SECURITY METRICS × CLAUDE · A FIELD GUIDE FOR PRACTITIONERS

# Make Security Metrics Talk.

Generate metric data from the systems you already run. Know what good looks like — and what bad looks like — for every area of information security. Read trends, connect policy changes to outcomes, and turn it all into a story leadership actually understands.

## THE FIVE SHORTCUTS INSIDE

| | | | | |
|---|---|---|---|---|
| `/god` | `/ghost` | `/artifacts` | `/blindspot` | `/deepdive` |

BUILT FOR SECURITY LEADERS, ANALYSTS, AND ANYONE WHO REPORTS UPWARD

Share it with your team · Save this

01 · WHY THIS GUIDE EXISTS

# Metrics are a language. Most programs mumble.

Every security program produces numbers. Very few produce **meaning**. The monthly slide says “14,302 alerts, 96% patched, 12 incidents” — and leadership nods, learns nothing, and funds nothing. This guide is about the other way: metrics that show whether you’re getting better or worse, that connect the policy change you fought for to the outcome it produced, and that tell a story an executive can repeat to the board without you in the room.

Claude’s role in this: it can compute the metrics from raw exports, benchmark them against what good looks like, find the trend you’d miss, test whether a change actually moved the needle, and draft the narrative — in minutes, not the last week of the quarter.

> *“A metric without a comparison is a number. A metric with a target, a trend, and a cause is a story.”*

## The five questions every metric must answer

- **What is it?** — defined precisely enough that two people compute the same number.
- **Compared to what?** — a target, a baseline, a benchmark, or last quarter.
- **Which way is it moving?** — trend over at least 6–12 data points, not a single snapshot.
- **Why?** — the change, event, or decision that explains the movement.
- **So what?** — the decision this number should drive. If there isn’t one, drop the metric.

**NON-NEGOTIABLE GROUND RULE BEFORE ANYTHING ELSE**

- Sanitize exports before uploading: strip usernames, hostnames, and anything sensitive — analyse patterns, not persons. Follow your organization’s AI acceptable-use policy.
- Never let AI invent a number. Every figure in a report must trace back to a real export Claude actually computed from — ask it to show its work.
- AI-drawn conclusions about causation are hypotheses. You confirm them against what actually happened operationally.

02 · CONTENTS

# What’s inside

Nine workflows plus the shortcut layer. Together: a metrics program that earns budget instead of filling slides.

01

FOUNDATION

# Designing Metrics Worth Collecting

*If a metric can't change a decision, it's decoration.*

Before generating anything, decide what deserves measuring. Two distinctions do most of the work:

| | | | |
|---|---|---|---|
| **KPI (performance)**<br>How well is the program operating? Patch latency, MTTR, training completion. | **KRI (risk)**<br>How exposed are we? Unmanaged assets, stale admin accounts, critical vulns past SLA. | **LEADING**<br>Predicts future outcomes — phishing report rate, control coverage, time-to-onboard logging. | **LAGGING**<br>Measures past outcomes — incident count, breach cost, audit findings. |

COPY-PASTE PROMPT · BUILD YOUR METRICS CATALOG

```
You are a security metrics architect who designs measurement programs
that drive decisions, not slide decks.

Design a metrics catalog for a security program covering: vulnerability
management, incident response, identity & access, phishing/awareness,
endpoint coverage, and third-party risk.

For each domain give me 3-4 metrics. For EACH metric specify:
- Precise definition and formula (numerator / denominator / time window)
- Type: KPI or KRI, leading or lagging
- Data source it's computed from
- The decision it should drive
- What "good" direction looks like, and one way the metric can be gamed

Cut anything that's a vanity metric. Output as a table I can put in a
spreadsheet.
```

### DESIGN RULES THAT SAVE YOU LATER

- Write the formula down — "patch compliance" means nothing until numerator, denominator, and window are fixed.
- Pair every volume metric with a quality metric (alert count *with* true-positive rate), or volume will get gamed.
- Fewer is better: 12–18 metrics with owners beats 60 nobody defends.
- Decide the target and the threshold for action when you define the metric — not after you see the number.

02

GENERATION

# Generating Metric Data from Raw Exports

*The metrics already exist inside your exports. Claude does the extraction.*

You don't need a BI project to get started. Every tool you run — SIEM, vulnerability scanner, IdP, EDR, phishing platform, ticketing — exports CSV or Excel. Upload the sanitized export and have Claude compute the catalog metrics, consistently, every month.

## When to use it

- Monthly metric computation — same definitions, same formulas, zero spreadsheet drift
- Backfilling history — compute 12 months of metrics from old exports to establish a baseline
- Merging sources — vuln data + asset inventory to get coverage-aware metrics instead of raw counts
- Data-quality checks — find the gaps that silently corrupt metrics (assets missing from scans, tickets without timestamps)

**COPY-PASTE PROMPT · COMPUTE THE MONTHLY METRICS**

```
You are a security metrics analyst who computes precise, reproducible
metrics from raw exports.

I've uploaded [e.g. "our vulnerability scanner export (sanitized) and our
ticketing export for May"]. Here are the metric definitions from our
catalog: [paste definitions, with formulas].

Compute each metric exactly as defined. Then:
1. Show your work: row counts, exclusions applied, and the formula used
2. Flag data-quality issues — missing fields, duplicates, impossible
dates — and tell me how they affect each number
3. Compare against last month's values, which I'm pasting here: [...]
4. Output a clean metrics table (metric, value, prior value, delta,
target, status) I can paste into our tracker.

If a metric can't be computed from this data, say so — do not estimate.
```

**THE TWO RULES OF GENERATED METRICS**

- **Reproducibility:** same export + same definition = same number, every time. Keep the definitions in a shared Project so every team member computes identically.
- **Denominator honesty:** "96% patched" of *what*? Always anchor to a full asset inventory, and report coverage ("of 2,140 known assets, 1,980 in scan scope") alongside the rate.

03

BENCHMARKS

# What Good Looks Like, by Domain

*Reference points to calibrate against — then beat your own baseline.*

There is no universal “good” — targets depend on size, sector, and risk appetite. Use the reference points below to see which neighborhood you’re in, then set targets against your own baseline.

| DOMAIN | METRIC TO WATCH | HEALTHY SIGNAL | WARNING SIGNAL |
|---|---|---|---|
| **Vulnerability mgmt** | Time to remediate critical / KEV-listed vulns | Criticals on internet-facing assets fixed in days, not months; KEV items treated as priority with near-total SLA adherence | Criticals aging 90+ days; SLA met on lows while criticals slip |
| **Incident response** | MTTD / MTTR by severity | Detection in hours for high-severity; containment trending down quarter over quarter | MTTR flat or rising; severity downgraded to make SLAs |
| **Phishing & awareness** | Report rate vs. click rate | Report rate climbing and exceeding click rate; repeat-clickers shrinking | Click rate “improves” only because simulations got easier |
| **Identity & access** | Privileged accounts, MFA coverage, dormant accounts | MFA on ~100% of remote/privileged access; dormant accounts removed within days; admin count flat or falling as company grows | Admin accounts growing faster than headcount; service accounts with no owner |
| **Endpoint / coverage** | % of known assets with EDR + logging onboarded | Coverage in the high 90s of a *verified* inventory, gap list owned and shrinking | Coverage “100%” of an inventory nobody trusts |
| **Patch & config** | Patch latency by ring; config baseline drift | Patch latency measured in days for critical rings; drift detected and reverted automatically | Latency unknown; baselines defined but never re-checked |
| **Third-party risk** | Vendors assessed / reassessed on schedule | Tiered assessments current; critical vendors with tested exit and incident clauses | Assessments done once at onboarding, never again |
| **Security ops** | Alert true-positive rate; automation rate | TP rate rising as tuning matures; routine triage increasingly automated | Alert volume celebrated as productivity; backlog quietly growing |

COPY-PASTE PROMPT · BENCHMARK MY NUMBERS

```
You are a security metrics advisor with deep knowledge of industry
benchmarks and maturity models (NIST CSF, CIS Controls).

Here are our current metrics: [paste the table from Module 02].
Context: [size, sector, regulatory environment].

For each metric: (1) strong, typical, or weak for an org like ours, and
why; (2) the benchmark you're anchoring on — search for current data,
and say plainly when none exists; (3) a realistic target two quarters
out. Do not flatter — anchor on our baseline where benchmarks are
thin.
```

04

ANTI-PATTERNS

# What Bad Looks Like

*Bad isn't just a red number. It's a number that lies politely.*

The dangerous metrics aren't the ones flashing red — those get attention. The dangerous ones look fine while hiding decay. Train yourself (and Claude) to spot these patterns:

## The classic failure patterns

- **Vanity metrics** — "2 million attacks blocked": big, impressive, decision-free. If nothing changes when the number changes, cut it.
- **Goodhart's law** — the metric became the target and got gamed: tickets closed-and-reopened to hit MTTR, severities downgraded to meet SLA, easier phishing templates to lower click rate.
- **Denominator drift** — patch compliance "improved" because unscanned assets fell out of inventory. The rate rose; the risk didn't.
- **Survivorship reporting** — only teams with good numbers report; the dashboard becomes a highlight reel.
- **Snapshot theater** — a single point in time, no trend, conveniently captured right after the cleanup sprint.
- **Averages hiding tails** — mean remediation looks fine while the worst 5% of criticals age past 180 days. The tail is where breaches live.
- **Coverage blindness** — every percentage quoted against "managed assets" while the unmanaged population grows.

### A NOTE ON COLOR — MAKE RED MEAN SOMETHING

- In any chart, dashboard, or RAG status, red should be rare and earned. If half the dashboard is red every month, red has stopped meaning anything and people tune it out; if nothing is ever red, the dashboard is decoration. Reserve red for the small number of metrics in this module's failure patterns — genuinely deteriorating trends, SLA breaches on critical items, or risk that is actively growing.
- Use amber/orange for "watch this" and green sparingly for "genuinely improving," not just "technically met target." When asking Claude to build dashboards (Module 08) or decks (Module 07), say so explicitly: "limit red to items that need executive attention this cycle."

COPY-PASTE PROMPT · INTERROGATE A METRICS PACK

```
You are a skeptical security metrics auditor whose job is to find where
this report flatters itself.

I've uploaded our quarterly security metrics pack [or: pasted the table].

For each metric, check for: vanity metrics, gamed targets (Goodhart
patterns), denominator drift, averages hiding bad tails, missing trend
context, and percentages without coverage statements.

Then give me: (1) the three numbers most likely misleading, and the
question to ask the owner; (2) what's MISSING — risks this pack is
silent about; (3) a rewritten version of the worst metric, done
```

```
honestly. Be blunt — I'd rather hear it from you than an auditor.
```

### A SIMPLE HONESTY TEST FOR ANY METRIC

› Could this number improve while real risk gets worse? If yes, name the mechanism — then add the paired metric that would catch it.

› Would you show this exact chart in the post-incident review of a breach in that domain? If it would look absurd there, it’s decoration now.

05

TRENDS

# Reading Trends Honestly

*Direction beats position. Twelve points beat one.*

Leadership doesn't need to know the number is 87. They need to know it was 71 two quarters ago, why it moved, and where it's heading. Upload your metric history and let Claude do the time-series read — including the part where it tells you a "trend" is actually noise.

## When to use it

- Quarterly reads — which metrics genuinely moved vs. wobbled within normal variation
- Seasonality — phishing clicks spike during open enrollment; alert volume dips over holidays; don't panic or celebrate the calendar
- Leading-indicator checks — is the report rate rising *before* the click rate falls, as it should?
- Early-warning — small consistent drifts (dormant accounts +4% three months running) that single snapshots hide

**COPY-PASTE PROMPT · TREND READ**

```
You are a security data analyst who reads time series honestly —
including telling me when there is no trend.

I've uploaded 12-18 months of our monthly metrics (sanitized).

For each metric:
1. Direction and magnitude of trend, and whether it's distinguishable
from normal month-to-month variation (show the variation)
2. Any seasonality or calendar effects I should expect, so we stop
reacting to them
3. Inflection points — months where behavior changed — listed with
dates so I can match them to events
4. The three trends that most deserve leadership attention: two
improving, one deteriorating, with one-line explanations
5. A small-multiples chart set; build an interactive artifact if the
data supports it.

Never extrapolate more than one quarter ahead, and label any forecast
as a forecast.
```

**TREND HYGIENE**

- Annotate the series with events (tool rollout, policy change, reorg, M&A) — Module 06 depends on it.
- Keep definitions frozen across the window; if a definition changed, mark the break in the chart rather than splicing the lines.
- Three points is a line, not a trend. Wait for six or be honest about uncertainty.

06

ATTRIBUTION

# Connecting Changes to Outcomes

*The money question: did the thing we changed actually change the number?*

This is where metrics earn budget. You enforced MFA, shortened the patch SLA, rolled out a new EDR, ran a training campaign — leadership wants to know what it bought. The honest method: mark the change date, compare the metric's behavior before and after, and rule out the boring explanations before claiming victory.

## When to use it

- Policy changes — new password/MFA policy vs. account-compromise and helpdesk-reset metrics
- Process changes — patch SLA shortened from 30 to 14 days vs. actual remediation latency distribution
- Tooling changes — EDR replacement vs. detection rate, MTTD, and false-positive load
- Awareness campaigns — targeted training vs. department-level click and report rates
- Making the case — "this control paid for itself" with before/after evidence, caveats included

COPY-PASTE PROMPT · BEFORE/AFTER ANALYSIS

```
You are a security analyst who evaluates whether interventions actually
worked, with the skepticism of a good scientist.

I've uploaded monthly metrics (sanitized). On [date] we [describe the
change — e.g. "enforced MFA for all remote access" / "cut the critical
patch SLA from 30 to 14 days"].

Analyse:
1. The relevant metrics before vs. after the change — levels, trends,
and variability, with enough pre-period to establish a baseline
2. Alternative explanations: seasonality, other changes in the same
window (I'll list known ones: [...]), measurement changes, or
regression to the mean
3. Your verdict: strong evidence / weak evidence / no evidence the
change moved the metric — and what data would strengthen it
4. A single annotated chart showing the change date and the effect

Write the verdict in plain language I can put in front of a CFO,
including the caveats.
```

### ATTRIBUTION HONESTY RULES

- Correlation with a rollout date is a hypothesis, not proof — say "consistent with," not "caused by," unless you've ruled out alternatives.
- Beware regression to the mean: changes launched right after a terrible month look effective by default.
- One change at a time is rare in real life — when changes overlap, report the bundle's effect and say you can't split it.
- Negative results are results: "the campaign didn't move click rates; here's what we'll try instead" builds more credibility than spin.

07

NARRATIVE

# Telling the Story

*Leadership doesn't remember numbers. They remember the sentence about the numbers.*

A metrics story has the same skeleton every time: **situation** → **movement** → **cause** → **consequence** → **ask**. "Phishing reports doubled (movement) since the new report button shipped (cause), which caught two real campaigns early (consequence); we want to extend it to mobile (ask)." Claude drafts this directly from the analysis in Modules 02–06 — then renders it as a deck or doc with the Skills from this same conversation.

## When to use it

- Quarterly board and risk-committee reporting — five charts, five sentences, one ask
- Budget cases — tie the metric movement to the control investment, caveats and all
- Monthly ops reviews — same data, different altitude: actions and owners instead of asks
- Translating for audiences — the same quarter told for the board, the CIO, and the engineering all-hands

**COPY-PASTE PROMPT · METRICS-TO-NARRATIVE DECK**

```
You are an executive communications specialist who turns security
metrics into stories leadership repeats correctly when you're not in
the room.

Using the metric computations, trend read, and before/after analysis
from this conversation, create a 9-slide PowerPoint for the quarterly
risk committee.

Structure: 1 situation slide, 3 "what moved and why" slides (one chart
each, title states the takeaway), 1 "what bad would look like / what
we're watching" slide, 1 wins-tied-to-changes slide, 1 risks-and-asks
slide, 1 appendix of definitions.

Use the pptx skill. Black and white palette with orange accents, one
idea per slide. Every chart title is a sentence with a verb — not
"Phishing Metrics" but "Reports doubled after the new button shipped".
```

**STORY RULES**

- One chart, one message. If a chart needs a paragraph of explanation, it's two charts.
- Lead with change, not state: "down 40% since March" beats "currently 12 days."
- Always include one thing that got worse and what you're doing about it — all-green packs destroy credibility.
- End with a decision: approve, fund, accept, or watch. A story without an ask is a status update.

08

NO-CODE

# Living Dashboards with Artifacts

*An interactive view leadership can poke at — built in one conversation.*

Claude Artifacts builds functional, styled dashboards directly in the chat — live preview, iterate in plain English, export the HTML. Perfect for the metrics layer between “spreadsheet nobody opens” and “BI project nobody finishes.”

## When to use it

- Quarterly metrics dashboard — trend sparklines, RAG status against targets, drill-down by domain
- SLA aging views — vulnerability remediation by age bucket and severity, with the tail visible
- Before/after explorers — a slider on the change date so leadership can see the effect themselves
- Metric catalog browser — definitions, owners, formulas, so “what does this number mean” has one answer

**COPY-PASTE PROMPT · METRICS DASHBOARD**

```
You are an expert dashboard designer who builds clean, fast,
self-explanatory metric views for security leadership.

Using the monthly metrics data from this conversation [or: I'm pasting
the metrics table], build an interactive dashboard as an artifact.

Include: (1) a headline row of 5 KPIs with delta vs. last quarter and
RAG status against the targets I provide; (2) a 12-month trend chart
per domain with event annotations; (3) a vulnerability aging view that
shows the tail, not just the average; (4) a toggle between "board view"
(5 numbers) and "ops view" (everything).

Black and white theme with orange accents. Embed the data in the
artifact — no external calls, no real hostnames or usernames anywhere.
```

### DASHBOARD NOTES

- Embed sanitized, aggregated data only — an artifact is a view, not a data store.
- Annotate change dates on every trend chart; an unannotated trend invites wrong stories.
- If a widget doesn’t answer one of the five questions from the intro, delete it.

09

AUTOMATION

# Automating the Monthly Grind

*Beyond chat: Co-Work and Claude Code assemble the metrics pack while you review it.*

The monthly metrics cycle is the same every month: gather exports, clean them, compute, compare, chart, write. Co-Work (Claude desktop app) and Claude Code (terminal) are agents that take scoped folder access and run that pipeline — with you reviewing the output instead of doing the plumbing.

## Metrics-flavored use cases

- Month-end assembly — point it at the exports folder; it cleans, computes per your catalog definitions, and produces the metrics workbook
- History building — batch-process a year of old exports into one normalized time-series file
- Consistency checks — diff this month's export schemas against last month's and flag silent changes that would corrupt trends
- Report production — regenerate the standard .docx pack and the deck skeleton, pre-filled with the new numbers and deltas

COPY-PASTE PROMPT · MONTH-END METRICS PIPELINE

```
You are a meticulous metrics engineer who turns a folder of raw exports
into a clean, reproducible monthly metrics pack.

Take access to ~/Metrics/2026-05 (sanitized exports only). Using the
metric definitions in catalog.md in that folder:

1. Validate each export: expected columns present, row counts sane,
schema unchanged from last month (compare to 2026-04) — STOP and
report if a schema changed rather than guessing
2. Compute every catalog metric; write metrics_2026-05.xlsx with values,
prior month, delta, and target status
3. Append this month to timeseries.csv without altering history
4. Generate draft_narrative.md: three biggest movements with the chart
for each
5. Produce a run log of every transformation so the numbers are
auditable.

Ask before overwriting anything. Do not modify the raw exports.
```

### PIPELINE SAFETY NOTES

- Agents read sanitized exports in a scoped folder — never production systems or raw logs with identifiers.
- Raw exports are immutable; the pipeline writes new files. History files are append-only.
- Schema changes halt the run for a human decision — silent adaptation is how trends get quietly broken.
- You review the run log and sign off on the numbers. The agent does the plumbing; you own the figures.

10

THE SHORTCUT LAYER

# /god · /ghost · /artifacts · /blindspot · /deepdive

*Five reusable prompt shortcuts to standardize across the team.*

**Straight talk first:** apart from Artifacts (a real Claude feature), these are **not** official built-in commands — you may see them hyped online as “secret modes.” They work because Claude reads them as shorthand for an instruction. That’s actually better for you: you get to define exactly what each one means, agree on it across your team, and get consistent behavior. Paste the definitions below into your user preferences or a Project’s instructions (or define them as real custom slash commands in Claude Code), and the shortcuts become reliable.

| `/god` | **Expert mode — peer-level depth, zero hand-holding** |
|---|---|

“When I type /god: respond as a principal security metrics engineer talking to a peer. Assume fluency with statistics, time-series analysis, and security operations. Skip basics, give the strongest technical answer, state assumptions and confidence intervals, and say ‘the data can’t answer that’ rather than hedge.” Use for: metric design reviews, statistical significance questions, sampling and denominator debates.

| `/ghost` | **Ghostwriter — sounds like me, ready to send** |
|---|---|

“When I type /ghost: write in my voice — plain, direct, no AI-isms. Match the audience I name (board, CFO, ops team, auditor). Output only the final text, ready to paste, no commentary.” Use for: the narrative paragraph under each chart, the budget-ask email, the ‘why this metric got worse’ explanation that needs to sound honest, not defensive.

| `/artifacts` | **Build it — real, usable output** |
|---|---|

“When I type /artifacts: don’t describe it, build it. Produce the deliverable as an artifact or file — dashboard, chart set, workbook, deck — and keep iterating on the same artifact as I give feedback.” Use for: dashboards (Module 08), metric workbooks (Module 02), narrative decks (Module 07). This one maps to a genuine product feature.

| `/blindspot` | **Red-team my numbers** |
|---|---|

“When I type /blindspot: stop agreeing with me. Attack the metrics or analysis above: which number could improve while risk worsens, where is the denominator dishonest, what would a skeptical CFO or auditor ask, which trend is actually noise? Rank the top 5 blind spots by likelihood × impact and suggest one fix each.” Use before: every quarterly pack goes out the door. The most metrics-native shortcut of the five — run it on Module 04’s output especially.

| `/deepdive` | **Exhaustive layer-by-layer analysis** |
|---|---|

“When I type /deepdive: go exhaustive. Work the question layer by layer — data lineage, definition history, statistical behavior, alternative explanations, open questions — and cite sources where facts are involved (use web search or Research mode if needed). Length is no constraint; completeness is.” Use for: ‘why did this metric really move,’ root-causing a broken trend line, benchmarking deep-dives.

11

INTEGRITY

# Metric Integrity & Safe AI Use

*A metrics program is only as good as its worst-handled number.*

## Data handling

- Sanitize exports before upload: strip usernames, emails, hostnames, and IP addresses you consider sensitive — metrics are about patterns, not persons. Follow your organization's AI acceptable-use policy.
- Prefer org-managed accounts (Team/Enterprise) over personal accounts for work content.
- Agents and connectors get least privilege: scoped folders of sanitized exports, never production systems.

## Number integrity

- Every figure traces to a computation Claude actually performed on real data — ask for row counts, exclusions, and formulas ("show your work") and spot-check against the source.
- Freeze definitions in a shared catalog; when a definition changes, version it and mark the break in every trend chart.
- If Claude says a metric can't be computed from the data provided, that's the right answer — never accept an estimate dressed as a measurement.
- Causation claims from Module 06 are hypotheses until you've checked them against operational reality. Label confidence honestly in everything that leaves the team.

## Prompt injection awareness

When Claude reads external content — exports, web benchmarks, documents — that content can contain text designed to manipulate AI tools. Claude treats observed content as data, not instructions, and will surface suspicious embedded instructions rather than follow them. Your part: notice when something odd appears in a workflow, and report it like any social-engineering attempt — because that's what it is.

**A TEAM OPERATING AGREEMENT WORTH ADOPTING**

- One shared Project containing the metrics catalog (definitions + formulas), the shortcut definitions from Module 10, and the sanitization checklist.
- Every metric has a named owner who can explain its formula, its target, and its last three movements.
- Run /blindspot on every pack before it goes to leadership. Nothing all-green ships without a 'what we're watching' section.
- Monthly 30-minute share-out: one metric improved, one metric retired, one definition tightened.

APPENDIX A · STARTER CATALOG AND TOOLS

# Steal this starter catalog

Twelve metrics that cover the core of a program. Adapt formulas and targets to your environment — then keep them frozen.

| METRIC | TYPE | WATCH FOR |
|---|---|---|
| Critical/KEV vulns past SLA (count + oldest age) | **KRI · lagging** | The tail age, not the average |
| Median + 95th percentile time-to-remediate criticals | **KPI · lagging** | P95 diverging from the median |
| % known assets in scan scope (coverage) | **KRI · leading** | Denominator drift |
| MTTD / MTTR by severity | **KPI · lagging** | Severity reclassification gaming |
| Phishing report rate vs. click rate | **KPI · leading** | Easier simulations inflating success |
| Repeat-clicker population | **KRI · leading** | Same names every quarter |
| MFA coverage of remote + privileged access | **KRI · leading** | Exceptions that never expire |
| Dormant accounts older than 30 days | **KRI · leading** | Steady small growth |
| Privileged account count vs. headcount | **KRI · leading** | Admins growing faster than staff |
| EDR + log onboarding coverage of verified inventory | **KRI · leading** | “100%” of an untrusted inventory |
| Alert true-positive rate | **KPI · lagging** | Volume celebrated without quality |
| Critical vendors past reassessment date | **KRI · lagging** | Assess-once-and-forget |

## Tools and references

| TOOL | WHAT FOR | WHERE |
|---|---|---|
| **Claude** | Chat, Research mode, Artifacts, Skills, Connectors | claude.ai |
| **Claude Desktop / Code** | Co-Work and terminal agents for the pipeline (Module 09) | claude.ai/download · claude.ai/code |
| **CIS Controls / NIST CSF** | Control framing for what to measure | cisecurity.org · nist.gov |
| **Verizon DBIR** | Annual breach data for context and benchmarks | verizon.com/dbir |
| **CISA KEV** | Known exploited vulns — the SLA list that matters most | cisa.gov/kev |

*“A security program that can’t measure itself is asking leadership for faith. One that measures honestly is asking for a decision. Pick one module. Run it on this month’s numbers.”*