THE AI TOOLCHAIN NO. 002

Head

THE WEEK'S TOP PICKS

PUBLISHED AUGUST 28, 2026 · FRIDAYS

// SHARE THIS WEEK ▽ Top 5 ▽ Top Pick

The best tooling updates that shipped this week.

// HOW THESE PICKS ARE MADE

Every feature release from the 154 tools on our watchlist goes into the daily newsletter. Once a week we read the whole field side by side and choose our top picks, judged on two questions: **how deep and complete is the single best capability in the release**, and **how much it changes what you can actually do**.

A tool is judged on everything it shipped that week, so a project that releases daily gets credit for the sum — and still only takes one slot. We would rather run a short list than a padded one.

// THE PICKS

01	**Giskard**	AI/LLM Security
02	**AI-Infra-Guard**	AI/LLM Security
03	**SkillSpector**	AI/LLM Security
04	**Promptfoo**	AI/LLM Security
05	**ToolHive**	AI/LLM Security

01 Giskard

5 RELEASES · 2026-08-06 → 2026-08-26

AI/LLM Security

Open-Source Evaluation & Testing library for LLM Agents

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 5/5

Ships `vulnerability_scan`, an autonomous red-teaming function that probes an agent across the OWASP LLM Top-10 (prompt injection, harmful content, stereotypes, misinformation) and reports findings — plus new LLM-judge checks (Toxicity, AnswerRelevance), regex/composition operators, and JUnit XML export for CI gating. The v3 rewrite into installable giskard-checks/agents/core packages makes it adoptable piecemeal.

> Giskard shipped a full v3 rewrite as a modular monorepo (`giskard-checks` , `giskard-agents` , `giskard-core` , `giskard-scan` , `giskard-llm`) with async Scenario/Suite testing APIs, built-in LLM-as-judge checks, and an OWASP LLM Top-10 vulnerability scanner, while dropping v2 APIs, requiring Python 3.12+, and tightening scenario validation and typing.

└──▷ WHAT SHIPPED · 9 FEATURES most completely described first ? what's the number?

01 v3 monorepo rewrite and dropped v2 APIs

BREAKING

90

Giskard v3 is a full rewrite structured as a modular monorepo with three installable packages — `giskard-checks` , `giskard-agents` , and `giskard-core` — plus `giskard-scan` (via `pip install 'giskard[scan]'`) and a new provider-agnostic LLM routing layer `giskard-llm` replacing litellm, with provider extras such as `pip install 'giskard[openai]'` and `pip install 'giskard[anthropic]'` . It requires Python 3.12+, drops v2 APIs (`giskard.Model` , `giskard.Dataset` , `giskard.testing` , Giskard Hub — retained via `pip install 'giskard[llm]>2,<3'`), removes `Scenario.from_sequence` and the scenario() factory in favor of the step-based `Scenario` constructor, makes the `Interact` injection name-based instead of positional, makes Jinja parsing in `Workflow.chat` opt-in, drops templating support in conformity rules, and adds opt-out telemetry via `DO_NOT_TRACK=1` or `GISKARD_TELEMETRY_DISABLED=1` .

— Full architecture, install commands and env vars named; no single runnable demo `v3.0.0` `v3.0.0b3`

02 OWASP LLM Top-10 vulnerability scanning

NEW

86

`giskard.scan.vulnerability_scan` automates red-teaming of agents across OWASP LLM Top-10 threat categories (prompt injection, harmful content, stereotypes, misinformation), taking `target` , `description` , and `languages` parameters; a

minimal OWASP LLM suite generator covers LLM01 indirect injection. `generate_suite` accepts custom `ScenarioGenerator` instances via a `vulnerability_suite_generator_registry`, which now also exposes default registry generators on the public API and integrates third-party `ScenarioGenerator` instances into the scan pipeline. The built-in prompt-injection dataset was expanded with additional scenarios and templates for broader adversarial coverage.

Automatically red-team your agent across OWASP LLM Top-10 categories, including prompt injection probes, without writing test cases manually.

```python
import asyncio
from giskard.scan import vulnerability_scan

async def my_agent(inputs: str) -> str:
    return f"Echo: {inputs}"  # replace with your real agent

async def main() -> None:
    await vulnerability_scan(
        target=my_agent,
        description="A customer support chatbot for an e-commerce platform.",
        languages=["en"],
    )

asyncio.run(main())
```

Red-team a customer-facing chatbot against prompt injection and harmful content probes without writing any test cases manually.

```python
import asyncio
from giskard.scan import vulnerability_scan

async def my_agent(inputs: str) -> str:
    return f"Echo: {inputs}"  # replace with your real agent

async def main() -> None:
    await vulnerability_scan(
        target=my_agent,
        description="A customer support chatbot for an e-commerce platform.",
        languages=["en"],
    )

asyncio.run(main())
```

— Named function, parameters and registry with runnable red-team example `v3.0.0`

`giskard-core/v1.0.1rc1` `v3.0.0b3` `giskard-core/v1.0.1b6`

---

03 **LLM-as-judge checks in giskard-checks** NEW 72

`giskard-checks` ships built-in LLM-as-judge checks `Groundedness`, `Conformity`, and `LLMJudge` (default model `openai/gpt-4o-mini`), later joined by `AnswerRelevance` and `Toxicity`; judge results now enforce non-blank reasons so every verdict carries an explanatory rationale. `set_default_generator` also accepts plain model name strings (e.g. `'openai/gpt-4o-mini'`) in addition to generator objects, reducing boilerplate when configuring the default judge/generator.

Set the default generator for LLM-as-judge checks using a model name string instead of constructing a generator object.

python

```python
from giskard.checks import set_default_generator

set_default_generator('openai/gpt-4o-mini')
```

Run a grounded-answer eval on your agent and print a human-readable report to spot hallucinations immediately.

python

```python
import asyncio
from giskard.checks import Scenario, Groundedness

def get_answer(inputs: str) -> str:
    return my_agent(inputs)  # replace with your model/agent

async def main() -> None:
    scenario = (
        Scenario("test_capital")
        .interact(inputs="What is the capital of France?",
outputs=get_answer)
        .check(
            Groundedness(
                name="answer is grounded",
                context="France is in Western Europe. Its
capital is Paris.",
            )
        )
    )
    result = await scenario.run()
    result.print_report()

asyncio.run(main())
```

Verify a RAG answer is grounded in its retrieved context using the built-in LLM-as-judge check, catching hallucinations in CI.

```python
import asyncio
from giskard.checks import Scenario, Groundedness

def get_answer(inputs: str) -> str:
    return "Paris"  # replace with your RAG pipeline

async def main() -> None:
    scenario = (
        Scenario("test_capital_grounded")
        .interact(inputs="What is the capital of France?",
outputs=get_answer)
        .check(
            Groundedness(
                name="answer is grounded",
                context="France is in Western Europe. Its
capital is Paris.",
            )
        )
    )
    result = await scenario.run()
    result.print_report()

asyncio.run(main())
```

— Named checks and config function with runnable examples across releases

giskard-checks/v1.0.3 v3.0.0 v3.0.0b3 giskard-core/v1.0.1b6

04 **Scenario and Suite APIs for composing evals** NEW 68

The Scenario API in `giskard.checks` composes multi-turn eval interactions via chained .interact() and .check() calls with an async .run() entrypoint and .print_report() for human-readable output; scenarios support `annotations` for attaching metadata to eval steps and can run multiple times within a suite. Suite runs batches of scenarios with dynamic binding and a chainable Suite.append(), and suite reports now include scenario/check error details plus JUnit XML export for `SuiteResult`, enabling CI integration.

— Names methods and CI export but no dedicated runnable example attached v3.0.0

v3.0.0b3

THINNER COVERAGE BELOW

05 **New check types: regex, boolean composition, JSON validity** NEW 57

New check types added to `giskard-checks`: `RegexMatching` for asserting outputs match a regular expression (with ReDoS mitigation via regex timeout), `AllOf` / `AnyOf` / `Not` composition operators for combining checks with boolean logic, and a JSON validity check; checks now also support pydantic-compatible input types.

— Names each check type and a mitigation detail but no usage example `v3.0.0`

---

06 **giskard-agents generator and tooling capabilities** NEW

54

`giskard-agents` gains generator retry and timeout policies, a step-level type discriminator with tool input coercion and output serialization, a generator-as-protocol-adapter pattern for wrapping arbitrary LLM backends, personas and extended context support for `UserSimulator`, and a `metadata` parameter on the generator completion pipeline.

— Several named additions but no examples or config specifics `v3.0.0`

---

07 **Strict scenario validation and target_key rename** BREAKING

52

Persisted scenarios now undergo strict validation — unknown fields anywhere in the persisted-scenario tree are rejected, so saved scenarios with unrecognised fields fail to load after upgrade — and the field holding the value under test on every check has been renamed to `target_key`, breaking code that referenced the old field name.

— Names the renamed field and validation behaviour, no migration example

`giskard-core/v1.0.1rc1`

---

08 **PEP 561 type stubs and public type exports** IMPROVED

38

`giskard-core`, `giskard-llm`, `giskard-checks`, and `giskard-scan` now ship PEP 561 `py.typed` marker files for full static-type-checking support in downstream projects, and `giskard.types` exports tightened public `Literal` `/status` types as a stable import surface.

— Names packages and module but no example of the typing surface `giskard-core/v1.0.1rc1`

---

09 **RAG knowledge-base quality scan** NEW

34

`quality_scan` with `KnowledgeBase` support in `giskard-scan` evaluates RAG knowledge-base quality, replacing the v2 RAGET functionality.

— Named function but no parameters, mechanism or example given `v3.0.0b3`

└──▷ BREAKING ON UPGRADE

! Giskard v3 is a full rewrite; v2 APIs (`giskard.Model`, `giskard.Dataset`, `giskard.testing`, Giskard Hub) are not available in v3. Install `pip install 'giskard[llm]>2,<3'` to keep v2.

- ! Requires Python 3.12+; Python versions below 3.12 are no longer supported.
- ! `Scenario.from_sequence` is removed; use the step-based `Scenario` API instead.
- ! Templating in conformity rules is no longer supported; configurations relying on template syntax in conformity rules will break.
- ! Jinja parsing in `Workflow.chat` is now opt-in; workflows that relied on Jinja template rendering by default will no longer render templates unless explicitly enabled.
- ! The `Interact` injection is now name-based; code using positional injection patterns will break.
- ! The scenario() factory is removed; use the mutable `Scenario` constructor directly.
- ! Unknown fields anywhere in the persisted-scenario tree are now rejected (strict validation); saved scenarios containing unrecognised fields will fail to load after upgrade.
- ! The field holding the value under test on every check is renamed to `target_key`; any code that referenced the previous field name will break.
- ! Giskard v2 is no longer actively maintained; the v2 automatic tabular scan (`giskard.Model` + `giskard.Dataset`), `giskard.testing` ML test suite, and Giskard Hub are not present in v3 — install `pip install 'giskard[llm]>2,<3'` to retain v2 behavior.
- ! Requires Python 3.12+; earlier Python versions are no longer supported.

---

02 AI-Infra-Guard

4 RELEASES · 2026-07-27 → 2026-08-26

AI/LLM Security

A full-stack AI Red Teaming platform securing AI ecosystems via Agent Scan, Skills Scan, MCP scan, AI Infra scan and LLM jailbreak evaluation.

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 5/5

Adds a dedicated API security audit module (API Checker) with web-proxy integration and new detection for LLM API poisoning attacks, alongside a refactored agent red-team mutation engine (aig-agent-redteam v5.0.0) — extending the scanner from infrastructure fingerprinting into live API and agent attack surfaces.

> AI-Infra-Guard's biggest window overhaul refactors its agent red-team mutation engine into a unified mutation-attack command, adds a new API Checker security-audit module with LLM API poisoning detection, and open-sources its frontend while modularizing Agent-Scan, MCP-Scan and Skill-Scan into standalone CLI/PyPI packages — alongside steady growth in its vulnerability rule library, jailbreak methods, and agent/MCP detection skills.

└──▷ WHAT SHIPPED · 17 FEATURES most completely described first ? what's the number?

01 Agent red-team mutation engine consolidated into mutation-attack

75

BREAKING

`aig-agent-redteam` v5.0.0 introduces a refactored mutation engine; the former `workflow-attack` command is merged into `mutation-attack`, so any automation or invocation referencing `workflow-attack` must be updated to use `mutation-attack`.

— Names exact commands and merge but no example. v4.6.0

02 Skill-Scan standalone auditing package NEW

75

Repackages Skill-Scan as a standalone PyPI package `aig-skill-scan` for CI/CD integration, adding Agent Skill security auditing across 9 risk categories and achieving a SkillTrustBench top score of 0.9848.

— Names package, categories, score; no install command. v4.5.0

03 Agent-Scan OWASP detection skills expanded to 10 IMPROVED

70

v4.5.0 added 4 new detection skills to `Agent-Scan`; v4.5.1 added 5 OWASP-based skills (`agentic-supply-chain`, `cascading-failure`, `human-agent-trust`, `inter-agent-comm`, `unexpected-code-execution`) plus `web-exfiltration-detection`, bringing Agent-Scan to 10 total detection skills.

– Skill names listed but no usage instructions. v4.5.1 v4.5.0

04 **Vulnerability and detection rule library growth** IMPROVED 60

The AIG rules library expanded to cover 130 AI components with 1888 rules and new AI component fingerprints (v4.5.0), grew further to 2000+ CVE rules (v4.5.2), and was refreshed with the 2026-07-24 rule set (v4.5.1).

– Numbers given but no mechanism behind growth. v4.5.2 v4.5.1 v4.5.0

05 **MCP-Scan detection rules for secrets and deserialization** IMPROVED 60

v4.5.0 added 2 new MCP security detection rules plus ATR-derived rules covering additional attack surfaces; v4.5.1 added 4 more MCP detection rules covering hardcoded secrets and insecure deserialization.

– Rule targets named, no example detections shown. v4.5.1 v4.5.0

06 **Multi-turn jailbreak attack methods in PromptSecurity** NEW 60

Adds `Many-Shot`, `PAIR`, `GOAT`, and `ActorAttack` multi-turn jailbreak attack methods to PromptSecurity evaluation.

– Attack methods named, no configuration steps given. v4.5.1

THINNER COVERAGE BELOW

07 **Strict YAML validation for vulnerability rules** IMPROVED 55

Vulnerability rule definitions now require `id` and `severity` fields, enforced through strict YAML validation.

– Names required fields but no example shown. v4.6.0

08 **GET-only fingerprinting for vector databases** NEW 55

Adds GET-only, non-invasive fingerprint detection for Qdrant, Chroma, and Weaviate vector databases.

– Named targets and mechanism but no usage shown. v4.5.2

09 **API Checker security audit module** NEW 50

New API security audit module (API Checker) adds web proxy integration, a unified CLI command, and new detection algorithms for auditing API security.

— Names surfaces but lacks concrete command names. v4.6.0

10 **MCP-Scan dual-mode and mcp-scan-lite module** NEW 50

Modularizes MCP-Scan with dual-mode support (CLI + AIG Web) and adds a standalone `mcp-scan-lite` module.

— Names dual-mode and lite module, no commands. v4.5.0

11 **Skill-Scan Stage 2 report format changed to Markdown** BREAKING 45

Skill-Scan Stage 2 Code Audit now outputs a Markdown report instead of XML.

— Clear before/after but no reason or example. v4.5.0

12 **Agent-Scan standalone CLI module** NEW 40

Modularizes Agent-Scan as a standalone CLI with AIG integration support.

— Thin description, no CLI name or flags. v4.5.0

13 **Agentic-tool-misuse evaluation dataset** NEW 30

Adds an agentic-tool-misuse evaluation dataset to the Eval module.

— Names module and dataset, no size or use. v4.5.0

14 **Open-sourced frontend code** NEW 25

Open-sources the full frontend code along with open-source environment configuration.

— Thin mention, no scope or file details. v4.5.0

15 **DeepSeek Harness prompt injection research** NEW 20

New DeepSeek Harness prompt injection assessment research project.

— Named only, no detail on method or scope. v4.6.0

16 **SkillJack research project** NEW 20

Adds the SkillJack research project, a new AI red-teaming research capability within the platform.

— Named research project, no capability detail given. v4.5.2

17 **LLM API poisoning attack detection** NEW 15

Adds detection for LLM API poisoning attacks.

— Bare mention, no mechanism or scope given. v4.6.0

└──▷ BREAKING ON UPGRADE

! In `aig-agent-redteam`, `workflow-attack` is merged into `mutation-attack`; any invocation or automation referencing `workflow-attack` must be updated to use `mutation-attack`.

! Skill-Scan Stage 2 Code Audit now outputs a Markdown report instead of XML.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

03 SkillSpector

7 RELEASES · 2026-07-27 → 2026-08-28

AI/LLM Security

Security scanner for AI agent skills. Detect vulnerabilities, malicious patterns, security risks, prompt injection, data exfiltration, and supply-chain risks in Claude Code, Codex, and MCP skills before you install them.

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 5/5

Detects malicious bundled lifecycle hooks and directly-proven remote exfiltration of sensitive file/event content in AI skills (BH1–BH3), and adds transitive scanning of referenced skills plus inspection of hidden and nested ZIP/DOCX artifacts — closing a supply-chain gap in agent skills that manual review misses.

> SkillSpector's biggest window additions are transitive supply-chain scanning of referenced skills, MCP registry posture scanning, and inspection-ledger accounting that lets CI fail closed on incomplete scans, alongside a wave of new static findings (BH1–BH3 bundled hooks, EA5 model switching, SC8/SC9 supply-chain, AST10/TT6/DS1–DS4 deserialization) and expanded LLM provider support (Ollama, Azure OpenAI, OpenAI-compatible) with new sampling and concurrency controls.

└──▷ WHAT SHIPPED · 25 FEATURES most completely described first ? what's the number?

01 Transitive scanning of referenced skills NEW

95

New `--transitive` flag opts into scanning skills referenced by the target skill, with `--transitive-depth`, `--transitive-allow-prefix`, and `--transitive-deny-prefix` to bound traversal depth and restrict which sources are followed, catching supply-chain risk in referenced dependencies.

Scan a skill and all skills it references transitively, limiting depth and restricting allowed sources, to catch supply-chain risks in referenced dependencies.

```
$ skillspector scan ./my-skill/ --transitive --transitive-depth 2 --transitive-allow-prefix https://github.com/trusted-org/
```

— Names every flag and includes a runnable example command. v2.10.0

02 Inspection-ledger scan completeness accounting NEW

95

Adds canonical inspection-ledger accounting across static and LLM analysis stages, with per-component coverage and explicit out-of-scope records in JSON and SARIF output. New `execution_successful` and `analysis_completeness.ledger_exceptions` fields let automation distinguish a complete scan from a partial or failed one; the CLI

now exits with code `2` for a fatal execution or accounting failure even when a JSON report was produced, and recursive scans propagate child scan failures into the combined report with a non-zero exit when any child fails. JSON integrations must treat invalid or missing output, a nonzero exit, or `execution_successful: false` as a blocking validation error.

Gate CI on scan completeness — block pipelines where the scan itself failed, not just where findings were found.

```
$ skillspector scan ./my-skill/ --format json --output
report.json; python3 -c "import json,sys;
r=json.load(open('report.json')); sys.exit(1 if not
r.get('execution_successful') else 0)"
```

— Names every field, exit code, and includes a CI-gating example. `v2.5.0`

---

03 **Deterministic LLM sampling controls** NEW 90

New `SKILLSPECTOR_TEMPERATURE` (values `0` – `1`) and `SKILLSPECTOR_SEED` (integer) environment variables let you pin LLM sampling for reproducible semantic analysis; both are forwarded to OpenAI-compatible and Azure OpenAI endpoints and left unset to preserve provider defaults.

Pin LLM sampling for reproducible semantic analysis results when scanning a skill against an OpenAI-compatible endpoint.

```
$ SKILLSPECTOR_PROVIDER=openai OPENAI_API_KEY="$OPENAI_API_KEY"
SKILLSPECTOR_TEMPERATURE=0 SKILLSPECTOR_SEED=42 skillspector
scan ./my-skill/
```

— Names both vars, their values/scope, and a runnable example. `v2.11.0`

---

04 **Baseline v2 fingerprints** BREAKING 85

`skillspector baseline <path>` now produces version 2 fingerprints that bind accepted findings to the scanner version, source content, and full finding evidence. Baseline files containing version 1 fingerprints are rejected on upgrade — run `skillspector baseline <path>`, review the generated version 2 entries, and commit the replacement; rules-only version 1 baselines remain supported with a warning.

After upgrading, regenerate a version 2 baseline so fingerprints bind to the new scanner version before committing.

```
$ skillspector baseline ./my-skill/ -o .skillspector-baseline.yaml
```

— Names the exact command, migration steps, and a runnable example. v2.5.0

05 **Bundled hook and permission-surface findings (BH1–BH3)** NEW

80

New BH1, BH2, and BH3 findings detect bundled lifecycle hook execution (`hooks/hooks.json`), directly proven remote transfer of sensitive event or file content, and broad or ignored project permission surfaces (`.claude/settings.json` , `.claude/settings.local.json`). Existing scans may start surfacing these findings, so review them before accepting into a baseline.

— Names every finding ID and file path they check. v2.11.0

06 **max_issue_severity field for policy gates** NEW

75

New `risk_assessment.max_issue_severity` field in JSON/SARIF output reports `NONE` when no active issue exists, giving downstream automation a single field to gate CI pipelines on.

Run a scan with finding text localized to French, then check the max severity field in the JSON report to gate a CI pipeline.

```
$ SKILLSPECTOR_OUTPUT_LANGUAGE=French skillspector scan ./my-skill/ --format json --output report.json && jq '.risk_assessment.max_issue_severity' report.json
```

— Names the exact field, its null value, and a gating example. v2.10.0

07 **langgraph-cli no longer bundled by default** BREAKING

75

`langgraph-cli[inmem]` is no longer included in the base installation; LangGraph Studio users who install only the base package must now install `skillspector[langgraph-dev]` explicitly.

— Names exact packages and the required migration step. v2.10.0

08 **Opt-in baseline auto-discovery** NEW

75

Adds opt-in automatic discovery of a top-level `.skillspector-baseline.yaml` file so CI can suppress known findings without an explicit flag; an explicitly supplied baseline via `--baseline` remains authoritative.

Let SkillSpector auto-discover the committed baseline so CI suppresses known findings without an explicit flag.

```
$ skillspector baseline ./my-skill/ -o .skillspector-
baseline.yaml
# commit .skillspector-baseline.yaml to the repo root, then in
CI:
skillspector scan ./my-skill/
```

— Names the file, the flag, and a working CI example. v2.9.5

09 **Bounded ingest limits for URLs, archives, and Git repos** NEW

75

Enforces `INGEST_MAX_BYTES` (100 MiB per-ingest cap) and `INGEST_MAX_ZIP_MEMBERS` (10,000 entries) for streamed URL downloads, zip archives, and Git repository clones, failing closed with `IngestLimitExceededError` on breach.

— Names both limits, their values, and the failure mode. v2.5.2

10 **Provider ecosystem and discovery for LLM analysis** NEW

73

Adds Ollama support for local OpenAI-compatible inference, Azure OpenAI deployment routing, and a configurable provider for other OpenAI-compatible endpoints, all selected via `SKILLSPECTOR_PROVIDER`. `skillspector scan --help` now lists every supported hosted, local, compatible, and CLI-backed provider together with its authentication path.

— Names the providers and env var but no single runnable example. v2.9.5 v2.11.0

11 **Hidden-file and nested-archive inspection (SC9)** NEW

73

Adds bounded local inspection of hidden files and ZIP-compatible nested artifacts (ZIP, DOCX, XLSX, PPTX) without extracting or executing members, raising HIGH-severity `SC9` findings for concealed executables.

— Names formats and finding ID; no direct command shown. v2.10.0

12 **Restricted-sandbox path traversal via O_PATH** IMPROVED

70

Scans can now safely traverse intermediate path components using `O_PATH` on Linux, allowing scans in restricted sandboxes where ancestor directories lack read permission, while preserving final-file and no-symlink protections.

— Explains mechanism and scope, no direct command shown. v2.11.0

13 **HTTP MCP transport restricted to remote sources** BREAKING

70

HTTP MCP clients can no longer scan local filesystem paths or supply local YARA-rule directories; use a remote repository or URL for HTTP requests, or use stdio transport for local scans.

— States exact restriction and the required workaround. v2.9.4

14 **MCP registry posture scanning** NEW

70

Adds MCP registry posture scanning via `skillspector mcp`, available by installing the `mcp` extra: `skillspector[mcp]`.

— Names the subcommand and install extra exactly. v2.5.2

15 **Localized finding text via SKILLSPECTOR_OUTPUT_LANGUAGE** NEW

60

New `SKILLSPECTOR_OUTPUT_LANGUAGE` environment variable sets the language of human-readable LLM-generated finding text across discovery analyzers, the meta-analyzer, and MCP tool-poisoning analysis.

Run a scan with finding text localized to French, then check the max severity field in the JSON report to gate a CI pipeline.

```
$ SKILLSPECTOR_OUTPUT_LANGUAGE=French skillspector scan ./my-skill/ --format json --output report.json && jq '.risk_assessment.max_issue_severity' report.json
```

— Names the env var and scope; example shows usage. v2.10.0

THINNER COVERAGE BELOW

16 **Insecure deserialization findings** NEW

55

Adds static analysis coverage for insecure deserialization patterns, surfaced as `AST10`, `TT6`, and `DS1`–`DS4` findings.

— Names every finding ID, no example or mechanism detail. v2.9.5

17 **AISOP/AISP bundle summaries** NEW

52

Adds structured skill summaries for valid AISOP/AISP bundles, rendered across terminal, Markdown, JSON, and SARIF output formats.

— Names bundle types and all four output formats. v2.10.0

18 **External model/provider selection findings (EA5)** NEW

50

New EA5 static findings flag external model or provider selection, covering silent coding-CLI account switches and top-level model pins.

— Names finding ID and two triggering patterns, no example. v2.10.0

19 **YARA matching accuracy fixes** IMPROVED 50

YARA matches are now mapped back to source lines using byte offsets, so non-ASCII content is reported at accurate locations. The destructive-autonomy YARA post-filter is now scoped to SkillSpector's built-in rule namespace, preventing custom YARA rules that reuse a built-in rule name from being incorrectly post-filtered.

— Explains both fixes' mechanisms but offers no user-facing action. v2.9.5

20 **Python bytecode supply-chain finding (SC8)** NEW 50

Adds a HIGH-severity `SC8` finding when a skill ships Python bytecode or `__pycache__` content, expanding supply-chain coverage.

— Names the finding ID and severity, no example command. v2.9.4

21 **Whitespace-padding prompt-injection detection** NEW 32

Adds detection for whitespace-padding techniques used to conceal prompt-injection instructions inside skill content.

— States the technique detected but no finding ID or example. v2.9.4

22 **Dynamic analyzer discovery and validation** IMPROVED 30

Adds dynamic analyzer discovery and validates risk-score inputs against the registered analyzer set.

— Thin description with no named surface or example. v2.10.0

23 **allowed-tools recognized as least-privilege guidance** IMPROVED 27

Remediation text and documentation now recognise `allowed-tools` as valid least-privilege permission guidance.

— Names the field but gives no further mechanism. v2.9.4

24 **npm lockfile scanning** NEW 23

SkillSpector now scans npm lockfiles as part of its supply-chain risk checks.

— Only named in the release summary, no mechanism given. v2.11.0

25 **Skill Inspector companion guide** NEW 15

Ships a Skill Inspector companion skill guide.

— Bare mention with no detail on contents or use. v2.9.4

└──▷ ALSO FROM THESE RELEASES

Serialize LLM analyzer requests to avoid bursting a rate-limited provider such as one with a strict requests-per-minute cap.

```
$ SKILLSPECTOR_MAX_LLM_CONCURRENCY=1 skillspector scan ./my-skill/
```

Persist a reduced concurrency limit for all scans run in a Docker-based workflow by adding it to your `.env` file.

```
$ SKILLSPECTOR_PROVIDER=anthropic
ANTHROPIC_API_KEY=sk-ant-...
SKILLSPECTOR_MAX_LLM_CONCURRENCY=3
```

└──▷ BREAKING ON UPGRADE

! Existing scans may now surface new BH1, BH2, or BH3 findings for supported bundled hook and settings files (`hooks/hooks.json` , `.claude/settings.json` , `.claude/settings.local.json`); review those findings before accepting them into a baseline.

! `langgraph-cli[inmem]` is no longer included in the base installation; LangGraph Studio users who install only the base package must now install `skillspector[langgraph-dev]` explicitly.

! HTTP MCP clients can no longer scan local filesystem paths or supply local YARA-rule directories; use a remote repository or URL for HTTP requests, or use stdio transport for local scans.

! Baseline files containing version 1 fingerprints are rejected on upgrade. Run `skillspector baseline <path>` , review the generated version 2 entries, and commit the replacement; rules-only version 1 baselines remain supported with a warning.

! JSON integrations must now treat invalid or missing output, a nonzero process exit, or `execution_successful: false` as a blocking validation error and surface `analysis_completeness.ledger_exceptions` for diagnosis.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

04 Promptfoo

1 RELEASE · 2026-08-23

AI/LLM Security

Promptfoo is a testing framework for evaluating and benchmarking language model prompts and applications across multiple providers.

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 4/5

Red team scans now support multiple input variables for probing systems with complex input structures, and add a Telecom industry plugin plus a RAG Source Attribution plugin to test whether retrieval systems properly attribute sources; a Transformers.js provider also enables fully local model testing in Node or the browser.

> Promptfoo's snapshot-20260823 release adds an adaptive rate limit scheduler, a Transformers.js provider for local model inference, new video-generation and AI gateway providers, a Telecom red team plugin, and support for multi-input red team scanning, alongside a broad set of CLI, config, and UI additions.

└──▷ WHAT SHIPPED · 14 FEATURES most completely described first ? what's the number?

01 New CLI flags for logs, descriptions, sharing, and env vars NEW

78

Adds a `promptfoo logs` command for viewing log files directly from the CLI; a `-d` / `--description` flag on `redteam generate` commands for labeling generated scans; `--share` and `--no-share` flags on Model Audit for controlling cloud sharing; `$VAR` syntax support in file paths for referencing environment variables in configuration; support for multiple `--env-file` flags to load several environment files in a single invocation; and an `--extension` CLI flag to load extensions via command line.

View Promptfoo runtime logs without digging through the filesystem — useful when debugging provider errors or rate limit behavior.

```
$ promptfoo logs
```

Cap test-case generation per strategy and label a red team run for tracking in CI.

```
$ promptfoo redteam generate -d 'nightly-telecom-scan' --config redteam.yaml
```

Load separate secret and environment config files in one run, and reference a secrets file path via environment variable.

```
$ promptfoo eval --env-file .env --env-file .env.secrets -c
promptfooconfig.yaml
```

— Six named flags/commands with runnable examples for three `snapshot-20260823`

02 **Eval config additions: numTests, sessionId, word-count, __count, test-level filters**

NEW 73

Adds a `numTests` config key to cap the number of test cases generated per red team strategy; `metadata.sessionId` as a surfaced variable column in eval result tables and exports; a `word-count` assertion type for validating response word counts; a `__count` variable for use in derived metrics to compute averages; and test-level prompts filter, providers filter, and per-test structured output configuration at the individual test case level.

Cap strategy test-case volume and assert word count in a single test config — useful for constraining long-running red team scans.

yaml

```yaml
strategies:
  - id: jailbreak
    numTests: 20
assertions:
  - type: word-count
    threshold: 200
```

— Five named config surfaces, runnable example for two of them `snapshot-20260823`

03 **xAI Voice provider config overrides and function calls** IMPROVED

60

Adds `apiBaseUrl` and `websocketUrl` override config and function call support to the xAI Voice provider.

— Names exact config keys, no example of usage shown `snapshot-20260823`

THINNER COVERAGE BELOW

04 **Transformers.js provider for local inference** NEW 50

New Transformers.js provider to run Hugging Face models locally in Node.js or the browser.

— Names the provider and runtime but no setup steps `snapshot-20260823`

05 **Telecom and RAG Source Attribution red team plugins** NEW

50

New Telecom red team plugin for industry-specific testing of telecommunications AI systems, and new RAG Source Attribution plugin to test whether RAG systems properly attribute sources in responses.

— Explains purpose of each plugin without invocation details `snapshot-20260823`

06 **Adaptive rate limit scheduler** NEW

45

New adaptive rate limit scheduler automatically adjusts concurrency based on provider rate limits and response headers.

— Mechanism described but no config key or flag to invoke it `snapshot-20260823`

07 **AWS Bedrock and Azure AI Foundry video providers** NEW

45

New AWS Bedrock Video provider supporting `Nova Reel` and `Luma Ray 2` video generation, and new Azure AI Foundry Video provider for `Sora` video generation.

— Names specific models but no config or usage steps `snapshot-20260823`

08 **OpenAI Codex provider collaboration mode and tracing** IMPROVED

45

Adds `collaboration_mode` support and integrated tracing to the OpenAI Codex provider.

— Names the field but not how tracing is configured `snapshot-20260823`

09 **Eval results UI filters and provider config hover** NEW

45

Adds a user-rated filter in the UI to show only manually rated eval results, and a provider config hover in eval results UI to view provider configuration details inline.

— UI additions named but no exact navigation path given `snapshot-20260823`

10 **Native session endpoint support in HTTP provider** IMPROVED

35

HTTP provider gains native session endpoint support for stateful conversations.

— Describes capability without endpoint or config detail `snapshot-20260823`

11 **Fork PR and comment-triggered code scanning** IMPROVED

35

Code scanning adds support for fork pull requests and comment-triggered scans.

– Names two scan triggers but no setup instructions snapshot-20260823

12 **Vercel and Cloudflare AI Gateway providers** NEW 30

New Vercel AI Gateway and Cloudflare AI Gateway providers for routing requests.

– Bare naming of two new providers with no usage detail snapshot-20260823

13 **Multi-input variable support in red team scans** IMPROVED 30

Red team scans now support multiple input variables for testing systems with complex input structures.

– States capability but no config example for multi-input use snapshot-20260823

14 **Automatic retries for transient errors in Enterprise** IMPROVED 25

Automatic retries for transient 5xx errors in Enterprise environments.

– Thin one-line description with no configuration detail snapshot-20260823

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

05 ToolHive

7 RELEASES · 2026-07-28 → 2026-08-27

AI/LLM Security

ToolHive is an enterprise-grade platform for running and managing Model Context Protocol (MCP) servers.

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 4/5

Blocks MCP plugin upgrades whose signer identity differs from the lock file (exit code 4) unless explicitly confirmed, stopping silent malicious plugin swaps, and adds private-CA trust for embedded auth servers — building on the prior release's end-to-end Sigstore bundle verification for plugin artifacts.

> Across seven releases (v0.41.0–v0.46.0), ToolHive built out a full skill and AI-plugin lifecycle — lock files, Sigstore-verified sync/upgrade/push, and a plugin CLI/registry — while layering in RFC 8693 delegated token exchange, end-to-end MCP 2026-07-28 'Modern' spec support, and tighter vMCP tool-aggregation and Cedar authorization controls.

└──▷ WHAT SHIPPED · 27 FEATURES most completely described first ? what's the number?

01 **Skill lock file, sync, upgrade and push** NEW 93

`thv skill sync` restores a project's pinned skill set and verifies on-disk content in CI, and `thv skill upgrade` re-resolves pinned skills to newer content without silent drift; both pin installs in `toolhive.lock.yaml` (including Sigstore provenance fields) with typed exit codes and a pre-install confirmation gate. Sigstore signature verification was added for skills at install, sync, and upgrade time. Signing of pushes was enabled by default (the lock feature gate removed), and lock provenance now records certificate ref and runner, enforced during skill verification. The `--clients` flag on `thv skill sync` and a `{"clients": [...]}` body field on `POST /api/v1beta/skills/sync` let operators scope which skill-supporting clients are targeted; `qoder` was added as the 18th skill-supporting client, materializing skills into `<project>/.qoder/skills/`. `thv skill push` now signs keylessly by default and requires exactly one of `--key`, `--identity-token`, or `--no-sign`.

Restore and verify a project's pinned MCP skill set in CI to ensure every machine uses exactly the locked versions.

```
$ thv skill sync
```

Re-resolve all pinned skills to newer content and update `toolhive.lock.yaml` without silent drift.

```
$ thv skill upgrade
```

Sync skills to only a specific client in CI so that the addition of the new `qoder` client does not cause unexpected drift and a non-zero exit.

```
$ thv skill sync --check --clients claude-code
```

Scope a REST skill sync to specific clients so CI pipelines do not unexpectedly expand to all skill-supporting clients after upgrade.

```
$ curl -X POST http://127.0.0.1:8080/api/v1beta/skills/sync \
  -H 'Content-Type: application/json' \
  -d '{"clients": ["claude-code", "cursor"]}'
```

— Names commands, flags, lock file and endpoint with runnable examples. `v0.45.0` `v0.43.0` `v0.42.0` `v0.41.0`

---

02 **AI plugin CLI, registry and lock provenance** NEW 83

`thv ai-plugin` ships a full CLI and REST API for end-to-end AI-tool plugin management, alongside a registry catalog for discovering and managing plugins. A `plugins` key was added to the lock file schema to track plugin entries, backed by a new `PluginLockService` and a managed install flag, with lock provenance recording certificate ref and runner. Plugin artifacts gained end-to-end Sigstore bundle verification; stored bundles and git commit payloads/signatures are rejected with HTTP 422 above 1 MiB. `thv ai-plugin upgrade` gained an `--allow-signer-change` flag so operators can explicitly confirm a signer rotation — without it, upgrades whose signature identity differs from the lock file, or that are unsigned, are blocked with exit code 4 and `signer-change-blocked`, gated behind the experimental `TOOLHIVE_PLUGINS_LOCK_ENABLED` environment variable.

Upgrade a plugin while explicitly approving a signer identity change — required when the new release is signed by a different identity than what the lock file recorded.

```
$ thv ai-plugin upgrade --allow-signer-change <plugin-name>
```

— Names CLI, flag, env var and exit code; REST path not given. `v0.46.0` `v0.45.0` `v0.43.0` `v0.42.0`

---

03 **vMCP tool aggregation visibility and composite tool annotations**

NEW

80

Adds `aggregation.defaultToolVisibility: deny` to vMCP config so only workloads explicitly listed in `aggregation.tools` have their tools advertised, closing the fail-open gap in tool aggregation; on the Modern (2026-07-28) path, tools excluded via `filter`, `excludeAll`, or `excludeAllTools` are no longer directly callable — `tools/call` now returns `-32602` at HTTP 400 instead of executing. Composite tools now support MCP tool annotations (`readOnlyHint`, `destructiveHint`, `idempotentHint`, `openWorldHint`), with a conservative fail-closed safety floor derived from the workflow's step tools when none are set explicitly.

Lock down a vMCP group so new workloads are hidden by default and only explicitly listed ones expose tools.

yaml

```yaml
name: "engineering-vmcp"
groupRef: "engineering-team"
aggregation:
  defaultToolVisibility: deny
  tools:
    - workload: github
    - workload: jira
```

— Exact config keys and annotation names with a runnable example. `v0.42.1`

---

04 **Private CA trust and plain-HTTP opt-in for OIDC upstreams** NEW

79

A `caBundleRef` field was added to OIDC and OAuth2 upstream specs, letting an embedded auth server trust a private CA for discovery, token, user-info, and dynamic client registration calls to that upstream only; the `operator-crds` chart must be upgraded to 0.46.0 before or with the operator chart, or a stale CRD silently prunes the field. An `insecureAllowHTTP: true` field under `spec.inline` in `MCPOIDCConfig` lets operators opt in to plain-HTTP issuer and JWKS URLs for dev/test; production configs must use HTTPS, and inline configs with a plain-HTTP, malformed, or scheme-less `issuer` / `jwksUrl` now flip to `Valid=False` on reconcile, blocking dependent `MCPServer`, `MCPRemoteProxy`, and `VirtualMCPServer` resources.

Point an embedded auth server at an in-cluster IdP behind a private CA so ToolHive can reach its OIDC discovery and token endpoints.

yaml

```yaml
caBundleRef: my-internal-ca-secret
```

Allow a dev/test inline OIDC config pointing at an in-cluster Keycloak over HTTP to pass URL validation after upgrading.

```yaml
apiVersion: toolhive.stacklok.dev/v1beta1
kind: MCPOIDCConfig
metadata:
  name: keycloak-auth
spec:
  type: inline
  inline:
    issuer: http://keycloak:8080/realms/toolhive
    jwksUrl:
http://keycloak:8080/realms/toolhive/protocol/openid-
connect/certs
    insecureAllowHTTP: true
```

— Exact field names with working config examples for both. `v0.46.0` `v0.42.1`

05

RFC 8693 delegated token exchange in auth server NEW

74

The embedded authorization server's token endpoint now supports RFC 8693 token exchange, with delegated token audience bounded by the subject token; subject tokens from trusted external OIDC issuers (Keycloak, Entra, Okta) can be validated for exchange, and audit logs capture the RFC 8693 `act` claim and full delegation chain. `trusted_issuers` was added to the embedded auth server config so agents can exchange subject tokens from external OIDC providers for ToolHive-scoped delegated tokens under a fail-closed RFC 8693 consent policy. `actor_token` support was added for delegated-identity scenarios where an acting party is distinct from the subject. RFC 8693 delegate (token-exchange) clients configured in `vmcp-config.yaml` are now reachable and usable for downstream token exchange workflows.

Configure an OAuth token-exchange backend in vmcp so that downstream calls use RFC 8693 delegate tokens — now fully reachable after this release.

```yaml
backends:
  github:
    type: token_exchange
    tokenExchange:
      tokenUrl:
"https://keycloak.example.com/realms/myrealm/protocol/openid-
connect/token"
      clientId: "vmcp-github-exchange"
      clientSecretEnv: "GITHUB_EXCHANGE_SECRET"
      audience: "github-api"
      scopes: ["repo", "read:org"]
```

— Names RFC and config keys but gives no endpoint path. v0.44.0 v0.43.0 v0.42.1 v0.41.0

06 **Workload reference fields removed from six config CRDs** BREAKING

74

`status.referencingWorkloads`, `status.referenceCount`, and the References printer column are removed from all six config CRDs (`MCPOIDCConfig`, `MCPAuthzConfig`, `MCPExternalAuthConfig`, `MCPToolConfig`, `MCPWebhookConfig`, `MCPTelemetryConfig`); replace any automation reading them with workload field queries via `-o json | jq`.

— Names all six CRDs and gives exact migration query. v0.42.0

07 **StorageVersionMigrator enabled by default for operator Helm chart**

73

BREAKING

The StorageVersionMigrator controller is now enabled by default (`operator.features.storageVersionMigrator: true`); namespace-scoped Helm installs (`operator.rbac.scope=namespace`) now fail `helm upgrade` at render time unless `operator.features.storageVersionMigrator: false` is explicitly set.

Opt out of the StorageVersionMigrator on a namespace-scoped Helm install to avoid a broken `helm upgrade`.

yaml

```yaml
operator:
  rbac:
    scope: namespace
  features:
    storageVersionMigrator: false
```

— Exact config key and runnable Helm example provided. v0.41.0

08 **MCP 2026-07-28 'Modern' spec support across proxies** NEW

67

Supports the MCP 2026-07-28 stateless ("Modern") spec revision end-to-end across transport proxies, transparent proxy, and Virtual MCP, bridging era-mismatched client×backend combinations; Modern client-facing dispatch is gated per capability instead of a global kill-switch, with listen-stream support and pagination. Adds opt-in strict MCP-Protocol-Version header validation for the streamable proxy; the readiness probe now sends the current MCP protocol version instead of a hardcoded 2024-11-05; guarantees `tools/list` pagination completeness for aggregated sets exceeding 1,000

tools; tool definitions carrying invalid `x-mcp-header` annotations (SEP-2243) are now rejected; W3C trace context propagates through outbound MCP `_meta` (SEP-414); backend `list_changed` notifications are consumed and propagated to clients for tools, resources, and prompts; multi-line Modern SSE events can now be parsed for better transport compatibility.

— Many named specs/headers but no command a reader can run. `v0.43.0` `v0.41.0`

---

09 **Package name validation blocks Dockerfile injection** BREAKING

67

Package names in `npx://`, `uvx://`, and `go://` references are now validated against `[A-Za-z0-9@/:._+=~[\]-]` at build time, blocking shell metacharacter injection into generated Dockerfiles; names outside this pattern now fail at build time with an "invalid package name" error instead of being interpolated into the Dockerfile.

— Exact regex and failure mode given, no example call shown. `v0.45.0`

---

10 **thv serve management API request hardening** BREAKING

64

`thv serve` management API now enforces `Content-Type: application/json` on state-changing requests over TCP, and adds Origin validation with a loopback-only allowlist on those same listeners; callers omitting the header receive `415 Unsupported Media Type`.

Create a workload via the management API now that `Content-Type: application/json` is required on state-changing TCP requests.

```
$ curl -X POST http://127.0.0.1:8080/api/v1beta/workloads \
  -H 'Content-Type: application/json' \
  -d '{"name":"fetch","image":"ghcr.io/example/fetch:latest"}'
```

— Exact header requirement and status code with example call. `v0.45.0`

---

11 **Workload runtime_config validation and env application fix**

64

BREAKING

`runtime_config.build_with` on `npx://` / `go://` images is now a `400 Bad Request`, and `runtime_config.runtime_env` is now actually applied via `POST /api/v1beta/workloads` (previously silently discarded).

— Names endpoint and fields but shows no example request. `v0.45.0`

---

12 **CLI API timeout override via environment variable** NEW

62

Adds `TOOLHIVE_API_TIMEOUT` environment variable to override the CLI API client timeout for `thv skill` and `thv ai-plugin` commands (default is 10 minutes).

Fail CI faster when `thv skill` calls time out in a slow environment by shortening the API client timeout.

```
$ TOOLHIVE_API_TIMEOUT=30s thv skill list
```

— Names env var, affected commands, default, and example. v0.42.1

THINNER COVERAGE BELOW

13 **Virtual MCP protocol conformance, stability and timeouts**

IMPROVED 54

Virtual MCP protocol negotiation now stabilizes instead of flapping between Modern and Legacy MCP revisions. Virtual MCP is now MCP-conformant, supporting completions, resource templates, subscriptions, and mid-call server-to-client forwarding. Virtual MCP also now honours `operational.timeouts` configured values and propagates backend health changes to live sessions — a configured value below 30s now actually cuts backend calls that previously received a silent 30s default.

— Names one config key but shows no working example. v0.45.0 v0.42.0 v0.41.0

14 **Cedar policy evaluated against post-mutation MCP requests**

49

BREAKING

Cedar authorization policy is now evaluated against the post-mutation MCP request instead of the original, closing a bypass window that allowed length-preserving mutating webhook rewrites to evade authorization; audit records also reflect the post-mutation body, so operators combining a `mutating:` entry in `--webhook-config` with Cedar should re-audit policies and SIEM rules.

— Explains mechanism and impact but gives no fix command. v0.42.0

15 **OAuth/DCR hardening and client support** IMPROVED 48

OAuth token and Dynamic Client Registration (DCR) endpoints are now guarded against Server-Side Request Forgery (SSRF) attacks. DCR now supports confidential clients, expanding the OAuth client types available to the auth layer, and CIMD documents advertising unsupported grant types are now ignored rather than rejected. Dynamically registered OAuth clients now automatically renew expiring client secrets per RFC 7591/7592.

— Lists mechanisms but gives no config example to act on. v0.44.0 v0.43.0 v0.41.0

16 **JSON-RPC batch requests now rejected** 48

BREAKING

JSON-RPC batch requests (top-level arrays) are now rejected with `HTTP 400` / error code `-32600` instead of being executed; send individual requests.

— Names exact error code and status, no example call. `v0.41.0`

---

17 **Rate-limiting observability and JSON-RPC error code change**

IMPROVED 46

Adds rate-limiting observability via metrics and tracing (OpenTelemetry), covering the proxy rate-limit path. The rate-limit JSON-RPC error code changed from `-32029` to `429`; clients branching on `error.code == -32029` must match `429` instead.

— Names error codes but observability detail stays generic. `v0.43.0` `v0.41.0`

---

18 **thv llm local proxy fixes and token helper simplification**

IMPROVED 44

The `thv llm` local proxy now returns `401 token_required` instead of `502 server_error` when the stored credential has been rejected by the IdP. LLM config is now reset when the last tool is torn down, preventing stale configuration from persisting after all tools exit, and the token helper now uses a bare `thv` command, simplifying token-helper integration for LLM clients.

— Names error codes and behavior but no reproduction steps. `v0.45.0` `v0.43.0`

---

19 **Cedar authorization strict Content-Type and JWT claim normalization**

IMPROVED 43

With Cedar authorization enabled (`--authz-config`), MCP POST requests without `Content-Type: application/json` (including a missing header) now return `400` instead of being forwarded unauthorized. Multi-valued JWT claims can now be normalized to canonical space-delimited form for Cedar policies.

— Names the flag and header but mechanism stays brief. `v0.42.1` `v0.41.0`

---

20 **Telemetry providers package removed from Go API** BREAKING 40

`pkg/telemetry/providers` is deleted and two `optimizerdec` constants are removed from the Go API; out-of-tree Go importers must drop references to these before upgrading.

— Names package and constants, relevant only to Go importers. `v0.42.0`

---

21 **Recovered HTTP panics no longer logged** BREAKING 28

Recovered HTTP panics no longer produce a `slog.Error` log line or stack trace; log-based alerts on recovered panics will silently stop firing unless Sentry is configured.

— Notes a side effect but offers no remediation steps. v0.42.0

22 **macOS thv binary signed with Developer ID certificate** IMPROVED 24

Signs the macOS `thv` binary with a Developer ID certificate, removing Gatekeeper warnings for macOS users.

— Bare statement, no verification steps or version given. v0.43.0

23 **Container image trust state shown in CLI** NEW 23

Displays recorded trust state to the user in the CLI, surfacing container image trust information at runtime.

— Single line, no exact command or UI path given. v0.43.0

24 **Prometheus metrics moved to dedicated diagnostics port** IMPROVED 21

Prometheus metrics move to a dedicated diagnostics port, controlled by a migration switch.

— No port number or switch name given, thin description. v0.45.0

25 **Build fingerprint dropped from proxy /health response** IMPROVED 21

Drops the build fingerprint from the proxy `/health` response, reducing information exposure on that endpoint.

— Single line, no mechanism or reproduction detail given. v0.43.0

26 **Multiple MCP clients sharing one stdio server** NEW 14

Enables multiple MCP clients to share a single stdio server simultaneously.

— One-line description with no mechanism or surface named. v0.42.0

27 **Opt-in Envoy network-isolation backend** NEW 14

Adds an opt-in Envoy network-isolation backend.

— Bare one-line mention, no config or mechanism given. v0.41.0

└──▷ **BREAKING ON UPGRADE**

! The `operator-crds` chart must be upgraded to 0.46.0 before or together with the operator chart; a stale CRD silently prunes the new `caBundleRef` field from applied resources instead of rejecting it.

- ! `thv serve` management API over TCP now requires `Content-Type: application/json` on state-changing requests with a body; callers omitting it receive `415 Unsupported Media Type`.
- ! Package names in `npx://`, `uvx://`, and `go://` references containing characters outside `[A-Za-z0-9@/:._+=~[\]-]` now fail at build time with an 'invalid package name' error instead of being interpolated into the Dockerfile.
- ! `thv skill sync` without `--clients` now targets every skill-supporting client; any skill locked under v0.44.0 will report as drifted on first sync after upgrade, and `thv skill sync --check` will exit non-zero in CI.
- ! `runtime_config.build_with` on `npx://` / `go://` images is now a `400 Bad Request`; `runtime_config.runtime_env` is now actually applied (was silently discarded) via `POST /api/v1beta/workloads`.
- ! `thv skill push` now returns `400` when both `--key` and `--no-sign` are supplied; exactly one of `--key`, `--identity-token`, or `--no-sign` is required.
- ! Virtual MCP now honours `operational.timeouts`; a configured value below 30 s will now actually cut backend calls that previously received the silent 30 s default.
- ! Exported Go interfaces `plugins.MaterializationAdapter`, `state.Store` writers, `storage.UpstreamTokenStorage`, and six function signatures gained required methods or changed signatures.
- ! The `thv llm` local proxy now returns `401 token_required` instead of `502 server_error` when the stored credential has been rejected by the IdP.
- ! With Cedar authorization enabled (`--authz-config`), `POST` requests without `Content-Type: application/json` (including a missing header) now return `400` instead of being forwarded unauthorized; all MCP POST clients must send `Content-Type: application/json`.
- ! vMCP tools excluded via `filter`, `excludeAll`, or `excludeAllTools` are no longer directly callable on the Modern (2026-07-28) path — `tools/call` now returns `-32602` at HTTP 400 instead of executing; un-filter the tool or wrap it in a composite tool.
- ! `MCPOIDCConfig` resources of `spec.type: inline` with a plain-HTTP, malformed, or scheme-less `issuer` or `jwksUrl` flip to `Valid=False` on next reconcile and block reconciliation of every `MCPServer`, `MCPRemoteProxy`, and `VirtualMCPServer` referencing them; add `insecureAllowHTTP: true` or switch to HTTPS.
- ! `status.referencingWorkloads`, `status.referenceCount`, and the `References` printer column are removed from all six config CRDs (`MCPOIDCConfig`, `MCPAuthzConfig`, `MCPExternalAuthConfig`, `MCPToolConfig`, `MCPWebhookConfig`, `MCPTelemetryConfig`); replace any automation reading them with workload field queries via `-o json | jq`.
- ! Cedar policy and audit records now evaluate against the post-mutation MCP request body; re-audit Cedar policies and update SIEM rules keyed on `type` or `target.name` before upgrading workloads that combine a `mutating:` entry in `--webhook-config` (or `MCPWebhookConfig.spec.mutating`) with Cedar authorization.
- ! Recovered HTTP panics no longer produce a `slog.Error` log line or stack trace; log-based alerts on recovered panics will silently stop firing unless Sentry is configured.

- ! `pkg/telemetry/providers` is deleted and two `optimizerdec` constants are removed from the Go API; out-of-tree Go importers must drop references to these before upgrading.
- ! Namespace-scoped Helm installs (`operator.rbac.scope=namespace`) now fail `helm upgrade` at render time unless `operator.features.storageVersionMigrator: false` is set, because the StorageVersionMigrator controller is now enabled by default (`operator.features.storageVersionMigrator: true`).
- ! JSON-RPC batch requests (top-level arrays) are now rejected with HTTP 400 / error code `-32600` instead of being executed; send individual requests.
- ! Rate-limit JSON-RPC error code changed from `-32029` to `429` ; clients branching on `error.code == -32029` must match `429` instead.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

// END OF TRANSMISSION

The Cyber Toolchain · This Week's Highlights · August 28, 2026
The full firehose — every feature release, every day — is in the newsletter.

▷ Subscribe