomes a detection requirement.
4	The "detection rate" of each engagement is measured (percentage of tester actions that produced telemetry, a detection, and an alert), trended across engagements, and improvement is a stated objective of the testing programme.
5	Testers deliver machine-readable activity timelines that are automatically diffed against SIEM data; detection gaps are auto-created; scoping of subsequent tests deliberately targets previously blind areas.

Evidence. Tester activity timeline reconciled against SOC telemetry · Engagement detection-rate metric and trend · Detection requirements traced to specific test actions

AV.6 Red teaming and independent assurance · weight 12%

Question. Is the detection and response capability tested by objective-based, intelligence-led adversarial engagements under realistic constraints?

L	Descriptor
0	No red teaming.
1	A one-off engagement, scoped as an extended penetration test.
2	Periodic red team engagements with limited objectives and heavy scope restrictions; the blue team is usually informed.

L	Descriptor
3	Objective-based, intelligence-led engagements against crown-jewel objectives, with a genuinely uninformed blue team, control group, and formal rules of engagement and legal cover.
4	Engagements follow a recognised framework where applicable (TIBER-EU, CBEST, CORIE, AASE, iCAST) or an equivalent internal standard; detection and response performance is measured against defined objectives; findings drive a tracked remediation plan with executive visibility.
5	A continuous or high-frequency adversarial assurance capability exists; results are compared across cycles to demonstrate improving detection depth and reducing adversary freedom of movement; findings feed threat models and attack trees, not only detections.

Evidence. Intelligence-led engagement scope and rules of engagement · Objective-by-objective detection and response performance record · Cross-cycle comparison showing improvement

AV.7 Findings-to-closure loop · weight 13%

Question. Do validation findings reliably become closed detection or control changes, confirmed by re-test — and is the loop's speed measured?

L	Descriptor
0	Findings are not tracked. Reports are the deliverable.
1	Findings are noted in a document; closure is unverified.
2	Findings enter a tracker; closure is claimed by the assignee without re-test.
3	Every finding has an owner, a severity derived from threat-profile relevance and crown-jewel proximity, a target date, and a mandatory re-test before it can be closed.
4	Closure rate, ageing and re-test pass rate are reported; overdue findings escalate; recurrence of previously closed findings is treated as a systemic defect and investigated.
5	The loop is automated — validation results create work items, deployment triggers re-validation, and mean time from finding to validated closure is a headline KPI under active reduction.

Evidence. Findings register with re-test evidence per closure · Ageing, closure-rate and recurrence reports · Mean-time-to-validated-closure trend

AV.8 Control efficacy scoring · weight 10%

Question. Are validation results expressed as a defensible efficacy score per technique across the prevent / detect / alert / respond chain, and used in reporting?

L	Descriptor
0	No efficacy scoring. Controls are described as present or absent.
1	Subjective assessment of "good" or "weak" coverage by opinion.
2	Pass/fail per test, aggregated informally.
3	A defined scale scores each in-scope technique separately for prevention, telemetry, detection, alerting and response, based on recorded validation evidence with a stated recency window.
4	Efficacy scores expire — a result older than the defined window is downgraded to unproven; scores drive the Validated Coverage Score and appear in domain reporting; disagreements are arbitrated by evidence.
5	Efficacy scoring is automated from validation pipelines, feeds risk quantification and investment cases, and is used to demonstrate risk reduction per pound or dollar spent.

Evidence. Efficacy scoring scale definition with recency rules · Per-technique efficacy matrix · Investment case referencing efficacy-derived risk reduction

AA — Analytics, Automation & Hunting (12%)

AA.1 Triage enrichment and context automation · weight 13%

Question. Does an alert arrive with the context an analyst needs, automatically, or does triage begin with twenty minutes of manual lookups?

L	Descriptor
0	No enrichment. Analysts pivot manually across consoles.
1	A few manual lookup shortcuts maintained by individual analysts.
2	Basic automated enrichment for some alert types (reputation, geolocation, asset name).
3	Enrichment is defined per detection type and covers asset criticality and owner, identity role and privilege, recent related activity, vulnerability and exposure state, threat intelligence context and prior case history.
4	Enrichment coverage and its effect on triage time are measured; enrichment failures are monitored; the enrichment set is reviewed against what analysts actually use.
5	Enrichment is adaptive — driven by detection metadata and case type, includes automated preliminary verdicts with confidence, and demonstrably reduces mean time to triage.

Evidence. Enrichment specification per detection class · Triage-time effect measurement · Enrichment failure monitoring

AA.2 Correlation and attack-chain assembly · weight 14%

Question. Are related alerts assembled into a single narrative of adversary progress, or triaged as isolated events?

L	Descriptor
0	Alerts are handled individually with no relationship awareness.
1	Analysts manually notice patterns from experience.
2	Basic correlation by entity (host, user) within a time window.
3	Correlation assembles alerts into incidents along entity and temporal relationships, annotated with ATT&CK tactic progression so an analyst can see how far along the chain the adversary is.
4	Risk-based alerting aggregates weak signals into scored entities; the balance between raw alert volume and assembled incidents is measured; correlation quality (false grouping and missed grouping) is reviewed.
5	Graph-based correlation spans identity, endpoint, cloud and network, reconstructs full attack paths against the estate's real topology, and predicts likely next steps from attack-tree data to prompt pre-emptive containment.

Evidence. Correlation logic documentation with tactic progression · Alert-to-incident aggregation ratio and quality review · Graph correlation output showing a reconstructed path

AA.3 Response automation and orchestration · weight 12%

Question. Is automation applied to response actions with appropriate safeguards, and is its coverage and value measured?

L	Descriptor
0	All response is manual.
1	A few scripts maintained by individuals; no governance.
2	A SOAR platform exists with playbooks for a small number of high-volume alert types; mostly enrichment rather than action.
3	Playbooks cover the highest-volume and highest-severity detection types, include automated containment for defined scenarios with explicit approval gates, and are version-controlled and tested.
4	Automation coverage of alert volume is measured, along with time saved and error rate; playbook failures are monitored and remediated; blast-radius controls and rollback procedures exist and are tested.
5	Automation decisions are risk-adaptive — confidence and asset criticality determine whether an action is automatic or approval-gated — and every detection ships with a linked response action by default.

Evidence. Version-controlled playbook repository with tests · Automation coverage and time-saved metrics · Rollback and blast-radius test records

AA.4 Advanced analytics governance · weight 11%

Question. Where machine learning, UEBA or statistical analytics are used, are they governed, explainable and validated like any other detection?

L	Descriptor
0	No advanced analytics, or vendor black boxes running unmonitored.
1	Vendor ML features enabled with default settings and no evaluation.
2	Some analytics tuned by trial and error; behaviour is not understood by the team operating it.
3	Each analytic has a documented purpose, the behaviour it targets mapped to ATT&CK, its input features, training or baselining approach, expected output, and a named owner.
4	Analytics are evaluated on precision, recall and drift with a defined re-baselining cadence; outputs are explainable enough for an analyst to justify an action; failure modes and adversarial manipulation risks are documented.
5	Analytics are lifecycle-managed as models — versioned, monitored for drift and poisoning, validated by emulation like signature-based content, and retired when they stop earning their place.

Evidence. Analytic register with owners, features and ATT&CK mapping · Precision/recall/drift evaluation records · Model lifecycle and adversarial-risk documentation

AA.5 Threat hunting programme · weight 15%

Question. Is hunting hypothesis-driven, threat-informed and productive of new detection — or is it browsing?

L	Descriptor
0	No hunting capability exists in any form.
1	Occasional unstructured exploration when analysts have spare time.
2	Scheduled hunts with loose scope; findings recorded inconsistently; little output beyond "nothing found".
3	Hunts are hypothesis-driven, derived from the threat profile, threat models, attack trees and validation gaps; each has a documented hypothesis, data scope, method, result and a required output — a new detection, a tuning change, a visibility gap, or an evidenced negative result.
4	Hunt outcomes are measured — coverage of hypotheses against prioritised techniques, conversion rate to deployed detections, findings per hunt — and negative results are retained so the same ground is not re-covered blindly.
5	Hunting is continuous, partly automated (recurring hunts promoted to scheduled analytics), integrated with emulation so hunts are validated against known-good ground truth, and is a primary source of the detection backlog.

Evidence. Hunt library with hypotheses, methods and outcomes · Hunt-to-detection conversion metric · Promoted-hunt-to-analytic records

AA.7 Deception and adversary engagement · weight 13%

Question. Are deception assets — honeytokens, canary credentials, decoy hosts and services, tripwire data — deliberately placed at attack-tree choke points, so that touching them is a high-fidelity signal an adversary cannot avoid without abandoning the objective?

L	Descriptor
0	No deception capability of any kind.
1	An unofficial honeypot someone once stood up; nobody monitors it and nobody would notice it firing.
2	A small set of deception assets deployed opportunistically — some canary files or a default honeypot — alerting exists but placement is unrelated to any threat model.
3	Deception is placed deliberately at attack-tree choke points and along modelled paths to crown jewels — canary credentials where credential theft is expected, decoy shares where discovery is expected, honeytokens inside the data an adversary would steal; every asset has an owner, a high-severity alert, and a response playbook, because a deception alert is close to zero false positive.

L	Descriptor
4	Deception coverage of modelled choke points is measured; assets are refreshed so they age like the estate around them; breadcrumbs lead credibly toward the decoys rather than leaving them to be stumbled upon; triggers are exercised in purple team scenarios and the alert path is proven end to end; alert noise is measured and kept negligible, and legitimate-touch incidents are investigated as placement defects.
5	Deception is adaptive — placement is recomputed as attack trees and the estate change, interaction telemetry feeds intelligence and emulation, and measured adversary cost (time wasted, tradecraft revealed, early eviction) is reported as a programme outcome.

Evidence. Deception asset register mapped to attack-tree choke points · Choke-point coverage metric and refresh records · Purple-team validation of trigger-to-alert path · Measured alert noise rate, including periods with no trips

AA.6 Case management and knowledge capture · weight 11%

Question. Is the knowledge generated by every investigation captured in a form that improves the next one?

L	Descriptor
0	Investigations are tracked in email, chat or spreadsheets.
1	A ticketing system is used, with free-text notes of variable quality.
2	A case management platform exists with basic categorisation; quality depends on the analyst.
3	Cases carry a mandatory structure — ATT&CK classification, verdict, root cause, affected assets and identities, actions taken, and the detection(s) that fired or should have — enabling analysis across cases.
4	Case data is analysed for patterns (which detections produce value, which alert types waste time, which techniques recur) and this analysis directly drives the detection backlog and tuning priorities.
5	Case knowledge is a managed asset feeding a reusable investigation knowledge base, automated triage guidance, and analyst onboarding, with measured effect on time to competence and time to resolve.

Evidence. Mandatory case schema and completion metric · Cross-case analysis driving backlog items · Knowledge base usage and effect on resolution time

AA.8 Agentic and AI-assisted operations · weight 11%

Question. Where AI assistants or autonomous agents take part in detection, triage, hunting or response, are they governed, bounded, auditable and measured?

L	Descriptor
0	No AI or agentic assistance, or unmanaged personal use of assistants with operational data.
1	Individual analysts use general-purpose assistants informally; no policy, no record of what was pasted into them.
2	An approved assistant is available for defined tasks such as summarisation or query drafting, with a data-handling policy, but its output is unmeasured.
3	Agent roles are defined with an explicit task scope, the data they may see, the actions they may take, and the human approval points; agent-authored detection content enters the same lifecycle, testing and evidence requirements as human-authored content.
4	Agent output quality is measured against human baselines (triage accuracy, false-verdict rate, detection quality), agent actions are fully audited and attributable, authority is bounded by asset criticality and confidence, and rollback is tested.
5	Agents operate inside the closed loop with measured cycle-time benefit and no loss of assurance — their work is validated by emulation like any other detection, their identities and tool access are modelled as attack paths, and prompt-injection resistance is explicitly tested using untrusted content in the telemetry they read.

Evidence. Agent role definitions with task scope, data scope, authority and approval points · Audit trail attributing agent actions and decisions · Quality measurement against a human baseline · Prompt-injection test results using adversary-controlled log content

IR — Incident Response & Recovery (10%)

IR.1 Response plan, playbooks and readiness · weight 18%

Question. Are there current, tested response procedures covering the scenarios the threat profile says are most likely?

L	Descriptor
0	No documented incident response plan.
1	A generic plan exists, out of date, unfamiliar to the people who would use it.
2	A maintained plan with defined roles and escalation, plus playbooks for a few common scenarios.
3	Scenario playbooks are derived from the prioritised threat profile and attack trees — ransomware, business email compromise, cloud identity compromise, supply chain, insider, destructive attack, extortion without encryption — each with decision points, authority levels and communication requirements.
4	Playbook coverage of prioritised scenarios is measured; playbooks are updated after every real incident and every exercise; out-of-hours and degraded-infrastructure operation is explicitly addressed and tested.
5	Playbooks are living, partly executable (linked to orchestration), version-controlled, and validated in the same cycle as detection content, with measured readiness per scenario.

Evidence. Scenario playbook set traced to threat profile · Post-incident and post-exercise update records · Out-of-band operating procedure test

IR.2 Detection-to-response handoff and SLAs · weight 17%

Question. Is the path from alert to responder defined, measured and reliable at all hours?

L	Descriptor
0	No defined handoff. Alerts are picked up when noticed.
1	Informal handoff by chat message; coverage depends on who is awake.
2	Defined triage tiers and escalation paths; response times are not measured.
3	Severity-based SLAs exist for acknowledgement, triage and escalation, with defined coverage hours, on-call rotation and escalation-of-last-resort.
4	SLA attainment is measured per severity and reported; breaches are analysed for cause; queue ageing and abandonment are monitored; detection severity is calibrated against actual incident outcomes.
5	Handoff is automated with dynamic routing by detection type and asset criticality; time to acknowledge and time to contain are trended and actively reduced; capacity is modelled against alert volume forecasts.

Evidence. Severity/SLA matrix with coverage model · SLA attainment and queue ageing reports · Severity calibration analysis

IR.3 Forensic readiness and evidence handling · weight 15%

Question. Can you acquire and defensibly preserve the evidence needed to answer what happened — including in cloud and SaaS?

L	Descriptor
0	No forensic capability. Evidence is destroyed by remediation.
1	Ad hoc collection using whatever tools are to hand; no chain of custody.
2	Documented collection procedures for endpoints; retention adequate for common cases; cloud and SaaS evidence is uncertain.
3	Forensic readiness is planned — retention aligned to realistic dwell times, remote acquisition capability, memory capture, cloud and SaaS evidence sources identified and pre-authorised, chain of custody maintained, and legal/HR/regulatory requirements built in.
4	Readiness is tested (can you actually acquire from that cloud workload, that SaaS tenant, that OT segment, on a Sunday?); acquisition times are measured; gaps in evidential coverage are registered and closed.
5	Evidence acquisition is automated on incident declaration, forensic data is preserved before containment destroys it, and readiness is validated in every major exercise and red team engagement.

Evidence. Forensic readiness plan covering cloud and SaaS · Acquisition test results and timings · Automated preservation-on-declaration configuration

IR.4 Containment, eradication and recovery · weight 17%

Question. Can you contain and recover at the speed and scale a real intrusion demands, and have you proven it?

L	Descriptor
0	No defined containment capability; response is improvised.
1	Manual containment by individual administrators, slow and inconsistent.
2	Containment options exist for endpoints (isolate, block hash) with defined authority; identity and cloud containment are manual and slow.
3	Containment is defined across all planes — endpoint isolation, identity disablement and session/token revocation, network segmentation, cloud role and key revocation, email clawback, third-party access suspension — with authority, prerequisites and business impact documented for each.
4	Containment and recovery times are measured in exercises and real incidents; mass-scale actions are tested; recovery capability is validated against destructive scenarios including backup integrity and immutability testing.
5	Containment is largely automated with risk-adaptive gating, tested at scale on a defined cadence, and recovery objectives are demonstrated rather than asserted, including for identity infrastructure and cloud control planes.

Evidence. Containment action catalogue by plane with authority levels · Timed containment and recovery exercise results · Backup immutability and restoration test evidence

IR.5 Exercising and crisis management · weight 16%

Question. Are response and crisis capabilities exercised realistically, including at executive level, with findings tracked?

L	Descriptor
0	No exercises of any kind are conducted.
1	An occasional informal walkthrough of the plan.
2	Annual tabletop exercise, generic scenario, limited participation.
3	A programme of exercises at varying intensity — tabletop, functional, technical simulation, and executive crisis — using scenarios drawn from the threat profile and involving legal, communications, business owners and relevant third parties.
4	Every exercise generates tracked findings with owners and dates; exercise realism increases over time; participation and performance are measured; regulatory notification and disclosure decision-making is exercised under time pressure.
5	Exercises are unannounced where appropriate, integrated with red team engagements so the exercise is a real detection event, and improvement across cycles is demonstrated with objective measures.

Evidence. Multi-year exercise programme and scenario provenance · Findings register with closure · Evidence of exercises integrated with red team activity

IR.6 Post-incident review to detection backlog · weight 17%

Question. Does every incident measurably improve detection — and is the "why did we not see this sooner?" question answered structurally every time?

L	Descriptor
0	No post-incident review.
1	Informal debriefs after major incidents only; nothing recorded.
2	Reviews are held and documented for significant incidents; actions are recorded but rarely closed.
3	Every incident above a defined threshold gets a blameless review that explicitly reconstructs the timeline, identifies the earliest point at which detection was possible, and produces detection, telemetry and control actions with owners.
4	Action closure is tracked and reported; the gap between adversary first action and first detection is measured per

L	Descriptor
	incident and trended; recurring root causes are escalated as systemic issues.
5	Reviews feed threat models, attack trees, emulation plans and the detection backlog automatically; "would we detect this now?" is answered by re-emulating the incident and proving it, with the result recorded against the review.

Evidence. Blameless review records with earliest-detection-point analysis · Action closure and time-to-first-detection trends · Re-emulation proof that the incident is now detected

GV — Governance, Metrics & Continuous Improvement (10%)

GV.1 Strategy, mandate and funding · weight 15%

Question. Does the detection capability have an articulated strategy, an explicit mandate and funding tied to the risks it is meant to reduce?

L	Descriptor
0	No strategy. The capability exists as a by-product of tool purchases.
1	Direction is set informally by whoever leads the function; funding is reactive.
2	A written plan exists, largely a list of tools and headcount for the year.
3	A multi-year strategy states target maturity per domain, is explicitly derived from the threat profile and business risk appetite, and has executive sponsorship and committed funding.
4	Strategy progress is reviewed against measured maturity at least twice a year; investment decisions cite validation evidence and coverage gaps; trade-offs are documented.
5	Strategy is dynamically adjusted from threat landscape change and measured risk reduction, with funding decisions demonstrably driven by validated efficacy data rather than vendor cycles.

Evidence. Signed multi-year strategy with target maturity per domain · Investment cases citing coverage and validation data · Review minutes showing strategy adjustment

GV.2 Roles, skills and capability development · weight 15%

Question. Are the roles the model requires actually defined and staffed, with a deliberate path to build the skills the work needs?

L	Descriptor
0	No defined roles; whoever is available does the work.
1	Roles exist in name; responsibilities overlap and gaps are covered informally.
2	Job descriptions exist for core SOC roles; detection engineering, threat modeling and emulation are collateral duties.
3	Distinct roles are defined and staffed for intelligence, detection engineering, hunting, emulation/purple team and response, with documented responsibilities and a competency framework.
4	Skills are assessed against the framework, gaps drive a training plan with budget, and key-person dependency is measured and reduced; succession and cross-training are deliberate.
5	Capability development is continuous — internal ranges, rotation between offensive and defensive roles, published research — and retention and competency are measured as leading indicators of capability.

Evidence. Role definitions and competency framework · Skills assessment and funded training plan · Key-person dependency analysis

GV.3 Metrics and performance measurement · weight 18%

Question. Are the metrics used to run and report the capability meaningful measures of detection effectiveness, or activity counts?

L	Descriptor
0	No metrics beyond what tools display by default.
1	Activity counts reported — alerts handled, tickets closed, threats blocked.
2	Operational metrics exist (volumes, MTTD, MTTR) with inconsistent definitions and no target.
3	A defined metric set covers effectiveness (Validated Coverage Score, detection precision, time from adversary action to detection), efficiency (triage time, automation rate), and programme health (backlog ageing, gap closure), each with a written definition, owner and target.
4	Metrics are trended, reviewed in a formal forum, and acted on; the metric set is periodically challenged for gaming and perverse incentives; definitions are stable enough for period-on-period comparison.

L	Descriptor
5	Metrics link measured detection capability to business risk reduction and are used in prioritisation and investment; anti-gaming controls are explicit and metrics are independently verifiable from raw evidence.

Evidence. Metric catalogue with definitions, owners and targets · Trended reporting pack and review minutes · Anti-gaming review record

GV.4 Risk and compliance alignment · weight 14%

Question. Is detection capability expressed in the organisation's risk language and mapped to the obligations it must satisfy — without letting compliance drive the design?

L	Descriptor
0	No connection between detection capability and risk management or compliance.
1	Compliance obligations are met by assertion; detection is not represented in the risk register.
2	Detection appears in the risk register as a generic control; regulatory obligations are tracked separately.
3	Detection capability gaps appear as named risks with owners and treatment plans; obligations (NIS2, DORA, sector regulation, contractual) are mapped to specific model sub-capabilities; crosswalks to NIST CSF 2.0 and ISO 27001 are maintained.
4	Risk assessments cite measured coverage and validation evidence rather than control existence; residual risk per crown jewel is expressed with reference to the attack trees and validated efficacy scores.
5	Detection risk is quantified and integrated with enterprise risk quantification; compliance evidence is a by-product of the operating model rather than a separate exercise.

Evidence. Risk register entries citing coverage and validation data · Obligation-to-sub-capability mapping · Automatically generated compliance evidence

GV.5 Executive and board reporting · weight 13%

Question. Do decision-makers receive an honest, comparable picture of what the organisation can and cannot detect?

L	Descriptor
0	No reporting above the security team.
1	Occasional narrative updates, usually after an incident.
2	Regular reporting of activity volumes and tool status.
3	Reporting states maturity by domain, validated coverage against the prioritised threat profile, known blind spots, and the risks accepted as a result — in business language, with the assumptions stated.
4	Reporting is comparable period on period, includes trend and forecast, distinguishes clearly between what has been tested and what is assumed, and explicitly reports capability that has degraded.
5	Reporting supports investment decisions with modelled risk reduction per option, is independently assured, and includes benchmarking against sector peers where credible data exists.

Evidence. Executive reporting pack showing tested versus assumed capability · Period-on-period comparability statement · Independent assurance or benchmarking record

GV.6 Continuous improvement cadence · weight 13%

Question. Is improvement a managed, funded, scheduled activity with measured outcomes?

L	Descriptor
0	Improvement happens only after a serious incident.
1	Individuals improve things when time allows.
2	An improvement backlog exists but competes unsuccessfully with operational work.
3	A defined cadence exists — regular retrospectives, a prioritised improvement backlog with protected capacity, and reassessment against this model at least annually.

L	Descriptor
4	Improvement outcomes are measured against the maturity baseline; reassessment is partly independent to counter self-assessment optimism; regression is investigated as seriously as stagnation.
5	Improvement is continuous and data-driven, with capacity formally allocated, cycle times measured, and demonstrable maturity progression across multiple assessment cycles.

Evidence. Improvement backlog with protected capacity allocation · Successive TID-CMM assessment results showing progression · Independent or peer-reviewed reassessment record

GV.7 Third-party and supply-chain detection · weight 12%

Question. Does detection extend to the third parties, managed services and software supply chain through which adversaries reach you?

L	Descriptor
0	Third-party activity is outside the detection scope entirely.
1	Third-party access exists but is not distinguished in telemetry.
2	Third-party accounts are identifiable in logs; managed security providers report to defined SLAs that are not verified.
3	Third-party and vendor access paths are modelled as attack paths, monitored with specific detections, and outsourced detection responsibilities are explicitly divided in a documented responsibility matrix.
4	Provider performance is verified independently — including by emulation against provider-monitored scope — rather than accepted from their reporting; software supply chain and CI/CD telemetry is covered; fourth-party dependency risk is considered.
5	Supply-chain compromise scenarios are emulated end to end, detection responsibilities are contractually specified with testable service levels, and provider detection efficacy is measured as part of the coverage score.

Evidence. Responsibility matrix for outsourced detection · Emulation results against provider-monitored scope · Supply-chain scenario emulation records

Appendix B — Framework crosswalk

Indicative mapping at sub-capability level, so a TID-CMM assessment can feed existing NIST CSF 2.0 and SOC-CMM reporting without a second exercise. Mappings are one-to-many in both directions and are not a claim of equivalence.

ID	Sub-capability	NIST CSF 2.0	SOC-CMM	Other
TI.1	Intelligence requirements and PIRs	ID.RA-03, ID.RA-04, GV.RM-01	Intelligence.Business, Intelligence.Process	iso_27001_2022: A.5.7
TI.2	Threat profile and adversary prioritisation	ID.RA-03, ID.RA-05	Intelligence.Process	
TI.3	Technical CTI ingestion and indicator lifecycle	ID.RA-02, DE.AE-07	Intelligence.Technology	
TI.4	TTP extraction and ATT&CK mapping discipline	ID.RA-03, ID.IM-02	Intelligence.Process	
TI.5	Intelligence-to-detection tasking	ID.RA-06, ID.IM-01, DE.AE-08	Intelligence.Process, Analysis.Process	
TI.6	Dissemination, sharing and community contribution	ID.RA-03, GV.OC-02, RS.CO-02	Intelligence.Process, Business.Customers	
TM.1	Asset, identity and crown-jewel identification	ID.AM-01, ID.AM-02, ID.AM-05, ID.AM-07	Business.Customers, Process.Use case management	
TM.2	System and data-flow threat modeling	ID.RA-01, ID.RA-04, PR.PS-06	Process.Use case management	
TM.7	Attack surface enumeration	ID.AM-01, ID.AM-02, ID.AM-04, ID.RA-01, DE.CM-06	Business.Services, Process.Use case management	ctem: Scoping, Discovery
TM.3	Attack tree construction	ID.RA-01, ID.RA-04, ID.IM-02	Process.Use case management	
TM.4	Attack path and exposure analysis	ID.RA-01, ID.RA-05, ID.IM-02, PR.AA-05	Process.Use case management	ctem: Scoping, Discovery, Prioritisation
TM.5	Abuse cases to detection requirements traceability	ID.IM-01, ID.IM-02, DE.CM-09	Process.Use case management	
TM.6	Model maintenance and change triggers	ID.IM-03, ID.RA-07, GV.OV-03	Process.Use case management, Business.Governance	
DC.1	Log source inventory and ownership	ID.AM-01, DE.CM-01, DE.CM-09	Technology.SIEM tuning, Process.Log management	
DC.2	Telemetry quality, completeness and timeliness	DE.CM-09, DE.AE-03, PR.PS-04	Technology.Log management, Process.Monitoring	
DC.3	Normalisation and data model discipline	DE.AE-03, PR.DS-01	Technology.SIEM, Process.Log management	
DC.4	ATT&CK technique coverage measurement	DE.CM-09, ID.IM-02	Process.Use case management	
DC.5	Coverage breadth across attack surfaces	DE.CM-01, DE.CM-02, DE.CM-03, DE.CM-06, ID.AM-04	Technology.Log management, Business.Services	
DC.6	Visibility gap management	ID.RA-06, ID.IM-01, ID.IM-03	Process.Use case management, Business.Governance	
DE.1	Detection lifecycle and intake	DE.CM-09, ID.IM-01	Process.Use case management	debmm: Foundational to Advanced
DE.2	Detection-as-code	PR.PS-01, PR.PS-06, DE.CM-09	Technology.SIEM, Process.Use case management	
DE.3	Detection standards, metadata and documentation	DE.AE-02, DE.CM-09, RS.MA-02	Process.Use case management, Process.Detection engineering	
DE.4	Testing and pre-deployment	ID.IM-02, DE.CM-09, PR.PS-06	Process.Use case management	

ID	Sub-capability	NIST CSF 2.0	SOC-CMM	Other
	validation			
DE.5	Tuning, precision and false-positive management	DE.AE-08, ID.IM-01, RS.AN-08	Process.Monitoring, Technology.SIEM tuning	
DE.6	Detection health and silent-failure monitoring	DE.CM-09, PR.PS-04, DE.AE-03	Technology.SIEM, Process.Monitoring	
DE.7	Versioning, deprecation and retirement	ID.IM-03, PR.PS-06	Process.Use case management	
DE.8	Portfolio composition and detection strategy	DE.CM-09, ID.IM-02, PR.IR-01	Process.Use case management	
DE.9	Detection content sourcing and provenance	ID.RA-02, ID.RA-03, GV.SC-07, DE.CM-09	Intelligence.Process, Process.Use case management	
DE.10	Detection modality breadth	DE.CM-01, DE.CM-02, DE.CM-04, DE.CM-09, PR.DS-06	Technology, Process.Use case management	
AV.1	Atomic testing and control verification	ID.IM-02, DE.CM-09, PR.PS-06	Process.Use case management	
AV.2	Breach and attack simulation automation	ID.IM-02, PR.PS-06, DE.CM-09	Technology, Process.Use case management	
AV.3	Threat-actor emulation plans	ID.IM-02, ID.RA-03, DE.CM-09	Process.Use case management	
AV.4	Purple team programme	ID.IM-02, ID.IM-01, RS.MA-01	Process.Use case management, People.Training	
AV.5	Penetration testing integration	ID.IM-02, ID.RA-01, PR.PS-06	Process.Use case management	
AV.6	Red teaming and independent assurance	ID.IM-02, GV.OV-02, RS.MA-01	Business.Governance, Process.Use case management	
AV.7	Findings-to-closure loop	ID.IM-01, ID.IM-03, RS.MA-05	Process.Use case management, Business.Governance	
AV.8	Control efficacy scoring	ID.IM-02, GV.RM-06, DE.CM-09	Process.Use case management, Business.Governance	
AA.1	Triage enrichment and context automation	DE.AE-02, DE.AE-07, RS.AN-03	Process.Monitoring, Technology.SOAR	
AA.2	Correlation and attack-chain assembly	DE.AE-02, DE.AE-03, DE.AE-04, DE.AE-06	Process.Monitoring, Technology.SIEM	
AA.3	Response automation and orchestration	RS.MA-01, RS.MI-01, RS.MI-02, PR.IR-01	Technology.SOAR, Process.Incident response	
AA.4	Advanced analytics governance	DE.AE-02, DE.AE-03, GV.SC-04, ID.RA-09	Technology.SIEM, Process.Monitoring	
AA.5	Threat hunting programme	DE.AE-02, DE.CM-09, ID.RA-01, ID.IM-02	Process.Threat hunting	
AA.7	Deception and adversary engagement	DE.CM-01, DE.AE-02, ID.RA-01	Process.Threat hunting, Technology	mitre_engage: Expose, Affect, Elicit
AA.6	Case management and knowledge capture	RS.MA-02, RS.AN-03, RS.AN-08, ID.IM-04	Process.Incident response, Technology.Case management	
AA.8	Agentic and AI-assisted operations	GV.RR-02, GV.SC-04, ID.RA-09, DE.AE-02, RS.MA-01, PR.AA-05	Technology, People.Roles, Business.Governance	
IR.1	Response plan, playbooks and readiness	RS.MA-01, ID.IM-04, PR.IR-03, RC.RP-01	Process.Incident response	
IR.2	Detection-to-response handoff and SLAs	DE.AE-08, RS.MA-01, RS.MA-02, RS.MA-03	Process.Incident response, Business.Services	
IR.3	Forensic readiness and evidence handling	RS.AN-03, RS.AN-06, RS.AN-07, PR.DS-01	Process.Incident response, Technology.Forensics	

ID	Sub-capability	NIST CSF 2.0	SOC-CMM	Other
IR.4	Containment, eradication and recovery	RS.MI-01, RS.MI-02, RC.RP-01, RC.RP-05, PR.DS-11	Process.Incident response, Technology	
IR.5	Exercising and crisis management	ID.IM-02, PR.AT-01, RS.CO-02, RC.CO-03	People.Training, Process.Incident response	
IR.6	Post-incident review to detection backlog	ID.IM-01, ID.IM-04, RC.RP-06, RS.AN-08	Process.Incident response, Business.Governance	
GV.1	Strategy, mandate and funding	GV.OC-01, GV.RM-01, GV.RM-03, GV.RR-03	Business.Governance, Business.Strategy	
GV.2	Roles, skills and capability development	GV.RR-02, GV.RR-04, PR.AT-01, PR.AT-02	People.Employees, People.Training, People.Roles	
GV.3	Metrics and performance measurement	GV.OV-01, GV.OV-02, GV.OV-03, ID.IM-03	Business.Governance, Process.Reporting	
GV.4	Risk and compliance alignment	GV.RM-02, GV.RM-04, GV.OC-03, ID.RA-05, ID.RA-06	Business.Governance, Business.Compliance	
GV.5	Executive and board reporting	GV.OV-01, GV.OV-02, GV.RR-01, RS.CO-03	Business.Governance, Process.Reporting	
GV.6	Continuous improvement cadence	ID.IM-01, ID.IM-03, GV.OV-03	Business.Governance, Process	
GV.7	Third-party and supply-chain detection	GV.SC-04, GV.SC-07, GV.SC-10, ID.AM-04, DE.CM-06	Business.Customers, Business.Governance	

Appendix C — ATT&CK alignment and scoping

This version of the model is aligned to MITRE ATT&CK Enterprise v19.2, snapshot 2026-08-05.

Element	Count
Tactics	15
Techniques (total)	697
Parent techniques	222
Sub-techniques	475
Data components	109

Table C.1 — ATT&CK Enterprise content at the aligned version.

Scoping the in-scope technique set

The in-scope set is the techniques an organisation could plausibly experience and has decided to measure itself against. Build it in this order:

23. Filter by platform. Remove techniques that target platforms you do not operate. This is objective and usually removes 15–25% of the matrix.
24. Filter by threat profile. Retain techniques used by the actors and campaigns you ranked as relevant, plus techniques on the attack paths to your crown jewels regardless of attribution.
25. Add back anything that appears in your attack trees. A technique nobody has publicly attributed to your adversaries but which sits on a short path to your crown jewels belongs in scope.
26. Record every exclusion with a reason. The exclusion list is an assessment artefact and should be reviewed at every re-assessment, because platforms change faster than threat profiles.

Do not scope by ease of detection. Excluding techniques because they are hard to see is the precise failure this model exists to prevent.

On ATT&CK version changes

ATT&CK is versioned and moves several times a year. Techniques are added, deprecated, merged and revoked. Re-baseline the in-scope set on each major release, report the diff, and note that a coverage score computed against a different ATT&CK version is not directly comparable. The workbook and the tooling both state the aligned version prominently for this reason.

Appendix D — Glossary

Term	As used in this model
Adversarial validation	Any activity that executes adversary behaviour against the defended environment to test whether it is prevented, seen, detected or responded to. Spans atomic testing, BAS, emulation, purple teaming, penetration testing and red teaming.
Attack path	A computed sequence of steps through the real estate — identity relationships, entitlements, network reachability, trust — by which an adversary could reach a crown jewel from a given starting position.
Attack tree	A decomposition of an adversary objective into the alternative branches by which it could be achieved, with nodes mapped to ATT&CK techniques and leaves carrying a prevent, detect or accept decision.
Choke point	A node appearing across many attack trees or paths, and therefore worth disproportionate detection or prevention investment.
Crown jewel	An asset, dataset, identity or process whose compromise would cause material business harm, identified through business impact analysis rather than by technical criticality alone.
Detection-as-code	Managing detection content with software engineering practice: version control, peer review, automated testing, CI/CD deployment and drift detection.
Emulation plan	A sequenced set of techniques reproducing a specific adversary's tradecraft against a realistic objective, as opposed to isolated technique execution.
In-scope technique set	The subset of ATT&CK techniques an organisation measures itself against, derived from its platforms and prioritised threat profile, with the rationale recorded.
Silent failure	A detection that no longer works because its data stopped arriving, its schema changed, or its schedule broke — while continuing to appear enabled and healthy.
Threat profile	The ranked, evidenced set of adversaries, campaigns and behaviours most relevant to a specific organisation, derived from sector, geography, technology and exposure.
Validated Coverage Score	Achieved points over maximum available across the in-scope technique set, on the 0–3 scale in Table 4.3, expressed as a percentage.
Strict mode	Scoring with the three integrity constraints applied. The default, and the only mode in which a score should be reported externally.

TID-CMM

Threat-Informed Detection Capability Maturity Model

Version 1.5.0 · 2026-08-17

https://tid-cmm.com

https://github.com/ReZaAdineH/tid-cmm

hello@tid-cmm.com

Model content CC-BY-4.0 · Tooling Apache-2.0

MITRE ATT&CK® is a registered trademark of The MITRE Corporation. This work is not affiliated with or endorsed by MITRE, NIST or SOC-CMM.