TALON | STUDIO

WHITEPAPER

Detection Engineering, as Code.

How a Git-backed, vendor-neutral detection platform turns SIEM content into tested, versioned, portable engineering artifacts — and why that same substrate is the safest foundation for agentic AI in the SOC.

TALON LABS · DETECTION PLATFORM RESEARCH

JULY 2026 · V1.1

ABOUT THIS PAPER

This whitepaper introduces **Talon Studio**, the self-hosted, vendor-neutral detection engineering platform from Talon Labs. It is written for detection engineers, SOC leaders, and security architects evaluating how to industrialize detection content — and how to prepare that content for the arrival of agentic AI.

Part I surveys the state of the discipline, drawing on published industry research. Part II describes the Talon Studio platform: its modules, architecture, and detection lifecycle. Part III examines agentic AI in security operations and the governance model Talon Studio applies to it.

WHO THIS IS FOR

Detection engineers and SOC leads industrializing a rule estate; security architects evaluating detection-as-code platforms; and CISOs of self-hosting or regulated organizations weighing how — and whether — to let AI agents near production detections.

ABOUT THE DATA

Industry figures come from published research — CardinalOps (2024/25), Picus (2025), SANS / Anvilogic (2025 n=264, 2026 n=307), SANS SOC surveys, IBM / Ponemon (2025), Mandiant M-Trends, Gartner — each cited inline and resolved in the References. Vendor-sponsored and vendor-reported figures are labeled as such. Product facts reflect the Talon Studio codebase and documentation as of July 2026; roadmap items are noted as such and subject to change.

CONTENTS

EXECUTIVE SUMMARY

The bottleneck is not telemetry. It is engineering.

Enterprise SOCs already collect enough data to detect more than 90% of adversary techniques — yet their SIEMs actually detect about 21%, and roughly one in eight production rules is silently broken.[1] Talon Studio exists to close that gap: it turns detection content into engineered software — versioned, tested, portable, and governed — and it does so on infrastructure you own.

21%
of MITRE ATT&CK techniques covered by production SIEM detections
CardinalOps, 2025[1]

14%
of simulated attacks raise an alert — even though 54% are logged
Picus Blue Report, 2025[3]

86%
of security pros say a new detection takes a week or more, end to end
ESG / Anvilogic, 2022[5]

$4.44M
global average cost of a data breach; $10.22M in the United States
IBM, 2025[6]

What Talon Studio is

Talon Studio is a **self-hosted, vendor-neutral detection engineering platform** from Talon Labs, delivered as a single-container appliance. It connects to your SIEM with metadata-only introspection, maps MITRE ATT&CK coverage from evidence, and manages the detection lifecycle: authoring in a neutral Sigma-first format, on-demand validation against Atomic Red Team-derived telemetry, change requests whose merges belong to humans, and compilation to Splunk, Microsoft Sentinel, Elastic, and Google Chronicle.

What makes it different

Every detection is a YAML file in a Git repository inside the appliance — not a row in a vendor's SaaS. Your rules, your history, your infrastructure. The platform's AI layer, Talon Claw, operates under a hard invariant enforced in the service layer: **agents can propose; only humans can merge.**

Why now

Two forces are converging. Detection engineering has matured from ad-hoc SIEM administration into a software discipline — 62% of teams now version-control their rules, yet only 42% have CI/CD for them.[4] Meanwhile, agentic AI is entering the SOC faster than governance can keep up: 83% of practitioners already use AI tools, but only 42% trust them for core work like tuning detections.[4]

This paper argues the two trends resolve each other: **Git-backed, validation-gated detection-as-code is the substrate that makes AI agents safe and useful** — every AI contribution becomes a reviewable diff behind a test suite and a human approval, instead of an unaudited change to production.

IN THIS PAPER

Part I — the state of the discipline, in published numbers. **Part II** — the Talon Studio platform: surfaces, architecture, lifecycle. **Part III** — agentic AI in the SOC, and the governance model that makes it deployable.

THE TALON LOOP · THE GOVERNED CYCLE THIS PAPER RETURNS TO

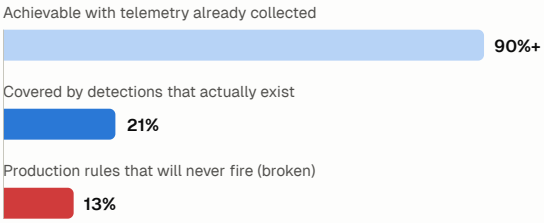
SIEM CONNECTED → EVIDENCE MAPPED → DETECTIONS AUTHORED → SAVED AS CODE → VALIDATED & DEPLOYED → DRIFT & GAPS VISIBLE

The coverage illusion

Security teams believe their SIEM has them covered. The measured reality, across hundreds of production deployments and millions of log sources, is a wide and persistent gap between what organizations *could* detect and what their rules *actually* detect.

Detection coverage vs. telemetry potential

Share of MITRE ATT&CK techniques, production SIEMs

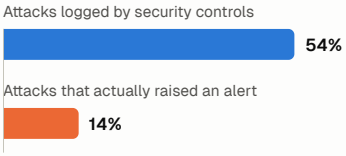


Source: CardinalOps, State of SIEM Detection Risk 2025 — a five-year cumulative sample of 13,000+ rules and ~2.5M log sources across Splunk, Microsoft Sentinel, CrowdStrike LogScale, and Google SecOps.[1]

FIG. 01 The gap is engineering, not data: organizations already ingest what full coverage requires, but the rules were never built — or quietly broke.

The validation gap

Outcomes of 160M+ real-world attack simulations, 2025



Source: Picus Security, Blue Report 2025. Vendor research; corroborated by CardinalOps' separately derived broken-rule findings.[3]

FIG. 02 Evidence reaches the SIEM and still no alarm sounds — broken rule logic, missing fields, and untested content absorb the signal.

Rules break silently

CardinalOps puts broken rules at 13% — the average across five years of its data — rules that will never fire because of misconfigured data sources, missing fields, or stale logic: a silent eighth of the average rule base.[1,2] Nothing alerts when a detection dies; the failure mode of a broken rule is indistinguishable from a quiet network.

The human cost compounds it

The average SOC fields roughly 4,484 alerts a day and its analysts are unable to deal with about two-thirds of them.[7] 73% of organizations name false positives their top detection challenge[8] — and in a separate survey, 66% of false positives were reported to originate in vendor-provided rules.[4] Downstream, 71% of SOC analysts report burnout.[10]

And the adversary is not waiting

Global median dwell time rose two consecutive years — 10 to 11 to 14 days — the first back-to-back deterioration in the metric's history.[11] Meanwhile 86% of security professionals say a new detection takes a week or more end to end.[5] When shipping one rule takes longer than the adversary's entire campaign, the workflow is the vulnerability.

> **The bottleneck is detection engineering workflow, not data collection.**
>
> EDITORIAL SYNTHESIS OF CARDINALOPS, PICUS & SANS FINDINGS, 2024-2026

WHY THIS PERSISTS

- Rules live inside SIEM consoles — unversioned, untested, unreviewed.
- Knowledge is tribal: the “why” of a rule leaves when its author does.
- 43% of enterprises run two or more SIEMs; content forks and drifts.[2]
- Migration is punitive: an illustrative industry scenario — 2,000 inherited rules, an estimated six months to port by hand.[16]

From craft to code

The discipline's response to the coverage illusion has been to treat detections the way engineers treat software: written in open formats, version-controlled, peer-reviewed, tested before deployment, and continuously validated after it. The industry calls this **detection-as-code**.

FIG. 03 A decade of institutionalization: from maturity models and documentation frameworks to open rule formats, analyst-endorsed lifecycles, and a recurring industry survey.[12,13,14,15]

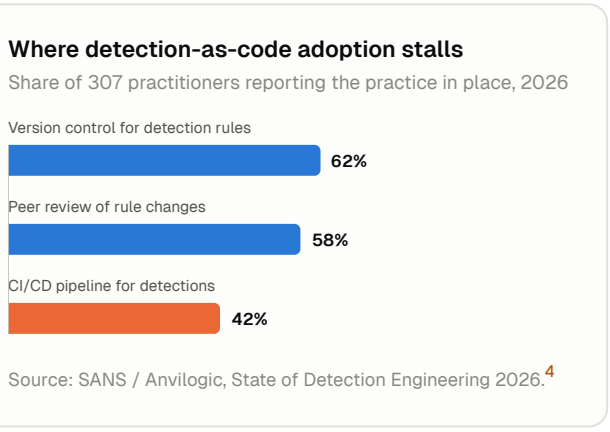


FIG. 04 Teams adopt version control, then stall 20 points short of CI/CD — the step that requires platform engineering most teams can't spare.

The barrier is not conviction but capacity: 72% cite time and resource constraints, 61% a lack of in-house skills, and only 13% report high software-engineering proficiency on the detection team.[4] The practice is proven; the tooling burden is what a platform must absorb.

What the practice delivers

- **Versioning & auditability.** Every rule change carries an author, a diff, a reason, and a rollback path — the property auditors and post-incident reviews actually ask for.
- **Testing before deployment.** Rules are exercised against known-bad telemetry (Atomic Red Team is the de facto free standard, 1,700+ tests mapped to ATT&CK) instead of being trusted on arrival.
- **Peer review.** The Palantir ADS framework (2017) popularized documented, peer-reviewed detections; a diff in Git is the natural unit of that review.[13]
- **Portability.** Open formats — Sigma above all, with 3,000+ community rules and 15,000+ contributions since 2017 — decouple detection logic from any one vendor's query language.[15]

> **"A more systematic, flexible and comprehensive approach to threat detection… somewhat inspired by software development."**
>
> ANTON CHUVAKIN, "CAN WE HAVE DETECTION AS CODE?", 2020

Gartner's consolidation of breach-and-attack simulation into "Adversarial Exposure Validation," with structured validation forecast to reach 60% of organizations by 2029, points the same direction: *continuous, evidence-based proof that detections work* is becoming table stakes.[17]

WHERE IS YOUR PROGRAM? · A FOUR-STEP LADDER

1 · Ad-hoc
Rules live in consoles; knowledge is tribal; breakage is silent.

2 · Versioned
Rules in Git with peer review — where 62% of teams now are.[4]

3 · Tested
Validation gates every deploy — where adoption stalls at 42%.[4]

4 · Governed & agentic
Evidence-gated content; AI proposes volume, humans keep the merge.

PART II

The Talon Studio platform

A self-hosted, vendor-neutral detection engineering appliance: twelve integrated surfaces over one shared spine — detection content versioned in Git, operational evidence in Postgres — delivered as a single container you run on your own infrastructure.

12
integrated product surfaces on one data spine

4
SIEM targets compiled from one canonical rule

1
container — app, database, translator, Git store

0
raw log streams ingested — metadata, evidence & artifacts only

One platform, twelve surfaces, a single system of record

Talon Studio organizes the detection lifecycle into named product surfaces that share one workspace spine: detection content and configuration versioned in Git, operational evidence in Postgres. Coverage analysis, authoring, validation, and operations all work from that shared state.

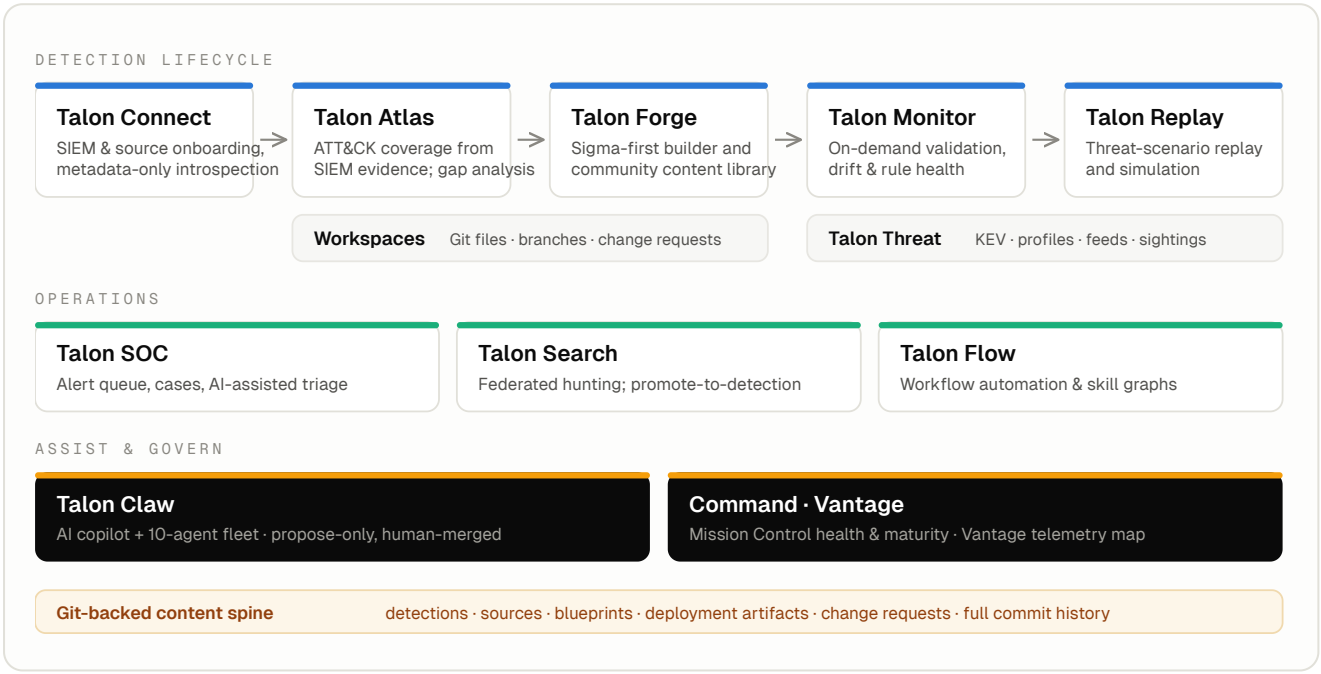


FIG. 05 The Talon Studio surface map. Lifecycle surfaces build and prove detections; operations surfaces consume them; Claw and Command assist and govern. All twelve share one Git-backed workspace spine — content versioned in Git, operational evidence in Postgres.

Metadata in, artifacts out

Talon Studio deliberately does not ingest or store raw logs. Talon Connect introspects the SIEM for indexes, sourcetypes, fields, and data models — enough to map coverage and generate correct queries — while events stay where they already live. The platform's outputs are engineering artifacts: versioned YAML, compiled queries, and deployment packages.

Workspaces as the cockpit

Each workspace is a Git repository with branches, change requests, releases, and rollback history. Detections, data-source declarations, and Forge blueprints are files. AI-proposed changes always arrive as reviewable change requests, and workspaces can protect main so human changes go through the same gate.

SECTION 02 · SURFACE CATALOG

What each surface does

Twelve surfaces cover the full arc from raw telemetry awareness to governed AI assistance. Each is summarized below with its place in the lifecycle.

SURFACE	ROUTE	PURPOSE
Command	`/command`	Mission Control: portfolio health, workspaces at risk, validation drift, release state, and a 0–100 Detection Maturity Score — plus Vantage, a visual map of the telemetry evidence Talon reads from your SIEM.
Talon Connect	`/connect`	Onboards SIEMs, sources, XDR (CrowdStrike Falcon, Microsoft Defender), and Cribl. Introspects indexes, sourcetypes, fields, and CIM data models — metadata only, never raw events.
Workspaces	`/workspaces`	The engineering cockpit: Git-backed files, branches, change requests with approval gates, releases, deployments, and rollback history.
Talon Atlas	`/atlas`	Coverage analysis: maps MITRE ATT&CK coverage from SIEM evidence, gates it by threat relevance (CISA KEV, threat profiles), and surfaces detection opportunities for the gaps.
Talon Forge	`/forge`	Detection builder and content library. Sigma-first code editor, guided form view, native SIEM-dialect tabs, and blueprints — seeded by a continuously synced library of community content.
Talon Monitor	`/validate`	On-demand validation against Atomic Red Team ground truth, drift detection when a passing rule regresses, per-rule health, and the Evidence Engine's gap queue — every gap tracked to a human- or AI-attributed close.
Talon Replay	`/replay`	Threat-scenario replay and simulation: exercises a detection against scenario telemetry in-platform, without touching the connected SIEM.
Talon Threat	`/threat-intel`	Threat intelligence workbench: CISA KEV tracking, threat profiles, feed management, IOC sightings with geo context, and readiness views that drive Atlas prioritization.
Talon SOC	`/soc`	The operational consumption layer: alert queue, case management, dispositions, and AI-assisted triage fed by SIEM notables and XDR detections.
Talon Search	`/search`	Investigation search across your SIEM and Cribl Lake — multi-source federation on Enterprise — with a promote-to-detection path that turns a good hunt into a governed rule.
Talon Flow	`/flow`	Workflow automation: skill graphs with webhook, schedule, and event triggers that stitch platform actions into repeatable operational playbooks.
Talon Claw	`/claw`	The AI layer: chat copilot, a ten-agent fleet, retrieval over a curated 234-skill security library and org content, agent memory, and run governance — all propose-only.

A deliberately opinionated shape

The surfaces are not a suite of loosely-related tools. They encode an opinion: coverage should be measured from evidence, content should be authored once in a neutral format, nothing should ship without recorded evidence, and AI should propose — never merge. The catalog above is the lifecycle made navigable.

386 API routes

121 database tables

~322 automated test files

One container, one volume, everything you own

The default deployment is deliberately simple: a single Docker container and one mounted `/data` volume. Four co-located components — the Next.js application, an embedded PostgreSQL, a pySigma translation sidecar, and a local Git store — keep the install trivial and the blast radius contained.

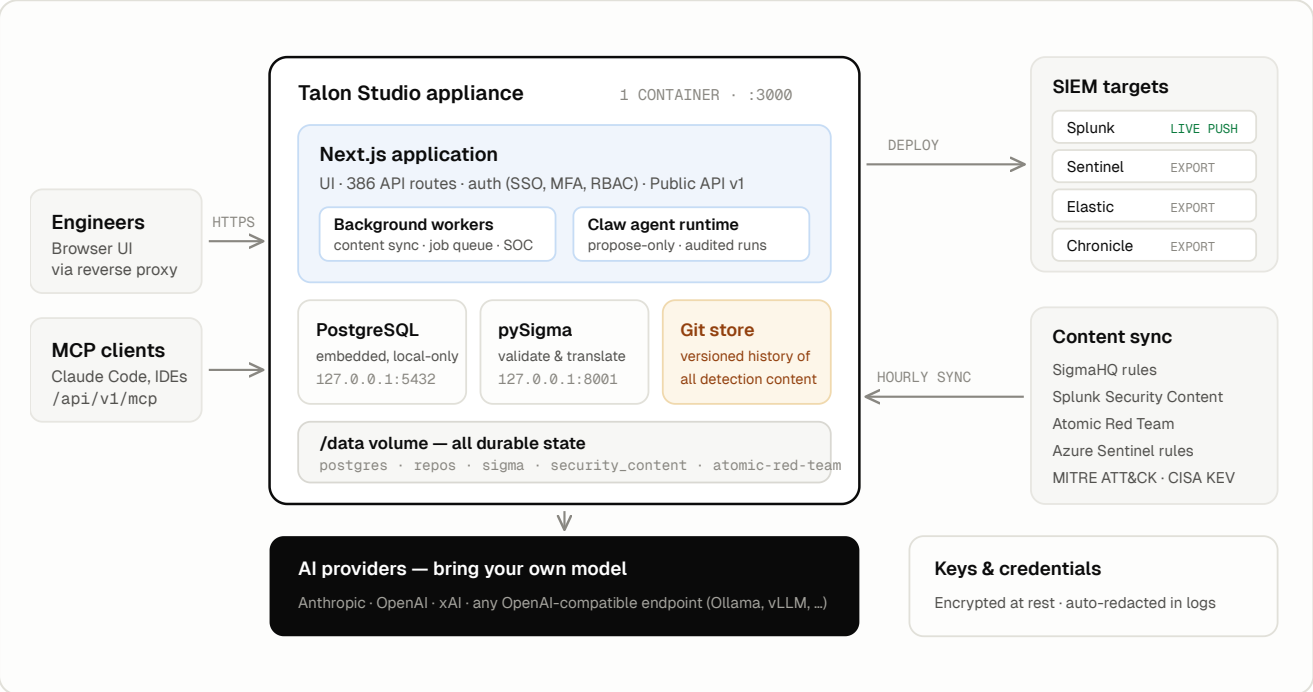


FIG. 06 The appliance architecture. Postgres and pySigma bind to localhost only; TLS terminates at the operator's reverse proxy; SIEM credentials and AI keys are encrypted at rest. Everything durable lives on one `/data` volume — backup and restore is a volume copy.

The runtime, component by component

COMPONENT	ROLE
Next.js application	UI, 386 API routes, authentication and RBAC, Public API v1, and the background loops for content sync, the durable job queue, SOC pull, and agent runs.
Embedded PostgreSQL	Operational state and the indexed content cache: 121 tables covering SIEM evidence, validation runs, coverage topology, threat intel, cases, and agent telemetry.
Embedded pySigma	Sigma validation, SIEM query translation, and rule-versus-events testing for the validation engine — four backend targets compiled from one canonical rule.
Local Git store	Versioned history for detection content: detections, data-source declarations, blueprints, and deployment artifacts as files, with change requests and full commit history.

From telemetry to deployed, proven detection

Every detection follows the same governed cycle — the **Talon Loop**. Detections ship with validation evidence attached, a release gate blocks anything that cannot run in the connected SIEM, and nothing merges to the protected main branch without a human approval — whether the author was an engineer or an agent.

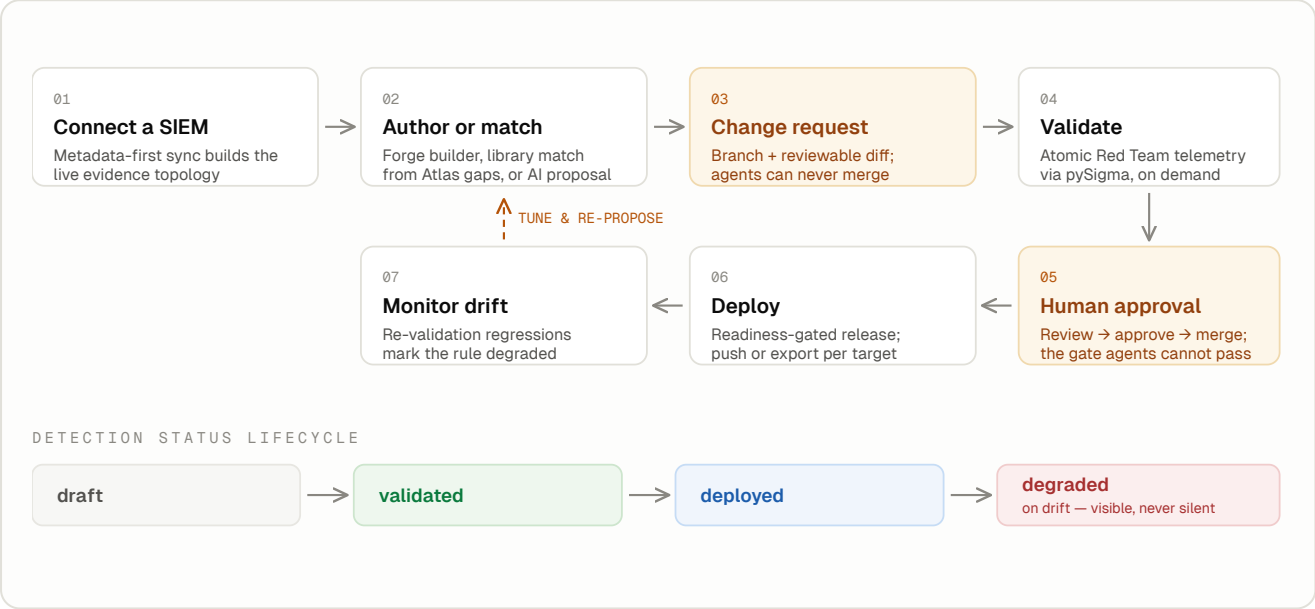


FIG. 07 The Talon Loop and the status lifecycle beneath it. Validation runs are recorded as evidence; a rule that regresses from pass to fail is marked degraded rather than failing silently.

The evidence engine

A live topology of the connected SIEM — indexes, sourcetypes, CIM models, deployed content — is distilled into canonical gap records: blind spots, missing fields, broken dependencies, failed deployments. Each carries a severity and a confidence — confirmed, probable, or inferred — and the same records surface in Monitor, Command, Atlas, and Threat. What it cannot prove, it does not claim.

Deploys are earned

Release gates block a detection that fails environment readiness — one the connected SIEM cannot actually run. Behavioral validation travels with every release as recorded evidence, and a strict org policy can escalate a failing validation from warning to hard block.

History is retrievable

Change-request work commits automatically; saves on main stage as pending changes for a human to commit. Validation runs and approvals are recorded alongside as structured records. Any detection can be diffed against — and restored to — any prior version.

THE AUDIT TRAIL · ILLUSTRATIVE · COMMIT FORMATS AS THE SYSTEM WRITES THEM

```
a41f2c9  detection: LSASS Memory Access via Rundll32
9d0e77b  change: Tune LSASS false positives Talon/detections/lsass-rundll32.yml
5b8c3aa  merge: Tune LSASS false positives
         # validation runs & approvals recorded as structured records beside the history
```

Write once, compile anywhere

Talon Studio treats the detection itself — not any SIEM's dialect — as the canonical artifact. A neutral YAML contract with Sigma as the canonical query is compiled through an embedded pySigma engine into each target's native language.

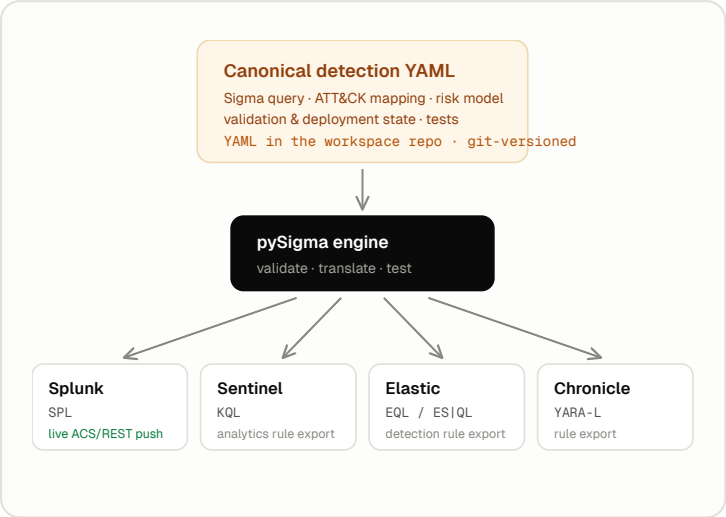


FIG. 08 One canonical rule, four native outputs. Per-vendor escape hatches (`x_splunk`, `x_sentinel`, `x_elastic`) preserve platform-specific tuning without forking the rule.

DETECTIONS/LSASS-RUNDLL32.YML · ILLUSTRATIVE, ABRIDGED

```
name: LSASS Memory Access via Rundll32
detection_type: TTP   severity: high
mitre_technique: T1003.001   csf: [DE.CM-01]
data_sources: [Sysmon Process Access]
required_fields: [TargetImage, GrantedAccess]
risk: { score: 72, entities: [attacker, victim] }
queries:
  sigma: … # canonical
  splunk: … # compiled · sentinel · elastic · yara_l
validation:
  atomic_test_id: T1003.001   last_result: pass
false_positives: [signed admin tooling]
```

The YAML contract

Each detection file carries everything an auditor or a migration would need: identity and provenance, MITRE ATT&CK tactic and technique, data-source requirements, a neutral risk model with attacker/victim entities, NIST CSF 2.0 codes, false-positive notes, validation state, and deployment history.

Why it matters

- **Migration insurance.** Content survives a SIEM change — recompile instead of rewrite.
- **Multi-SIEM reality.** Teams running Splunk beside Sentinel keep one logical rule set.
- **Reviewable diffs.** A YAML file in Git is something a peer — or a CI gate — can actually review.
- **Community leverage.** A continuously synced library (SigmaHQ, Splunk Security Content, Azure Sentinel rules, Atomic Red Team) seeds authoring instead of starting from a blank page.

CONTENT LIBRARY SYNC · HOURLY BY DEFAULT, CONFIGURABLE

SigmaHQ · Splunk Security Content · Atomic Red Team · Azure Sentinel · Stratus Red Team · MITRE ATT&CK · CISA KEV · NIST 800-53 mappings

PART III

The agentic frontier

AI agents are arriving in the SOC faster than governance can keep up. The teams that will benefit are the ones whose detection content is already code: versioned, tested, and gated by human review. That is not a coincidence — it is the design.

88%
piloting or using (42%) — or planning (46%) — AI assistants for detection & response
Gartner, 2025[20]

>40%
of agentic AI projects predicted to be canceled by end of 2027
Gartner, 2025[19]

40%
of Block's new detections in 2025 were AI-assisted — human review still required
Detection at Scale, 2026[24]

29 min
average eCrime breakout time — the clock defenders now race
CrowdStrike, 2026[30]

Every vendor has an agent. Few have a governance story.

In under two years the market moved from chat copilots to autonomous agents. Microsoft shipped eleven Security Copilot agents and an MCP server over Sentinel; Google previewed a Detection Engineering Agent that writes and tests rules; CrowdStrike and Palo Alto field prebuilt agent fleets, and SentinelOne ships agentic triage and investigation in Purple AI; AI-SOC startups raised well over $200M in 2024–25 funding rounds.[32]

The evidence for real gains

The strongest results are controlled studies, not marketing: in Microsoft's randomized trials, experienced analysts were 22% faster and 7% more accurate with an AI copilot, and agent-augmented phishing triage delivered up to 6.5× as many true positives per analyst-minute.[21] IBM measured organizations with extensive security AI and automation saving $1.9M per breach and shortening breach lifecycles by 80 days.[6]

And the reasons for caution

Gartner charts AI SOC agents racing toward the peak of inflated expectations and predicts over 40% of agentic AI projects will be canceled by end of 2027 over cost, unclear value, or inadequate risk controls.[18,19] Practitioners agree: SOC teams rank AI/ML at the bottom of technology satisfaction,[9] and OWASP's 2025 guidance codified *excessive agency* — too much autonomy, permission, and function — as a first-class LLM risk.[28]

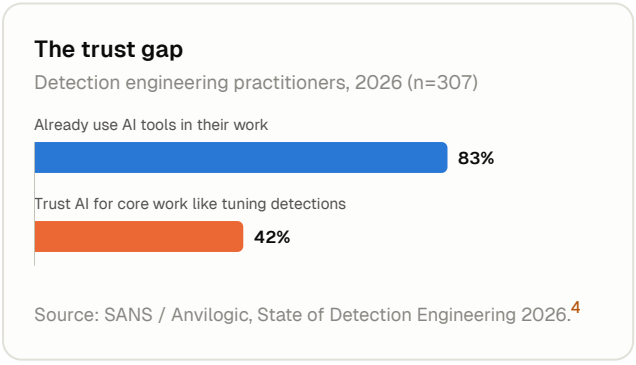


FIG. 09 Usage has outrun trust. Closing that gap is a governance problem, not a model problem.

The adversary is already agentic

The urgency is not hypothetical. Average eCrime breakout time fell to 29 minutes — 65% faster than a year earlier, with the fastest observed at 27 seconds — and AI-enabled adversary operations rose 89% year over year.[30] In November 2025, Anthropic disclosed the first reported largely AI-orchestrated espionage campaign, in which agents executed an estimated 80–90% of tactical operations.[30] Machine-speed attack timelines make machine-assisted defense a requirement — and make *ungoverned* machine-assisted defense a liability.

THE 2024-26 AGENTIC WAVE, ABBREVIATED

ACTOR	SIGNAL
Microsoft	11 Copilot agents; Sentinel repositioned as an agentic platform with an MCP server (2025)
Google	Detection Engineering Agent that drafts rules and validates them with synthetic events (2026, preview)[22]
CrowdStrike	Charlotte AI triage; seven-agent “agentic security workforce” (2025)[23]
SentinelOne	Purple AI “Athena”: agentic triage, hunting, rule creation (2025)
Startups	Dropzone $37M B; Prophet $30M A; Exaforce $75M A; 7AI $36M seed (2024–25)
Standards	OWASP agentic-AI threat guidance (2025); NSA security-design sheet for MCP (2026)[28,29]

Vendor performance figures in this market (98% triage accuracy, 30-minutes-to-60-seconds, 98% MTTR reduction) are vendor-reported and not independently audited; this paper cites controlled studies where they exist.

Detection-as-code is what makes agents safe

Across the publicly documented production deployments of AI in detection engineering — Block, Panther, Google, Elastic — the same pattern appears: **agents propose, automation validates, humans approve through code review.** The prerequisite is always the same — detection content that lives as code, not as rows in a console.

> **"AI promises to reduce operational overhead by automating routine rule creation while amplifying human nuance."**
>
> JACK NAGLIERI, "THE AI-POWERED DETECTION ENGINEER", 2025[25]

Block runs this in production: an internal agent plus an MCP server lets any engineer author detections in natural language. Roughly 40% of Block's new detections in 2025 were AI-assisted — and every one landed as code, committed under the operator's identity, behind a mandatory human pull-request review.[24] Panther's AI proposes improvements through existing Git workflows with unit tests attached; nothing deploys without approval.[26] Google's Detection Engineering Agent validates its own generated rules against synthetic events before a human ever sees them.[22] Anvilogic's practitioners are blunt: agentic detection engineering presupposes clean schemas, exemplary rule libraries, and a human gatekeeper.[27]

The inverse also holds. An agent pointed at a SIEM console inherits every weakness of console-managed content: no diff to review, no test to run, no history to roll back, and an audit trail that cannot distinguish the agent's hand from the human's — the exact failure modes OWASP (excessive agency) and ISACA (loss of traceability and accountability) warn about.[28]

PROPERTY	AGENT ON A SIEM CONSOLE	AGENT ON DETECTION-AS-CODE
Unit of change	Live rule edit	Reviewable diff on a branch
Verification	Hope	Validation evidence on the change request; readiness-gated release
Attribution	Shared service account	Commit + run log per agent action
Blast radius	Production, immediately	A branch, until a human merges
Rollback	Manual reconstruction	`git revert`
Learning loop	None	Review feedback becomes agent memory

FIG. 10 Why the substrate decides whether agentic AI is an asset or a liability.

THE CONVERGENT PATTERN

Independently arrived at by Block, Panther, Elastic, Google, and Anvilogic in 2025–26:

AGENTS PROPOSE → CI VALIDATES → HUMANS APPROVE

Talon Studio ships this pattern as a hard platform invariant rather than a convention — the subject of the next section.

Talon Claw: a governed agent fleet

Talon Claw brings a fleet of ten purpose-built agents and an in-context copilot to every surface of the platform — on models you choose, including fully self-hosted ones. Every agent operates propose-only: the merge button belongs to humans, and the service layer enforces it.

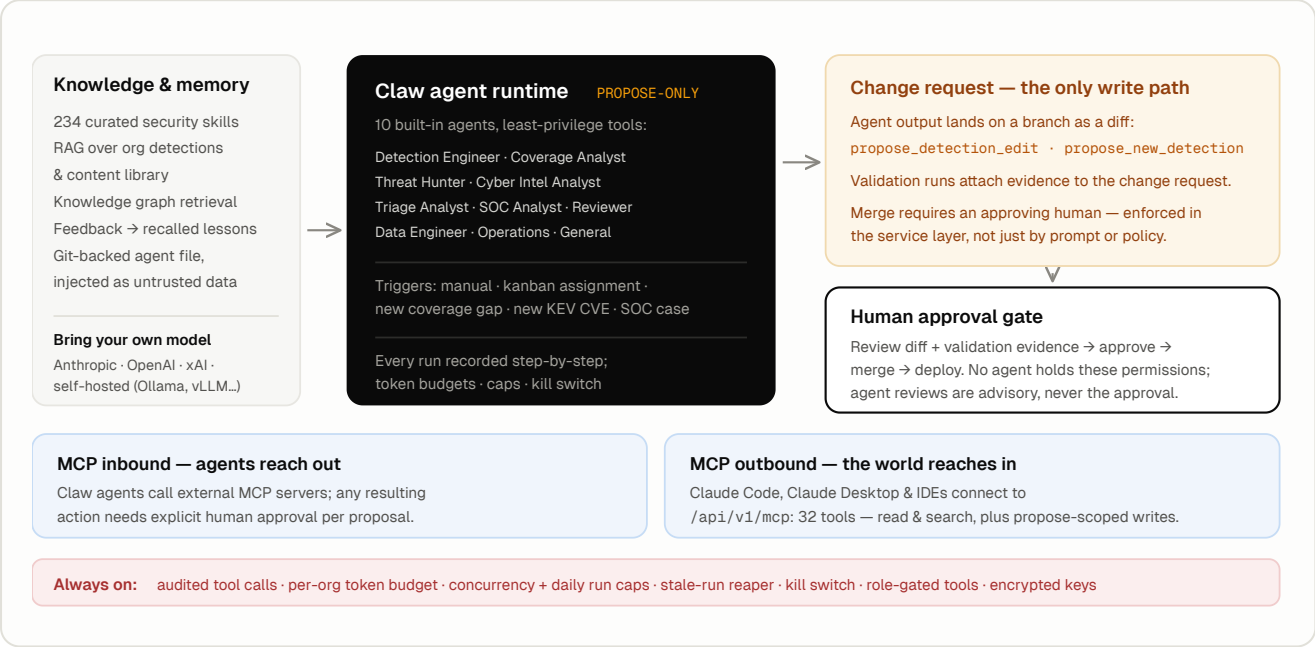


FIG. 11 The Claw agentic architecture. Agents draw on governed knowledge, act through least-privilege tools, and can only ever produce proposals; humans own approval, merge, and deploy. MCP connects the fleet to external tools in both directions under the same rules.

A copilot where the work is

"Ask AI" is embedded in the workflow, not a separate chat window: explain or improve a rule from the Forge detail view, diagnose a failed validation run or a drift alert in Monitor, adapt a library rule to your environment, or draft a detection straight from a KEV CVE card. Context is derived server-side from where you are — never supplied by the model.

Autonomy, triggered deliberately

Background runs start from defined events: a new evidence gap is routed by its fix type to the right specialist — Coverage Analyst, Detection Engineer, or Data Engineer; a newly published KEV CVE wakes the Detection Engineer; a SOC case can be sent to "Investigate with AI." Each run produces the same artifact as an interactive session — a proposal with evidence, waiting for a human verdict.

Autonomy you can audit

Talon Studio's answer to the trust gap is architectural. Each risk the industry has named — excessive agency, hallucinated content, unauditable actions, prompt injection, runaway cost — is met with a specific, enforced mechanism rather than a policy document.

NAMED RISK	INDUSTRY FRAMING	TALON STUDIO MECHANISM
Excessive agency	OWASP LLM Top 10 (2025): excessive functionality, permissions, autonomy[28]	Propose-only invariant enforced in the service layer: merge of a change request throws unless an authorized human approved it. No agent can push to a protected branch, approve, merge, or deploy. Tools are least-privilege per agent; readonly users yield read-only agents.
Hallucinated detections	Plausible-but-wrong rules create silent coverage gaps (NIST GenAI Profile; Sublime, 2025)[29,31]	AI output faces the same validation as human work — Atomic Red Team ground truth through pySigma, recorded as evidence on the change request — and a release gate blocks detections the connected SIEM cannot run. AI-synthesized test events are labeled `synthetic_ai` and never counted as authoritative evidence.
Unauditable actions	Agentic decision-making "often lacks clear traceability," weakening accountability (ISACA, 2025)	Every agent run is recorded step-by-step with its tool calls; every governed edit lands in the user audit log; every merged proposal exists as a Git commit with full provenance. Nothing the fleet does is off the record.
Prompt injection	Retrieved content steering the agent (OWASP; NSA MCP guidance, 2026)[28,29]	The system prompt is immutable and code-owned. The editable agent file, loaded skills, and retrieved content are all injected as explicitly untrusted data. External MCP actions require per-action human approval.
Runaway autonomy & cost	Gartner: cost and inadequate risk controls drive agentic project failure[19]	Per-org token budgets (refusals return 429), per-org concurrency caps, daily run limits, a stale-run reaper, and a global kill switch (`AGENT_RUNS_DISABLED`).
Data sovereignty	Data privacy is the top AI-adoption barrier in the SOC (Prophet survey, 2025)	Bring your own model — including fully self-hosted endpoints — on a self-hosted platform. Keys encrypted at rest; secrets auto-redacted from logs; no raw log events retained anywhere in the appliance.

Autonomy is a dial, not a switch

Agents can be triggered manually, assigned work from a kanban board, or run in the background against new coverage gaps and newly published KEV CVEs. The autonomy tier of every built-in agent is *propose-only*; what scales up with trust is the volume of work proposed — never the permission to act alone.

The learning loop stays human

Reviewer feedback — a thumbs-down with a reason, an edited proposal, a rejected change request — is captured and recalled as lessons in future runs. The fleet gets better at proposing precisely because humans keep judging the proposals.

> **"AI is allowed to suggest fixes, but only from structured gap evidence. AI is not a source of truth."**
>
> TALON STUDIO ENGINEERING PRINCIPLE · EVIDENCE ENGINE DESIGN DOCUMENT

SECTION 07 · SECURITY & DEPLOYMENT POSTURE

Built like the appliance it protects

A platform that governs your detections must clear a higher bar than the tools it manages. Talon Studio ships secure-by-default, treats its own supply chain as attack surface, and keeps every byte of durable state on a volume you control.

Secure by default

- **Refuses to boot insecurely.** Production or network-exposed deployments fail startup while placeholder secrets, default database credentials, or the default admin account remain.
- **Secrets encrypted, logs scrubbed.** SIEM, cloud, and AI credentials are encrypted at rest, with credential rotation from the UI; the logger auto-redacts tokens, keys, and credentials before any line is written.
- **Minimal exposure.** One HTTP port; embedded Postgres and pySigma bind to localhost only; TLS terminates at your reverse proxy; rate limits and timing-safe token comparisons on sensitive endpoints.
- **Nothing to breach.** No raw log events are ingested or retained — historical telemetry samples are purged on startup by design.
- **Identity done properly.** OIDC SSO on licensed tiers (GitHub, Google, Microsoft Entra), MFA, four org roles and three workspace roles over a six-capability matrix, custom roles, and platform-wide audit logging.

Supply-chain discipline

Every release image is published to GHCR with an SBOM, Cosign signature, provenance attestation, and Trivy scan. CI gates include a generated route-security inventory (auth regressions fail the build), dependency scanning, and Docker smoke suites for persistence, backup-restore, and upgrade paths.

Run it anywhere

Docker appliance — DEFAULT

One container plus one `/data` volume. Backup and restore is a volume copy. Pinned, signed image tags.

Kubernetes — HELM CHART

OCI-distributed chart for cluster deployments; external Postgres and split services available as enterprise options.

Air-gapped & regulated — OFFLINE LICENSE

Signed Ed25519 offline license format; self-hosted AI model support keeps even the copilot inside your boundary.

Operable from day one — OBSERVABILITY

Prometheus metrics endpoint, liveness/readiness probes, structured JSON logging with an in-app log viewer, documented backup, upgrade, and rollback runbooks.

> Your rules never leave your infrastructure — and neither do your keys, your Git history, or your telemetry metadata.
>
> THE SELF-HOSTED CONTRACT

SECTION 08 · ROADMAP & POSITIONING

Where the platform is headed

Talon Studio is in open beta — and deliberately candid about what is live versus what is landing next. The direction is fixed: deepen the closed loop across every SIEM, and scale governed agentic assistance on top of it.

> **"The only detection engineering platform you can own — YAML in your Git, on your infrastructure or our cloud, with AI that runs on your models and can't act without approval."**
>
> TALON LABS · PRODUCT VISION, 2026 — CLOUD DELIVERY IS A ROADMAP ITEM BELOW

Shipped today

- Single-container appliance with signed GHCR releases, Helm chart, one-click Render deploy
- Git-backed workspaces with branch-based change requests and human approval gates
- Forge library tracking SigmaHQ, with pre-converted Splunk / Sentinel / Elastic / Chronicle dialect tabs
- Splunk closed loop: live ACS/REST deployment, continuously verified in CI against a live Splunk; artifact export for the other three targets
- Atlas ATT&CK coverage from SIEM evidence; Talon Threat (KEV, profiles, feeds); SOC, Search, and Flow surfaces
- Claw agent fleet: 10 propose-only agents, RAG knowledge base, agent memory, feedback loop
- Rule health & tuning: scored grades (broken / silent / noisy / stale) for every installed rule, trend snapshots, tuning proposals via change requests
- Public API v1 with scoped tokens; Talon Studio as an MCP server (32 tools); org-level RBAC, SSO, MFA, audit

In development

- NEXT Sentinel closed loop — live deployment beyond export, the top competitive gap
- NEXT "Adopt from SIEM" import — onboard existing rule estates into detection-as-code
- NEXT Forge builder redesign — guided authoring with live multi-SIEM translation
- PLAN Replay simulation range — richer scenario-level validation depth
- PLAN Talon Cloud — the same image, hosted; self-hosted remains first-class
- PLAN Agentic SOC production plan — governed autonomous triage, evidence & safety layers, retrieval at scale

Roadmap items reflect current planning documents and are subject to change. Talon Studio today is an early-access product: the Splunk loop is live, other SIEM targets are export-only, and the platform has not yet completed an external security audit — the internal plan for one is published in the repository.

Community FREE

Self-hosted starter: 2 users, 1 workspace, 1 connected SIEM, 1 deploy target. The full lifecycle, on your hardware.

Team LICENSED

For working detection teams: 10 users, 5 workspaces, 5 SIEM connections and deploy targets. All paid licenses are signed Ed25519 offline keys — no phone-home, air-gap friendly.

Enterprise BY AGREEMENT

Full Public API v1 write access and token volume, the MCP server surface, and ServiceNow ITSM integration.

CONCLUSION

Own your detections. Govern your agents.

The industry's own measurements tell a consistent story: enterprises collect enough telemetry to detect nearly everything and actually detect about a fifth of it; rules break silently; a single detection takes a week to ship while adversaries break out in minutes. The discipline's answer — detection-as-code — is proven but stalls at exactly the point where it starts to demand platform engineering.

Talon Studio absorbs that burden: a self-hosted appliance that makes versioning, testing, review, and multi-SIEM deployment the default path rather than an aspiration, with coverage measured from evidence and validation recorded as fact.

The same substrate answers the harder question agentic AI now poses. When detection content is code behind a validation gate and a human approval, an AI agent's ambition is structurally bounded: it can propose all day, and it can merge nothing. That is the pattern the field's most credible practitioners converged on independently — and Talon Studio enforces it in the service layer, on models you choose, on infrastructure you own.

The teams that industrialize detection engineering now will be the ones who can safely hand volume to agents later. The substrate decides.

REFERENCES

1. CardinalOps — State of SIEM Detection Risk, 5th annual report, 2025.

2. CardinalOps — State of SIEM Detection Risk, 4th annual report, 2024.

3. Picus Security — Blue Report 2025; 160M+ attack simulations (vendor research).

4. SANS Institute / Anvilogic — State of Detection Engineering 2026 (n=307).

5. SANS Institute / Anvilogic — State of Detection Engineering 2025 (n=264).

6. IBM / Ponemon Institute — Cost of a Data Breach Report, 2025.

7. Vectra AI — State of Threat Detection, 2023 (vendor research).

8. SANS Institute — Detection & Response Survey, 2025.

9. SANS Institute — SOC Survey, 2025.

10. Tines — Voice of the SOC Analyst, 2022; Voice of the SOC, 2023.

11. Mandiant (Google Cloud) — M-Trends 2025 and M-Trends 2026.

12. A. Chuvakin — “Can We Have Detection as Code?”, 2020.

13. Palantir — Alerting & Detection Strategy (ADS) Framework, 2017.

14. K. Bailey — Detection Engineering Maturity Matrix, 2021.

15. SigmaHQ — Sigma Specification v2.0 (2024) and SigmaHQ rule repository.

16. Help Net Security — “Cutting the cost of SIEM rule conversion”, May 2026.

17. Gartner — Market Guide for Adversarial Exposure Validation, 2026 (as cited by Picus, Cymulate, Hadrian).

18. Gartner — Hype Cycle for Security Operations, 2025 (as cited by named vendors).

19. Gartner — press release, June 25, 2025: >40% of agentic AI projects canceled by end-2027.

20. Gartner — Cybersecurity Innovations in AI Risk Management and Use Survey, 2025.

21. Microsoft — Security Copilot randomized controlled trials, 2024; Phishing Triage Agent RCT (arXiv:2511.13860), 2025.

22. Google Cloud — Security Operations agents, incl. Detection Engineering Agent, previewed April 2026 at Cloud Next (vendor-reported).

23. CrowdStrike — Charlotte AI Detection Triage GA, 2025 (vendor-reported).

24. Detection at Scale #75 — J. Nettesheim (Block CISO) on Goose + Panther MCP, February 2026.

25. J. Naglieri — “The AI-Powered Detection Engineer”, Detection at Scale, 2025.

26. Panther — AI SOC platform & detection-engineering pages, 2025–26 (vendor-reported).

27. Anvilogic — “Redefining Detection Engineering with Agentic AI”, July 2025.

28. OWASP GenAI Security Project — Top 10 for LLM Applications 2025; Agentic AI: Threats & Mitigations, 2025.

29. NIST — AI RMF and Generative AI Profile (AI 600-1), 2024; NSA — CSI: MCP Security, June 2026.

30. CrowdStrike — Global Threat Report 2026; Anthropic — disclosure of AI-orchestrated espionage campaign, November 2025.

31. Sublime Security — “Evaluating LLM Generated Detection Rules in Cybersecurity” (arXiv:2509.16749), 2025.

32. Vendor announcements, 2024–26: Microsoft Security Copilot agents & Sentinel MCP server (2025); Google Cloud Next \'26; CrowdStrike Fal.Con (2025); Palo Alto Cortex AgentiX (2025); SentinelOne Purple AI Athena (2025); public AI-SOC funding announcements.

Product facts: Talon Studio repository documentation and source, July 2026. Vendor-reported figures are labeled as such throughout.

TALON

DETECTION ENGINEERING, AS CODE

About Talon Labs. Talon Labs builds Talon Studio, the self-hosted, vendor-neutral detection engineering platform: your detections as code in your Git, validated with evidence, compiled to your SIEMs, with AI that proposes and never merges. Deploy from a single signed container image — on your infrastructure.
