THE CYBER TOOLCHAIN NO. 015

Head

THE WEEK'S TOP PICKS

PUBLISHED OCTOBER 9, 2026 · FRIDAYS

// SHARE THIS WEEK ▽ Top 5 ▽ Top Pick

The best tooling updates that shipped this week.

▸ // HOW THESE PICKS ARE MADE SHOW

▾ // THE BRIEF HIDE

What stands out in this week's picks, grouped by what it lets you do. Every tool named links to its pick below.

PROTECT

Stop a malicious package at resolution time instead of finding it in a scan report

Dependency Firewall evaluates packages against malicious status, vulnerability severity, license and package age before they enter the build, and can block outright. Package age is the quiet one — it is the cheapest signal against a freshly published typosquat, and nothing in a post-build SCA scan catches it in time.

GitLab

PROTECT

Keep SSH signatures intact against a store-now-decrypt-later adversary

10.6 adds a hybrid post-quantum signature algorithm, plus sshd warnings when weak crypto is negotiated and tighter control over the agent socket path. The warnings matter operationally: most fleets have no idea which hosts are still falling back to legacy algorithms until something tells them.

OpenSSH

vim ? help

RESPOND

Dry-run a triage rule before it rewrites your findings backlog

Rules Engine 2.0 adds branching logic, asset-level automation and run traceability, but simulation mode is the one that changes behaviour — automated deduplication and severity rewrites are usually written conservatively because a bad rule silently buries real findings and nobody can reconstruct why.

defectdojo

IDENTIFY

See which AI agent skills are actually running in your repos, and what they execute

StepSecurity inventories agent skills per GitHub repository with their executable content and provenance; Zenity's free AI Total detonates a skill in a sandbox to show runtime behaviour. Skills are installable code with repo credentials and no review gate, so most teams currently have no list of what they have accepted.

StepSecurity · Zenity

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every change in this week's picks, in full.

all 51 changes

Do you prefer **scroll**? 👍 👎

// THE PICKS

01 **GitLab** — SAST / SCA / Code Security

02 **Zenity** — AI/LLM Security

03	**SonarQube**	SAST / SCA / Code Security
04	**Elastic Security Labs**	SIEM
05	**GitHub**	SAST / SCA / Code Security

01 GitLab

Dependency Firewall: Block risky packages before the build (2026-10-06)

SAST / SCA / Code Security

GitLab is a web-based DevOps platform providing Git repository hosting, CI/CD pipelines, and collaboration tools for software development teams.

// WHY IT MADE THE LIST — DEPTH 5/5 — IMPACT 4/5 — NOVELTY 4/5

GitLab Dependency Firewall blocks malicious, vulnerable, and non-compliant packages before they enter a build, with warn/block modes, policy-as-code, and a logged bypass across npm, pip, Maven and more — closing a gap teams could previously only watch after the fact. It ships alongside Artifact Central, an org-level registry that auto-attaches build provenance to every artifact.

> GitLab launched two major early-access/beta products on the same day: Dependency Firewall, which warns on or blocks risky packages before they enter a build using policy dimensions for malicious status, vulnerability severity, license, and package age; and Artifact Central, a new organization-level registry for Maven, npm, Docker, and OCI artifacts with built-in build provenance and `CI_JOB_TOKEN` authentication.

└──▷ **WHAT SHIPPED · 11 FEATURES** most completely described first ? what's the number?

01 Repository types and zero-copy migration path NEW 95

Artifact Central supports three repository types — Hosted (private packages your teams publish), Remote (a proxy with caching for external sources like Docker Hub or Maven Central), and Virtual (a single endpoint combining hosted and remote, with automatic fallback and cache). This enables gradual, zero-bulk-copy migration: add an existing registry as a Remote repository behind a Virtual repository, then convert to Hosted when ready — in Artifact Central, go to your organization > Artifact Central > Repositories > New Repository, select Remote, enter the existing registry URL, create a Virtual repository that includes the new remote, and update your pipeline to point at the virtual repository URL.

Add an existing external registry as a Remote repository behind a Virtual repo to start a gradual, zero-downtime migration without bulk-copying data.

📍 In Artifact Central, go to your organization › Artifact Central › Repositories › New Repository, select Remote, enter your existing registry URL, then create a Virtual repository that includes the new remote. Update your pipeline to point at the virtual repository URL.

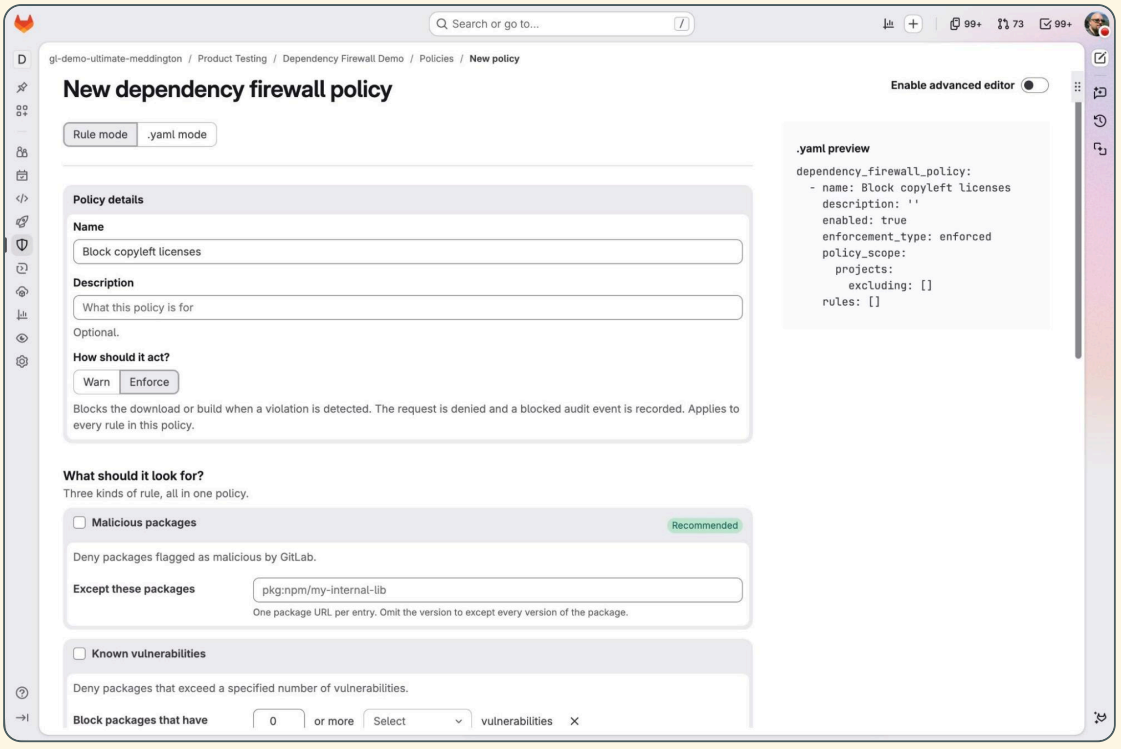
— Includes exact migration navigation steps `Every artifact your teams ship, assembl…`

02 Dependency Firewall enforcement modes and policy dimensions

NEW

85

GitLab Dependency Firewall (early access) adds `warn` and `block` enforcement modes: `warn` logs every policy match as an audit event and CI summary line without stopping the build, `block` halts the pipeline and records the reason. Policy dimensions include malicious status (checked against GitLab's malware advisory database), vulnerability severity (`critical` , `high` , `medium` , `low`) with a configurable finding count, license type (allow/deny by full name, including handling for undetermined licenses), and package age (minimum age before a version may enter a build). Available in early access for `GitLab.com` and GitLab Self-Managed customers on Premium or Ultimate tiers.



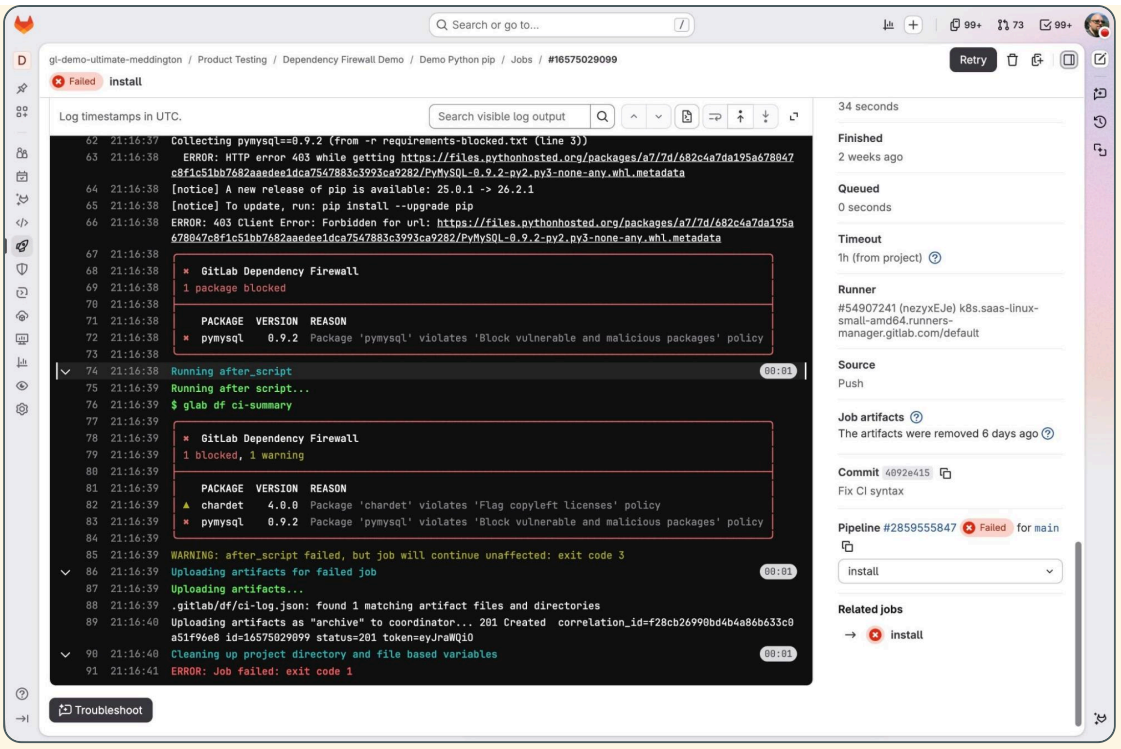
— Names modes and policy values but no exact config syntax

`Dependency Firewall: Block risky packag…`

03 CLI pre-flight check for Dependency Firewall policy NEW

70

A new GitLab CLI command checks whether a package will pass or be blocked by Dependency Firewall policy before it is added as a dependency, covering npm, pip, Poetry, Maven, Gradle, and Bundler.



— Names package managers covered but not the exact command syntax

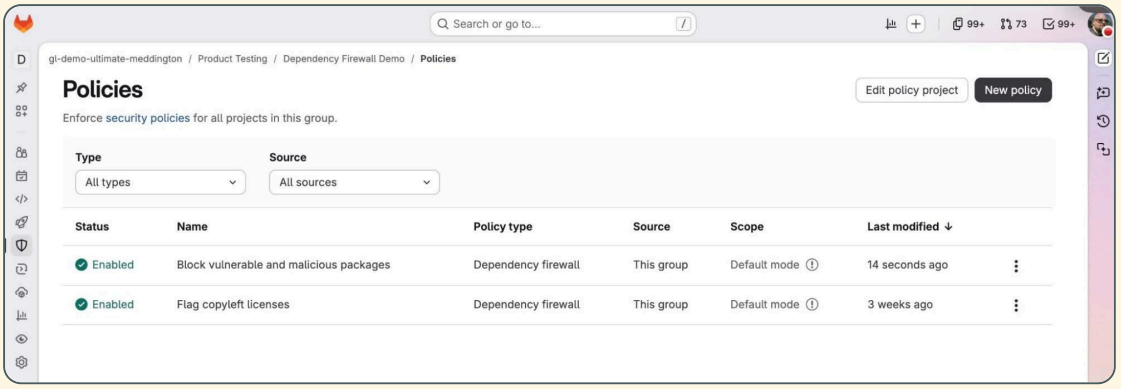
Dependency Firewall: Block risky packag…

---

04 Policy-as-code with group inheritance NEW

65

Dependency Firewall rules are stored as code in a security policy project, reviewed and merged like any configuration change, with inheritance from top-level group down to each project and strictest-rule precedence applied when rules overlap.



— Explains mechanism and precedence, no file schema shown

Dependency Firewall: Block risky packag…

---

05 CI_JOB_TOKEN authentication for Artifact Central NEW

65

Artifact Central uses `CI_JOB_TOKEN` as the authentication mechanism for publishing and pulling artifacts, eliminating the need for separate service accounts.

— Names the exact token but gives no usage example

Every artifact your teams ship, assembl…

06 **Artifact Central organization-level registry** NEW 60

GitLab Artifact Central (beta) is a new organization-level registry replacing per-project package and container registries, with centralized retention rules, storage quotas, and access policies. It supports Maven, npm, Docker, and OCI artifact formats during the beta period.

— Describes scope and formats but no setup command

Every artifact your teams ship, assembl…

THINNER COVERAGE BELOW

07 **Dependency Firewall activity dashboard** NEW 50

A new Dependency Firewall dashboard shows allow, warn, and block activity and outcomes across all projects in scope, backed by an immutable audit event recorded for every decision.

— Describes dashboard content but no navigation path given

Dependency Firewall: Block risky packag…

08 **Dedicated artifact roles** NEW 50

Artifact Central introduces four dedicated artifact roles — Admin, Manager, Contributor, and Viewer — kept separate from existing GitLab project roles so artifact access never affects project privileges.

— Names roles but no assignment steps Every artifact your teams ship, assembl…

09 **Automatic build provenance** NEW 50

Artifact Central automatically attaches build provenance — pipeline, branch, commit, and triggering user — to every artifact published through it.

— Describes provenance fields but no verification steps

Every artifact your teams ship, assembl…

10 **Logged bypass for Dependency Firewall** NEW 45

Dependency Firewall adds a logged bypass mechanism that lets a designated user or token push an approved package through the firewall, with the override recorded on the audit trail.

— Mechanism named but no exact flag or workflow given

Dependency Firewall: Block risky packag…

11 **Registry sources for Dependency Firewall** NEW 45

Dependency Firewall supports GitLab Artifact Central and external registries from Sonatype Nexus Repository and JFrog Artifactory without requiring a separate tool deployment.

— Names supported registries but no setup steps

Dependency Firewall: Block risky packag…

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

02 Zenity

announced · Oct 3, 2026

AI/LLM Security

Zenity discovers, secures and governs enterprise AI agents from buildtime to runtime across SaaS, endpoint and cloud.

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 5/5 NOVELTY 5/5

AI Total detonates an AI agent skill inside a live sandboxed agent seeded with bait credentials and sensitive files, recording every domain contacted, package pulled, file touched, and command run to expose runtime behavior static analysis misses — free, before you install.

> Zenity launched AI Total, a free sandbox service that detonates AI agent skills before deployment to expose their real runtime behavior.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 AI Total sandbox detonation service NEW

80

AI Total, a free service at `aitotal.io`, runs AI agent skills inside a live sandboxed agent and records every domain contacted, package pulled, file touched, command run, and tool call, revealing runtime behavior hidden from static analysis. The sandbox is seeded with planted credentials and sensitive files (e.g. `~/.aws/credentials`) as bait so skills that hunt for secrets or phone home to unknown domains expose themselves during detonation rather than in production. It supports skills used with Claude Code, OpenClaw, and other agent platforms that load skills on demand.

Before adding a public skill from a registry to your Claude Code or OpenClaw agent, submit it to AI Total to see exactly what it does at runtime — domains it contacts, files it touches, and commands it runs.

📍 Visit `https://aitotal.io`, submit the skill package URL or file, and review the behavioral evidence report before installing.

— Names mechanism, bait technique and platforms; gives URL but no API.

launch-20261003-59f3ed6f

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

03 SonarQube

announced · Oct 1, 2026

SAST / SCA / Code Security

Trusted by 7M+ developers, SonarQube is the AI code review and code verification platform that catches quality and security issues before they reach production.

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 4/5 NOVELTY 4/5

The Remediation Agent now runs on self-hosted SonarQube Server, auto-generating fixes that are re-analyzed in a sandbox and submitted as pull requests without code leaving operator-controlled infrastructure, covering C#, Java, JS/TS, and Python including leaked secrets and logic flaws. It supports air-gapped and VPC-restricted environments with bring-your-own-LLM endpoints.

> SonarQube's 2026.5 LTA release and the Remediation Agent launch bring a wave of self-hosted AI agents — the Remediation Agent, Sonar Vortex, and Hunter Agent — plus an embedded MCP server for agentic coding tools, deeper reachability and taint analysis, new language support including Rust and MuleSoft DataWeave, and expanded compliance reporting.

└──▷ WHAT SHIPPED · 16 FEATURES most completely described first ? what's the number?

01 Self-hosted Remediation Agent for automated PRs NEW

96

The Remediation Agent (SonarQube Server Enterprise and Data Center) auto-generates verified pull requests for C#, Java, JavaScript/TypeScript, and Python issues — including maintainability, reliability, and select security issues like leaked secrets and logic flaws — without code leaving self-hosted infrastructure, including air-gapped and VPC-restricted environments. It supports bring-your-own-LLM endpoints (AWS Bedrock, Azure AI Foundry, native Anthropic and OpenAI keys, custom gateways, GPT-5.5, Claude Opus 4.6) across GitHub, GitLab and Azure DevOps, verifies every fix by re-running Sonar analysis in a sandbox before opening a PR — discarding fixes that don't clear the issue or introduce new ones — and can run on a schedule (daily or weekly, per org or project, with a configurable PR cap) or on-demand by filtering the Issues page for agent-fixable items. An Agent activity page lists every running and finished job with status, duration, source and opened pull requests.

Enable the Remediation Agent on a self-hosted SonarQube Server instance connected to Azure AI Foundry for regulated environments that cannot use external AI endpoints.

📍 In the SonarQube Server administration UI, go to AI Capabilities, connect your DevOps platform (GitHub, GitLab, or Azure DevOps) with the required permissions, select Azure AI Foundry as your model endpoint, and enable the Remediation Agent.

Schedule weekly automated remediation at the project level to keep technical debt low without manual triage, capping PR volume to avoid overwhelming reviewers.

> 📍 In the SonarQube Server project settings under the Remediation Agent section, set the schedule to weekly and configure the maximum number of pull requests per run, then save.

Trigger on-demand remediation before an audit by filtering the Issues page to agent-fixable items in a single project and assigning them to the agent for immediate PR generation.

> 📍 On the SonarQube Server Issues page, filter by issues the Remediation Agent can fix, select the target issues, and assign them to the agent to generate and submit pull requests grouped by rule and file type.

— Named endpoints, flags and UI steps given for setup and use. `launch-20261001-95eff5a5` `launch-20260929-8ac52e0b`

---

02 **Rust analysis beyond Clippy** NEW — 73

Rust analysis now goes beyond Clippy, layering centralized quality profiles, quality-gate enforcement, Cognitive Complexity and Cyclomatic Complexity metrics, code-coverage correlation from `LCOV` or `Cobertura`, duplication detection, and coverage of popular Rust crates.

— Mechanism and concrete formats named, but no invocation shown. `launch-20260929-8ac52e0b`

---

03 **SonarQube Hunter Agent for identity and data-flow findings** NEW — 65

`SonarQube Hunter Agent` is a self-hosted deployment that uses your own Anthropic key to trace data and identity flows across a full codebase, detecting IDOR, privilege escalation, and session fixation, and delivers confirmed findings with severity directly into the existing SonarQube issue list.

— Clear mechanism and findings named but no setup steps. `launch-20260929-8ac52e0b`

---

04 **MISRA, WCAG, and CRA compliance reporting** NEW — 63

MISRA C automated coverage is raised to 38.4% for MISRA C:2012, and a new MISRA C/C++ Compliance Report generates audit-ready, tamper-evident attestation artifacts automatically. New WCAG Accessibility and CRA Compliance Reports prove conformance at the project, application, and portfolio level.

— Gives a concrete coverage number and names each report. `launch-20260929-8ac52e0b`

---

05 **Embedded SonarQube MCP Server** NEW — 60

An embedded `SonarQube MCP Server` endpoint lets Cursor, Claude Code, GitHub Copilot, and other agentic tools reach project issues and quality gates, governed by a

token-based kill-switch.

— Named surface and integrations but no path or usage shown. launch-20260929-8ac52e0b

06 **New reporting dashboards** NEW 60

Customizable portfolio and project dashboards add a new overview landing page for organization-wide reporting, a project coverage dashboard showing which DevOps platforms are connected, which repositories are imported, and which projects have not yet been analyzed, and a quality gate adherence dashboard flagging risky releases that shipped despite a failing gate.

— Names three distinct dashboards and what each shows. launch-20260929-8ac52e0b

THINNER COVERAGE BELOW

07 **Language and framework coverage expansion** NEW 55

Adds native support for `MuleSoft DataWeave` and `R` as new analysis languages, and expands language coverage to include `Java 25`, Python web frameworks (`FastAPI`, `Flask`, `Django`), `Jenkins Groovy`, `PowerShell`, `Gosu`, `Ruby`, and `Apex`.

— Names every added language and framework explicitly. launch-20260929-8ac52e0b

08 **Reachability analysis for Java and C#** NEW 50

Direct reachability analysis for Java and C# identifies which dependency vulnerabilities the application code actually calls.

— Names languages and purpose, lacks usage detail. launch-20260929-8ac52e0b

09 **Incremental taint analysis for faster PR scans** IMPROVED 50

Incremental taint analysis cuts pull request scan times on large Java and C# projects by up to 90%.

— Concrete performance figure but no usage detail. launch-20260929-8ac52e0b

10 **Azure API Management for LLM governance** NEW 45

`Azure API Management` support enables LLM governance so Azure OpenAI customers can authenticate to Sonar's AI features without static API keys.

— Names the integration but no configuration steps. launch-20260929-8ac52e0b

11 **Quality gate for agent-generated code** NEW 45

A new quality gate purpose-built for agent-generated code enforces stricter security, reliability, and supply-chain standards while staying permissive on coding style.

— Describes intent but not configuration mechanism. launch-20260929-8ac52e0b

12 **C/C++ taint and cross-file symbolic execution** NEW 45

New taint analysis for C and C++ covers command-injection and path-traversal detection, and cross-translation-unit awareness in C/C++ symbolic execution makes analysis consistent across file boundaries.

– Names detection types but no scan invocation detail. launch-20260929-8ac52e0b

13 **Sonar Vortex agentic coding guardrail** NEW 40

Sonar Vortex is a self-hosted deployment for SonarQube Server Enterprise and Data Center that injects project context and constraints into coding agents and verifies their changes inside the agent loop before commit.

– No setup command or UI path given. launch-20260929-8ac52e0b

14 **Security hotspots merged into unified vulnerability model**

IMPROVED 35

Security hotspots are migrated into a unified vulnerability model, preserving full history.

– Describes change but not migration mechanics or effect on workflows.

launch-20260929-8ac52e0b

15 **Architecture drift detection** NEW 30

Architecture management automatically catches architectural drift and surfaces it in the existing developer issue list.

– Thin prose description, no mechanism or scope detail. launch-20260929-8ac52e0b

16 **Secrets masked and redacted by default** IMPROVED 30

Detected secrets are now masked and redacted by default to reduce potential exposure.

– Bare statement, no scope or configuration detail. launch-20260929-8ac52e0b

GOOD PICK? GOOD REASON?

04 Elastic Security Labs

`Behind the tags: How Elastic SIEM grades 1,781 detection rules on noise, speed, and th`

SIEM

Elastic Security Labs is a research division that publishes threat intelligence, malware analysis, and security research to help defenders detect and respond to cyber threats.

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 5/5 NOVELTY 5/5

Every one of Elastic SIEM's 1,781 prebuilt detection rules now carries monthly-refreshed Noise, Performance, and Threat tags scored from 30-day fleet telemetry, plus a Profile: Recommended/Aggressive label — so teams decide which rules to enable on real evidence rather than guesswork.

Elastic SIEM introduces monthly-recalculated Noise, Performance, Threat, and Profile tags on prebuilt rules, scored from real fleet telemetry via an automated pipeline with human review.

COVERAGE

UPDATED detections

Monthly tag refresh for prebuilt detection rules — Noise, Performance, Threat, and Profile, Standardised prebuilt-rule tag taxonomy — Domain, Platform, Data Source, Service, Vuln, Resources

NEW detection Planned MITRE ATLAS tags for prebuilt rules

All of Elastic Security Labs's coverage changes →

- An automated monthly pipeline scores every prebuilt rule across `Noise`, `Performance`, `Threat`, and `Profile` dimensions from a 30-day telemetry window and opens a draft PR on `elastic/detection-rules` for human review before merging.
- `Profile: Recommended` and `Profile: Aggressive` synthesise severity, noise, performance, and threat coverage into a single enablement-posture label, with hard gates preventing `Noise: High` rules from ever being `Recommended`.
- LLMs are used only as a fallback for `Threat` tag classification where heuristics find no match, capped at five tags per rule and filtered against a human-maintained catalog; manually curated tags (e.g. `Threat: Cobalt Strike`) are never overwritten by the pipeline.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

05 GitHub

Local sandboxing for GitHub Copilot now generally available (2026-10-07)

SAST / SCA / Code Security

GitHub is a web-based platform for version control and collaborative software development using Git repositories.

// WHY IT MADE THE LIST — DEPTH 5/5 — IMPACT 4/5 — NOVELTY 4/5

GitHub Copilot's local sandboxing (Microsoft eXecution Container) reaches GA, enforcing per-policy filesystem, network, and credential boundaries on agentic tool and command execution with enterprise-managed settings developers cannot weaken; a purpose-built secret-detection model also extends to push protection, catching unstructured credentials before they enter history.

> GitHub's biggest moves this window are on Copilot: computer use for desktop apps, code-defined dynamic workflows across CLI/app/SDK, GA local sandboxing via MXC, local Ollama model discovery, Claude Haiku 5.5, and a purpose-built secret-detection model with AI push protection. Security teams also get a richer security-advisory API (comments, confidential comments, new GraphQL fields), stronger private vulnerability reporting controls (rate limits, structured forms), npm trusted-publishing hardening, wider pull request archiving, and a batch of VS Code Copilot agent features.

└──▷ **WHAT SHIPPED · 23 FEATURES** most completely described first ? what's the number?

01 Structured forms for private vulnerability reports NEW

100

`.github/VULNERABILITY_REPORT.yml` configures a custom structured form for private vulnerability reports (place it in the org/account `.github` repo to apply org-wide); fields support `min_length` (issue-form syntax) to enforce detail minimums. A new REST API endpoint returns the form a repository enforces, surfaced in error responses when an API submission doesn't match it. The default form now requires summary, details, proof of concept (minimum 150 characters), and impact; reporters can check an 'I used AI assistance to find or write up this report' disclosure box; organizations/enterprises can require a CWE before submission via Settings > Advanced Security > Private vulnerability reporting; and repos with a `SECURITY.md` show a banner linking to it before submission.

Enforce a structured vulnerability report form with minimum detail requirements across all repositories in your organization.

```yaml
# .github/VULNERABILITY_REPORT.yml in your org's .github repo
name: Vulnerability Report
description: Report a security vulnerability
body:
  - type: textarea
    id: summary
    attributes:
      label: Summary
    validations:
      required: true
      min_length: 50
  - type: textarea
    id: details
    attributes:
      label: Details
    validations:
      required: true
      min_length: 100
  - type: textarea
    id: poc
    attributes:
      label: Proof of Concept
    validations:
      required: true
      min_length: 150
  - type: textarea
    id: impact
    attributes:
      label: Impact
    validations:
      required: true
```

Require reporters to assign a CWE before submitting, and enforce the setting across your organization via policy.

> 📍 In the repository, go to Settings › Advanced Security › Private vulnerability reporting and enable the option to require a CWE. Organization and enterprise owners can then enforce this setting via a policy applied across all repositories.

— Exact config file, field, API and settings path given

Structured forms for private vulnerabil…

---

02 **Computer use for desktop apps in Copilot** NEW 100

Computer Use (public preview) lets GitHub Copilot interact with desktop apps on macOS and Windows — clicking controls, entering text, pressing keys, scrolling, dragging, and navigating workflows, including legacy and GUI-only software with no API, CLI, or MCP integration. Enable via `/computer on` in Copilot CLI, check status with `/computer show`, disable with `/computer off`, or toggle it in the GitHub Copilot app at Settings > Computer Use > Enable Computer Use, which also offers per-app approval controls and review/reset of always-allowed apps. On macOS it guides users through required Accessibility and Screen Recording permissions, and organizations can disable it across managed devices.

▶ Watch video

Enable computer use in Copilot CLI so Copilot can automate a GUI-only desktop workflow — for example, moving data through a legacy expense-report app with no API.

```
$ /computer on
```

Check whether computer use is active in the current Copilot CLI session, or turn it off after finishing an automated workflow.

```
$ /computer show
/computer off
```

Enable Computer Use from the GitHub Copilot app UI on macOS or Windows when you prefer not to use the CLI.

📍 Open the GitHub Copilot app › Settings › Computer Use › turn on Enable Computer Use. On macOS, follow the prompted steps to grant Accessibility and Screen Recording permissions.

— Runnable commands and exact settings path given

GitHub Copilot can now interact with de…

03 **Dynamic workflows in Copilot CLI, app, and SDK** NEW

100

Dynamic workflows enable code-defined, multi-agent orchestration: in Copilot CLI via `--experimental` or `/experimental on` in an interactive session; in the GitHub Copilot app with no setup required, supporting parallel agent execution, checkpoints, and structured result passing; and in the Copilot SDK, giving extension authors programmatic access to Copilot's extensibility APIs. Workflows support pause-and-resume checkpoints to review intermediate findings before continuing, parallel sub-agent execution (e.g. reviewing many PR files or sweeping a large codebase across directories

simultaneously), and built-in authoring guidance so Copilot can write a workflow for you or answer 'What dynamic workflows are available?'

Enable dynamic workflows in Copilot CLI to orchestrate a multi-agent incident investigation across logs and telemetry.

```
$ gh copilot --experimental
```

Turn on experimental features mid-session and discover existing dynamic workflows without restarting the CLI.

```
$ /experimental on
What dynamic workflows are available?
```

Have Copilot author a reusable release-check workflow that pauses for human review before resuming.

> 📍 Ask Copilot: 'Create a dynamic workflow that runs release checks, asks one agent to assess failures, and pauses at a checkpoint for me to review before resuming.'

— Exact flags, commands, and mechanism given `Dynamic workflows in Copilot CLI and th…`

---

04 **Discover local Ollama models in Copilot CLI** NEW 95

`/model` command (Copilot CLI v1.0.94-0+) discovers models from a running local Ollama instance alongside configured and cloud models, letting you add a discovered model for the session via `Add and use for this session` or `Add without switching` without restarting the CLI. The picker surfaces provider connection failures with explanations, offline mode can be forced independently via `COPILOT_OFFLINE=true`, and local model providers can be configured in the GitHub Copilot app at Settings > Model providers.

Discover and add a local Ollama model to your Copilot CLI session without restarting — useful when you want to test a locally hosted model mid-workflow.

```
$ /model
```

Add a supported local model provider in the GitHub Copilot app so it appears in the model picker alongside cloud models.

> 📍 In the GitHub Copilot app, go to Settings › Model providers and add your local Ollama provider.

— Runnable command plus exact env var and settings path

Discover local models in GitHub Copilot…

05 Repository security advisory comments REST API NEW

95

Adds REST endpoints to list comments on a repository security advisory (with optional filtering by last-updated time), get a single comment, `POST` a new comment, and `PATCH` to edit a comment. Repository security advisory responses now include a `comments_count` field, and global advisory responses include the `comments_count` from their linked repository advisory.

Automatically post a triage note to a security advisory when your pipeline finishes analysis — keeping advisory discussion out of the web UI and in your toolchain.

```
$ curl -X POST \
  -H 'Authorization: Bearer <token>' \
  -H 'Accept: application/vnd.github+json' \
  https://api.github.com/repos/<owner>/<repo>/security-advisories/<ghsa_id>/comments \
  -d '{"body": "Triage complete: exploitability confirmed on v2.3.1. Patch queued for next release."}'
```

Check which open advisories already have discussion before pulling full comment threads — useful when triaging a backlog.

```
$ curl -H 'Authorization: Bearer <token>' \
  -H 'Accept: application/vnd.github+json' \
  'https://api.github.com/repos/<owner>/<repo>/security-advisories' \
  | jq '[.[] | select(.comments_count > 0) | {ghsa_id, summary, comments_count}]'
```

Export all advisory comments updated since a given date for an audit trail or migration to another system.

```
$ curl -H 'Authorization: Bearer <token>' \
  -H 'Accept: application/vnd.github+json' \
  'https://api.github.com/repos/<owner>/<repo>/security-advisories/<ghsa_id>/comments?since=2026-01-01T00:00:00Z'
```

— Named endpoints with runnable curl examples Repository security advisory comments A…

06 Actions retention covers checks, runs, and statuses

BREAKING

95

The existing Actions retention setting (Check, workflow run, status, artifact and log retention in repo Settings > Actions > General) now also cleans up checks, workflow runs, and statuses created by GitHub Actions and third-party applications, not just artifacts and logs. Public repository retention for checks, runs, and statuses is capped at a maximum of 90 days; data older than the configured period is automatically deleted and cannot be restored.

Review or update your repository's unified retention period so checks, runs, and statuses are kept as long as your audit or compliance workflow requires.

> 📍 In the repository, go to Settings › Actions › General, then locate 'Check, workflow run, status, artifact and log retention' and set the desired number of days (max 90 for public repositories).

— Exact setting label, path, and 90-day cap given

Actions retention now covers checks, ru…

---

07 Purpose-built secret-detection model and AI push protection

NEW

90

Adds AI-detected secret scanning to the `/security-review` command in Copilot CLI and Copilot App sessions, and adds AI push protection checking for unstructured credentials before they enter history (private preview, GitHub Enterprise Cloud and GitHub Teams with GHSP or GHAS). Existing AI-detected Password alert scans are automatically upgraded to the new fine-tuned model at no extra charge, AI-detected secret alerts come to GitHub Enterprise Server 3.23 in public preview, and a dedicated Secret Protection AI Credits SKU appears under AI usage insights with opt-in budget controls at `Billing and licensing > Budgets and alerts` (SKU-level budget, Product: Advanced Security , SKU: Secret Protection AI Credits).

Run an AI-powered security review of active changes — including the new secret-classifier checks — before committing or opening a pull request.

```
$ /security-review
```

Cap spending on AI push protection and secret-classifier checks before opting into the public preview.

> 📍 In the GitHub UI: Billing and licensing › Budgets and alerts › New budget › SKU-level budget › Product: Advanced Security › SKU: Secret Protection AI Credits › Enable 'Stop usage when budget limit is reached'.

— Names command, SKU, settings path and preview scope

Purpose-built model for leaked secret d…

08 AI Scan enablement status in security overview NEW

85

Security overview coverage view adds `code-scanning-ai-scan-pr-scan:enabled` and `code-scanning-ai-scan-pr-scan:not-enabled` filter qualifiers to isolate repositories by AI Scan for pull requests status, a `Code Scanning AI Scan for pull requests` column (values `enabled` / `not-enabled`) in the coverage CSV export, and per-repository rows plus summary counts of enabled/not-enabled repos in the coverage view.

.

Filter the security overview coverage view to find all repositories that have NOT yet enabled AI Scan for pull requests, so you can target rollout gaps across your org.

> 📍 In the GitHub security overview coverage view, enter `code-scanning-ai-scan-pr-scan:not-enabled` in the filter bar to list all repositories where AI Scan for pull requests is not enabled.

— Names exact filter qualifiers and export column

Code scanning AI Scan enablement status…

09 npm trusted publishing configurations expire unvalidated

85

BREAKING

Unvalidated npm trusted publishing configurations now expire 48 hours after creation and can no longer authorize publishing until a successful publish validates them; changing the repository or project identity resets the trust relationship with a fresh 48-hour window, while ordinary edits do not restart the deadline. npm also rejects trusted publishing tokens from GitHub Actions `issue_comment` events — workflows must move publishing to `push`, `release`, or `workflow_dispatch`. Expired configurations remain visible in trusted publisher settings but don't count toward per-package limits.

— Exact windows and events named, migration path implied

Unvalidated npm trusted publishing conf…

10 npm staged publishing creates new packages NEW

85

`npm stage publish` can now create brand-new npm packages (public scoped, unscoped, and private scoped) without a manual first publish, using a local session or a granular access token including a stage-only token. First-version publishes enter a queue requiring maintainer promotion before the version becomes installable, and package settings/trusted publishing can be configured immediately after creation.

Bootstrap a brand-new npm package from CI without a manual first publish — the version enters the staged queue awaiting maintainer promotion.

```
$ npm stage publish
```

— Runnable command with clear before/after workflow

npm staged publishing now supports crea…

11 **Confidential comments on security advisories** NEW 85

Security advisories support confidential comments, visible only via GraphQL (not returned by REST) to users with repository write access; they're clearly marked in the advisory timeline, views are logged in the audit log, and access follows write permissions — losing write access blocks reading existing confidential comments. Post one by selecting Confidential below the comment box before submitting; the setting can't be changed afterward.

Post a confidential internal note on an advisory — e.g. to record investigation details without exposing them to the reporter or invited collaborators.

📍 In the security advisory, type your comment, select Confidential below the comment box, then click Submit. The comment will appear with a confidential marker visible only to maintainers with write access.

— Full mechanism and exact UI steps given Confidential comments on repository sec…

12 **Rate limits for private vulnerability reports** NEW 85

Daily per-user rate limits now apply to new private vulnerability reports (reporters who hit the limit see a try again later message; comments on existing advisories are unaffected). Repository admins can set a custom daily overall reporting limit and add trusted reporters to an allow list so they're never rate limited, both configurable under Settings > Advanced Security > Private vulnerability reporting.

Add a trusted security researcher to the allow list so their reports are never blocked by rate limits.

📍 1. Go to your repository on GitHub. 2. Click Settings › Advanced Security. 3. Click Settings next to Private vulnerability reporting. 4. Under the allow list section, add the trusted reporter's GitHub account. 5. Save changes.

Set a custom daily reporting cap for your repository to throttle bulk submissions while still accepting legitimate reports.

📍 1. Go to your repository on GitHub. 2. Click Settings › Advanced Security. 3. Click Settings next to Private vulnerability reporting. 4. Enter your desired daily overall reporting limit. 5. Save changes.

— Exact settings path and allow-list mechanism given

Rate limits for private vulnerability r…

13 **Pull request archiving opened to triage role, made fully read-only**

75

BREAKING

Extends pull request archive and unarchive permissions from admin-only to users with the `triage`, `write`, `maintain`, or `admin` role, and makes archived PRs fully read-only — blocking new comments, reactions, and automated comments for everyone, including admins (previously admins could still comment on a locked/archived PR). Archived pull requests are hidden from public view but remain visible to repository administrators.

Triagers handling spam or abandoned PRs can now archive them directly in the UI without escalating to an admin.

> 📍 In the pull request page, open the pull request actions menu and select Archive pull request. The PR will be closed and its conversation made read-only immediately.

— Clear role and behavior change, UI-only action path

Triage role users or higher can now arc…

14 **Local sandboxing for Copilot (MXC) reaches GA** IMPROVED

75

Local sandboxing for GitHub Copilot (powered by Microsoft eXecution Container, MXC) is now generally available in Copilot CLI, the GitHub Copilot app, and VS Code Agent Host sessions, restricting agentic tool and command access to the filesystem, network, and credentials. Per-policy controls govern which files/directories agent commands can read or modify and access to the internet, local networks, Git credentials, and GitHub CLI credentials; local MCP and language servers can be sandboxed where supported, and enterprise-managed settings can require sandboxing that developers cannot weaken.

— Describes mechanism and scope but no exact command shown

Local sandboxing for GitHub Copilot now…

15 **New fields and filters on SecurityAdvisory GraphQL object** NEW

75

The `securityAdvisories` GraphQL query gains `severities` and `isWithdrawn` filters for server-side narrowing, and the `SecurityAdvisory` object gains four new fields: `sourceCodeLocation` (links to affected source code), `githubReviewedAt` (when GitHub reviewed the advisory), `nvdPublishedAt` (when NVD published its record), and `repositoryAdvisoryUrl` (link to the associated repository security advisory).

– All field and filter names given, no query example

New fields for SecurityAdvisory GraphQL…

16 **Default Copilot code review effort level changed to Balanced** 70

BREAKING

The default Copilot code review effort level changes from `Lite` to `Balanced` for new and existing repositories and organizations (effective September 28, 2026); per-level UI controls let you set `Lite` or `Balanced` at the enterprise, organization, repository, or personal scope via settings pages.

Override the default effort level back to Lite for a specific repository whose reviewers prefer lighter feedback.

> 📍 In your repository settings, go to Copilot -> Code review and set the review effort level to `Lite`.

– Clear before/after default with settings path Copilot code review: API support and ne…

17 **Scheduled agent automations in VS Code** NEW 70

Copilot in VS Code adds automations (preview) to schedule recurring agent tasks on an hourly, daily, or weekly cadence, or run them on demand from a template or custom prompt.

Run a nightly dependency-audit prompt against your repo without manual intervention.

> 📍 In the Agents window, select `New Automation`, choose a daily schedule, and enter your prompt (or pick a template) to have Copilot run the task automatically each day.

– Clear cadence options and starting point GitHub Copilot in VS Code, September 20…

18 **Agent merge in Copilot sessions** NEW 70

Agent merge (preview) in active Copilot sessions lets the agent handle review feedback, failed checks, merge conflicts, and workflow reruns, enabled via `Enable agent merge` in the Agents window.

Hand off post-PR-creation chores to the agent so you can focus on the next task instead of babysitting CI.

> 📍 In the Agents window, open an active agent session, then select `Enable agent merge` to let the agent automatically respond to review feedback, resolve merge conflicts, rerun failed checks, and merge the pull request when ready.

– Names scope of automation and exact toggle GitHub Copilot in VS Code, September 20…

19 **Dev Container option for agent sessions** NEW 70

The Use Dev Container option in the Agents window starts agent sessions inside a project's Dev Container on local, SSH, Tunnel, and WSL hosts.

Run an agent session inside your project's Dev Container so it has access to the exact toolchain and dependencies defined in your devcontainer config.

> 📍 In the Agents window, open the menu for your local or remote folder and select Use Dev Container to launch the agent session inside the configured container (works on SSH, Tunnel, and WSL hosts).

— Names exact option and supported hosts GitHub Copilot in VS Code, September 20…

20 **Claude Haiku 5.5 model in Copilot** NEW 65

Adds Claude Haiku 5.5 as a selectable model in GitHub Copilot's model picker across Visual Studio Code, Visual Studio, Copilot CLI, GitHub Copilot cloud agent, GitHub Copilot app, github.com, GitHub Mobile (iOS and Android), JetBrains IDEs, Xcode, and Eclipse. Copilot Enterprise and Business admins can manage access via the model policy in Copilot settings, with automatic enablement under the default model enablement policy unless explicitly disabled.

Select Claude Haiku 5.5 for fast, token-efficient coding tasks like quick edits or terminal work in VS Code.

> 📍 In GitHub Copilot, open the model picker and select Claude Haiku 5.5 from the available models list.

— Names surfaces but no deeper mechanism Claude Haiku 5.5 in GitHub Copilot (202…

THINNER COVERAGE BELOW

21 **Pull request creation from agent sessions** NEW 55

Pull requests can now be created directly from Copilot, Claude, or Codex agent sessions in the Agents window, with title/description editing and draft and merge options.

— Describes options but no exact navigation steps given

GitHub Copilot in VS Code, September 20…

22 **VS Code Copilot session and UI improvements** IMPROVED 55

Also adds: `HydraFusion` model-picker option (research preview) to automatically select models and workflows for coding tasks; Mark as Done suggestions with automatic cleanup to keep completed sessions organized (preview); application badge support (preview) surfacing new results, input requests, and PR checks on the dock, launcher, or taskbar; a hierarchical sessions list view letting agents spawn new chats/sessions with a navigation link back to the source chat; an Add Context entry-

point and URL-paste support in the new-session input to attach a GitHub issue or pull request to any chat; continuing a general Copilot quick chat (started without a workspace) by attaching a local folder mid-conversation; and picking up a Codex conversation from the ChatGPT app inside VS Code to continue work without copying context or files.

— Names each surface but mostly thin UI additions

GitHub Copilot in VS Code, September 20…

23 **Copilot code review via REST and GraphQL APIs** NEW 50

GitHub Copilot code review can now be requested via REST and GraphQL APIs, with an optional review effort level parameter per request.

— Confirms API access but no endpoint path given Copilot code review: API support and ne…

└──▷ ALSO FROM THESE RELEASES

▶ Watch video

▶ Watch video

▶ Watch video

└──▷ BREAKING ON UPGRADE

! Archiving a pull request now blocks all new activity (comments, reactions, automated comments) for every role including admins — previously admins could still comment on a locked/archived PR.

! The default review effort level changed from Lite to Balanced on September 28, 2026 — repositories and organizations that relied on Lite as the default will now receive Balanced reviews unless Lite was explicitly selected in settings.

! Trusted publishing tokens issued from GitHub Actions `issue_comment` events are now rejected by npm; workflows that publish on `issue_comment` must be moved to a permitted event (`push` , `release` , or `workflow_dispatch`).

! Unvalidated trusted publishing configurations older than 48 hours can no longer authorize publishing; affected configurations must be recreated to start a new 48-hour window.

! If you add a custom `VULNERABILITY_REPORT.yml` form, reports submitted through the REST API must match it — API submissions that do not match will receive an error.

! Checks, workflow runs, and statuses older than the configured retention period are now automatically deleted — including those created by third-party applications. Data already removed before this change cannot be restored.

Secret scanning adds detectors for Lovable, Supabase, and more (2026-10-05)

COVERAGE

NEW detections

Lovable Labs secret scanning partner and detector, Pydantic Services Inc. secret detectors, Supabase secret detectors

All of GitHub's coverage changes →

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

// END OF TRANSMISSION

The Cyber Toolchain · This Week's Highlights · October 9, 2026
The full firehose — every feature release, every day — is in the newsletter.

▷ Subscribe