OSMS ACADEMY · FUNDAMENTALS

# OSMS Fundamentals

The open standard for security metrics - terms, principles, answers.

Free foundation material for CISOs, SOC teams, GRC & audit.

opensecuritymetrics.org As of July 2026 · free to share

# Security is measured everywhere - just never the same way

**No comparability** - Three teams, three definitions of “coverage”. The same metric means something different in every company - often in every department.

**Painful audits** - Every review starts from zero: what does the number mean, where does it come from, why this threshold? Without a normative definition, every answer is manual work.

**Boards without confidence** - Metrics without denominators, averages without distributions, green without data trust - decisions built on apparent precision.

**The root cause is rarely the tooling - what’s missing is a shared definition.** That is exactly what OSMS provides: an open standard that defines how security metrics are specified, calculated and reported.

# A standard that normalizes metrics - not yet another framework

**What** - An open, vendor-neutral standard for security metrics. Every metric is a normative “card” with formula, denominator, threshold guidance, owner and framework anchor.

**Why** - An end to “everyone measures differently”: comparable across teams and companies, auditable against standards, board-ready through drill-down.

**How** - Machine-readable (YAML/JSON), round-trip validated, star schema for reporting. The standard is code, not just prose.

### THE 30-SECOND PITCH

“OSMS defines security metrics so that two companies mean the same thing by the same number - with formula, denominator, threshold logic and audit anchor. Machine-readable, framework-anchored, explainable down to the raw data row.”

**What OSMS is NOT:** not a tool · not a framework replacement (it operationalizes NIST/ISO/CIS) · not a compliance checklist · not a benchmark report

AUDIENCE

# Four perspectives - one shared standard

## CISO & management

Defensible numbers for the board: a cockpit whose values can be explained down to the raw data row - and trends that survive definition changes.

## SOC & detection teams

Clear measurement rules instead of debates: defined denominators, reproducible calculations, guardrails against perverse incentives.

## GRC, audit & regulatory

Every card carries framework anchors (NIST CSF, ISO, CIS, DORA/NIS2) - evidence becomes derivable instead of renegotiated every time.

## Consultants & vendors

An open, machine-readable target format: build analyses, dashboards and products against a standard instead of house definitions.

FOUNDATION

# The six principles - everything hangs on these

## Denominator discipline

Measurement runs against an authoritative population. Without a trustworthy denominator, every percentage is fiction.

## Framework anchoring

Every card carries ≥ 1 framework anchor (NIST CSF, ISO, CIS, ATT&CK, DORA/NIS2) - audit defense built in.

## Decision orientation

Every card has a decision chain and a management_question. A metric without a decision is decoration.

## Honest statistics

P50/P90 instead of averages, direction instead of “more is better”, a confidence gate instead of fake green.

## Anti-gaming

Guardrails against perverse incentives (Goodhart’s law): metrics should steer, not flatter.

## Machine readability

YAML/JSON as the source of truth, star schema, lossless reconciliation from board to raw data row.

BUILDING BLOCKS

# Card → Domain → Catalog - and the toolchain behind them

### Card

The normative unit: ONE metric, fully defined (47 mandatory fields). Stable IDs with prefixes like STD- / SOC- / AI- / HRM-.

### Domain

A subject area that bundles cards - e.g. SOC – Detection & Response, Cloud Security, IAM, Governance & Compliance.

### Catalog

All cards in one machine-readable file - over 300 cards (as of July 2026). Latest version: opensecuritymetrics.org.

## THE TOOLCHAIN

**osms-catalog.yaml**
The source of truth - all cards, all fields.

**drill_spec.json**
Drill-down mechanics per archetype (calculation_type).

**star_schema.sql**
Reporting schema with GROUP-BY reconciliation.

**drill_engine.py**
Reference implementation - validates the round trip.

**Mnemonic:** “The standard is code, not just prose” - implementers parse the YAML instead of retyping a PDF.

TERMS I

# The six metric types - who measures what?

## KPI

**Key Performance Indicator**

Measures the performance and outcome of a process. Question: “How well are we running?”

*e.g. MTTR, patch SLA*

## KRI

**Key Risk Indicator**

Warns before it gets expensive - rising risk. Question: “Where will it burn next?”

*e.g. EOL exposure, stale log sources*

## KCI

**Key Control Indicator**

Proves a control exists and works. Question: “Is the control biting?”

*e.g. log source coverage*

## Outcome

**Outcome Metric**

The security outcome that actually occurred. Question: “What really happened?”

*e.g. confirmed exfiltration*

## Confidence

**Confidence Metric**

How trustworthy are data & denominator? Question: “Do we trust the number?”

*e.g. asset inventory completeness*

## Maturity

**Maturity Metric**

How mature is the process (e.g. CMMI logic)?

*e.g. PQC readiness*

TERMS II

# Lagging vs. leading - proving vs. preventing

LOOKING BACK

## Lagging

Measures what has happened. Proves impact and responsiveness - in hindsight.

*e.g. MTTD, MTTR, repeat-incident rate*

LOOKING AHEAD

## Leading

Measures preconditions before anything happens. Prevents - steerable in advance.

*e.g. coverage KCIs, baseline compliance, scan cadence*

**“Leading prevents, lagging proves.”** The catalog is deliberately balanced: lagging metrics prove the response - the leading layer of coverage and denominator KCIs is what makes steering possible in the first place.

# P0 / P1 / P2 and MVP-1/2/3 - how important vs. when

## P0

**Priority: MUST**

Core set - without these cards there is no credible reporting.

## P1

**Priority: SHOULD**

Important - second implementation stage after the core.

## P2

**Priority: COULD**

Extension stage - depth and special cases.

**MVP - Minimum Viable Product → the smallest working state.** In OSMS, MVP is a dedicated card field with three build-out stages: MVP-1 = go-live set, MVP-2 = second wave, MVP-3 = full build-out. **Mnemonic: P says HOW IMPORTANT a card is - MVP says WHEN it gets implemented. The two correlate strongly but are not identical.**

*Answer to "Where do I start?": implement the MVP-1 set, P0 first - go live, then add MVP-2/3.*

# P50 / P90 - why the average lies

A percentile is the value below which X% of all cases fall. P50 = median = “the typical case”. P90 = “the bad day” - only 10% take longer.

**Mean 6.6 h**
average - distorted by the outlier

**P50 = 4 h**
median - the typical case

**P90 = 9 h**
90% are faster - the SLA anchor

**Example: 10 incidents, resolution time sorted (h)**


**OSMS rule:** duration metrics report P50 + P90 - never just the mean. SLAs belong on P90; the outlier beyond P90 gets its own review. The same logic applies in risk quantification (FAIR): loss estimates as P50/P90 ranges instead of point values.

# Card anatomy - the mandatory blocks (real field names)

### 1 Identity & labels

id · name · domain · type · lag_type · priority · mvp · card_version · status · scope · unit

### 2 Purpose & question

purpose · definition · story · management_question (“which management question does the card answer?”)

### 3 Calculation

calculation · calculation_type · calculation_plain · variables_inputs · numerator_denominator · rebuild_steps

### 4 Evaluation

target_thresholds · threshold_mode · direction · interpretation · special_cases_gates

### 5 Ownership & cadence

accountable_owner · escalation_owner · frequency

### 6 Data & trust

data_sources · min_data_fields · data_confidence · confidence_production_rule · reproducibility

### 7 Impact & anti-gaming

decision_chain · next_chain · example_tasks · guardrails

### 8 Integrity & audit

standards (framework anchors) · drilldown_lineage · version_break_rule

**Definition of done: all blocks filled + round-trip validated = the card is “normative”.**

CALCULATION

# calculation_type - the calculation archetypes

## Ratio

**quotient / share**

Numerator ÷ denominator × 100. The workhorse - most of the catalog calculates this way.

*e.g. coverage %*

## Composite

**composite index**

Several sub-scores condensed into one value.

*e.g. EDR effectiveness*

## Duration

**time span**

Time spans - reported as P50/P90, never just the mean.

*e.g. MTTR, dwell time*

## Weighted Sum

**weighted sum**

Partial values × weights, summed up.

*e.g. weighted target*

## Count

**count**

Absolute counting - often with zero_is_target.

*e.g. open P0 findings*

## Weighted Average

**weighted average**

Mean with per-element weighting.

*e.g. weighted maturity*

13 calculation_types in total: the six above plus Unit Cost · Delta · Penalty · Index · Score · Monetary Risk · Ranking. The drill framework maps all of them onto 6 generic mechanics (ratio · duration · count · delta · component_tree · ranking) - the archetype determines the drill-down logic in drill_engine.py.

PRINCIPLE #1

# Numerator ÷ denominator - the denominator is the standard

**Numerator** (what is measured) · **Denominator** (the authoritative population measured against)

**Effective Log Coverage (%)** = onboarded **AND** healthy ÷ in-scope assets per logging baseline × 100

**Where does the denominator come from?** - Inventory (CMDB) × policy/baseline: which asset class MUST deliver what? That is the authoritative "should".

**Decomposition, not competition** - Layers compose into the headline: depth × health = 50% × 97.5% = 48.75% - exactly 3,900 ÷ 8,000.

**Who guards the denominator?** - Confidence metrics (e.g. inventory completeness) - they measure whether the population can be trusted.

**"Without a trustworthy denominator, every percentage is fiction."**

EVALUATION I

# direction & threshold_mode - what does “good” mean?

| higher_is_better | lower_is_better | context_dependent | zero_is_target | band_is_better |
|---|---|---|---|---|
| **higher = better** | **lower = better** | **context only** | **target is 0** | **target band** |
| *coverage* | *MTTR, backlog* | *volume, cost* | *overdue P0 findings* | *over- & under-steering critical* |

**threshold_mode - the threshold principle:** OSMS prescribes NO universal thresholds. The guide provides guidance - thresholds only become binding through local anchoring to risk appetite, service criticality and data trust. Helper cards are not independently reportable; they inherit thresholds from their main card.

*Answer to “Which target does OSMS prescribe?”: none fixed - guidance yes, anchoring to your own risk appetite is mandatory. A standard demanding 95% everywhere would not be serious.*

# RAG logic & the data-confidence gate

RAG = Red / Amber / Green - traffic-light logic per card (target_thresholds). Example guidance:

| Scope | Green | Amber | Red |
|---|---|---|---|
| Critical assets | ≥ 95% | 85–94% | < 85% |
| Overall estate | ≥ 90% | 80–89% | < 80% |

*Important: these are example bands (guidance) - anchor them locally to risk appetite (threshold_mode).*

Plus direction: with lower_is_better the traffic light flips - the logic stays the same.

## DATA-CONFIDENCE GATE

**Green only comes with trustworthy data.**
Fields: data_confidence + confidence_production_rule + min_data_fields. The card is only "produced" when minimum data fields and confidence criteria are met - otherwise it counts as not defensible (no fake green).

Typical question: *"The value looks good - why isn't the light green?"*
**Answer: because the denominator is shaky. Establish data trust first, then color.**

IMPACT

# Decision chain, next_chain & guardrails

## DECISION CHAIN

**Green → observe**
Keep trend and context in view - no action needed.

**Amber → owner + deadline**
accountable_owner acts, escalation_owner is informed.

**Red → escalation**
Ticket, named owner, tracked to closure - predefined, not improvised.

**next_chain → follow-up**
Which metric/question comes next? Metrics chain into steering.

> "A metric without a decision is decoration."

## GUARDRAILS - AGAINST GAMING

- Never report onboarding without health - otherwise "connected but silent" sources count as success.
- Utilization only against a curated, security-relevant set - otherwise the ratio punishes chatty sources.
- "Used" means: referenced in an active, tested rule - otherwise the ratio invites dummy rules.

**Goodhart's law:** "When a measure becomes a target, it ceases to be a good measure." Guardrails are the built-in countermeasure.

AGGREGATION

# From event to board - and back without loss

**1**

### Raw data / events

SIEM, EDR, cloud logs - the lowest truth.

**2**

### Operational cards

Individual metrics per domain (SOC-/STD-/AIM-/HRM- IDs).

**3**

### Domain roll-up

Condensed per subject area - same formula logic.

**4**

### Board cockpit (L4)

9 canonical KPIs - the management view.

**Reconciliation guarantee:** every aggregated number decomposes losslessly into its parts (GROUP-BY guarantee in the star schema). drilldown_lineage documents the path per card - every board number can be explained down to the raw data row. That is the difference between a dashboard and a standard.

*Board question “Where does this number come from?” → an answer in seconds instead of a special analysis.*

ANCHORING

# standards - the framework anchors of every card

### NIST CSF 2.0

Function model GV · ID · PR · DE · RS · RC - the backbone of the mapping.

### ISO/IEC 27001

ISMS anchor - management system & controls (Annex A).

### CIS Controls v8.1

Concrete, prioritized controls - the operational bridge.

### ISO/IEC 27004

The measurement methodology standard - how to measure security.

### NIS2 / DORA

EU obligations: reporting deadlines (24h/72h), registers, TLPT, resilience.

### MITRE ATT&CK

Techniques & data sources - the basis for detection coverage.

**Function:** every card carries ≥ 1 anchor in the standards field → audit defense built in. OSMS does not replace these frameworks - it makes measurable whether their requirements are met. In progress: CRA, EU AI Act.

STATUS & PATH

# A living standard - versions, waves, stability

## v0.9

**Base release**

Core catalog, domain structure, toolchain — public review extended to 31 Oct 2026.

## v1.0

**Foundation + regulatory**

Coverage/denominator layer, DORA & NIS2 mandatory metrics.

**Target: end of November 2026**

## v1.1

**Operational depth**

Detection, data, network, supply-chain and resilience families.

## v1.2

**Scope extension**

AppSec/CRA, OT/ICS, privacy ops - as deliberate scope decisions.

Stability promise: card IDs are never renumbered; definition changes break trends in a controlled way (version_break_rule) - investments in OSMS stay valid.

Conformance annex: meta-metrics test the measurement system itself (denominator drift, confidence gate, owner attestation) - OSMS is a self-testing standard.

The catalog is growing: over 300 cards (as of July 2026). Latest version, changelog and roadmap: opensecuritymetrics.org.

GET STARTED

# Four steps to your first OSMS reporting

### 1 · Get the catalog

kpi_catalog.yaml - machine-readable, versioned, with all mandatory fields.

### 2 · Pick the starter set

MVP-1 set, P0 first - the go-live set for your first reporting.

### 3 · Fix denominators & thresholds

Inventory × policy as the denominator; anchor thresholds to your risk appetite.

### 4 · Report & steer

A fixed cadence - with confidence gate, decision chain and drill-down.

**Open & free.** OSMS is vendor-neutral, machine-readable and freely accessible. Catalog, documentation, terms of use and updates: **opensecuritymetrics.org**

*Questions, feedback or interest in the OSMS Academy? → Contact via the website.*

## IF YOU TAKE AWAY JUST FIVE SENTENCES

1. “The denominator is the standard.”
2. “Leading prevents, lagging proves.”
3. “P50 is the everyday, P90 is the bad day.”
4. “A metric without a decision is decoration.”
5. “No green without trustworthy data.”

Catalog, docs & updates: **opensecuritymetrics.org**