THE CYBER TOOLCHAIN

NO. 141

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED NOVEMBER 16, 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

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every tool in the issue, in full.

all 65 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 24 tools

▾ // THE BRIEF

HIDE SAME ISSUE, SAME PROMPT, TWO WRITERS: Opus 5 | GPT-5.5

What stands out across today's releases, grouped by what it lets you do. Every tool named links to its entry below.

Three unrelated things stand out. Socket now traces CVEs to actual call chains across six ecosystems and ships a `socket fix` command to

vim ? help

close them. StepSecurity surfaces which environment secrets exist, who uses them and when they were rotated. Horizon3 NodeZero gives MSSPs one console across client tenants and licenses instead of logging in per customer.

PROTECT

Cut a dependency CVE list down to the ones your code can actually reach

Socket's reachability analysis now spans Ruby/Bundler, .NET/NuGet, Go, JVM, Python and JavaScript, so an advisory in a transitive package only lands on the backlog if a call chain gets to it — and `socket fix` plus SCA Patches attempt the upgrade rather than just filing it. Veracode's Reporting API can now filter for ageless SCA findings, the long-tail items that no longer age into an SLA bucket and quietly disappear from reports.

Socket · Veracode

IDENTIFY

See which environment secrets exist, who consumes them, and when they were last rotated

GitHub environment secrets are normally write-only once set, so teams track rotation in a spreadsheet and guess which workflows still consume a given credential. A permission-check endpoint plus an Environment Secrets tab with per-environment rotation and usage tracking turns that into something you can query before deciding whether a leaked token matters.

StepSecurity

GOVERN

Run one console across client tenants instead of logging in per customer

NodeZero's Client and License Management console gives MSSPs a single place to see seat allocation and client state, which is the difference between pentest capacity being manageable at twenty accounts and being a billing problem. Okta extends agent identity governance to third-party IdPs via an open protocol plus agent discovery, so service-account-shaped agents created outside your primary directory stop being invisible.

Horizon3 NodeZero · Okta for AI Agents

DETECT

Query threat intel, alerts and cases conversationally during triage

> An MCP server exposes TI, alerts, cases and SOL data to assistants like Claude Code, so an analyst pivoting from an IOC to related cases does it without hand-writing SOL queries. The risk is the usual one — the assistant gets whatever read scope you grant it — but the pivot cost during live triage drops sharply.
>
> Sekoia.io

Does **Opus 5** read better? 👍 👎

OFFENSIVE SECURITY

◆ Exploitation & C2

NetSPI OCInferno

Feedback ⓘ changes since 2026-11-13 CODE ↗

OCInferno adds `--no-group-hop`, `--no-default-admin-group-hop`, and `--threads` flags, and now models dynamic-group rule modification as an attack path edge

└──▷ TRY IT

Enumerate attack paths while skipping hops through groups, to surface only direct principal-to-resource escalation routes.

```
$ ocinferno process_og_attack_paths --no-group-hop
```

Filter out default-admin-group hops when reviewing privilege-escalation paths, narrowing results to non-admin-group pivots.

```
$ ocinferno process_og_attack_paths --no-default-admin-group-hop
```

Filter attack paths that hop through any group node with `--no-group-hop` on `process_og_attack_paths`.

Filter attack paths that hop through default admin groups with `--no-default-admin-group-hop` on `process_og_attack_paths`.

Parallelize IDD credential scans with `--threads <N>`; defaults to `1` (sequential) so parallelism is opt-in.

OCInferno now models a principal's ability to rewrite dynamic-group matching rules as the attack-path edges `OCI_UPDATE_DYNAMIC_GROUP_RULES` and `OCI_CAN_MODIFY_DG_RULES`, surfacing dynamic-group injection as a privilege-escalation path.

Pathfinder now loads all graph edges by default without requiring `--include-containment`.

+1 more

└──▷ BREAKING ON UPGRADE

! The `--include-containment` flag has been removed; containment edges are now always loaded by the pathfinder.

! The `--exclude-service-agents` flag has been removed as it was not applicable to OCI.

WAS THIS USEFUL? 👍 👎

◆ Hacking Hardware

JustCallMeKoko LLC ESP32 Marauder v8 Feedback ⓘ

back in stock · Nov 15, 2026 STORE ↗

ESP32 Marauder v8 is the official handheld ESP32-C5 board for the ESP32 Marauder firmware, with a touchscreen, dual-band Wi-Fi 6, Bluetooth 5 and GPS for Wi-Fi and BLE auditing and wardriving.

Marauder v8 (marauder-v8) back in stock at justcallmekokollc.com on 15 Nov 2026.

WAS THIS USEFUL? 👍 👎

APPLICATION SECURITY

◆ Secrets Detection & Management

passdetective

Feedback ⓘ 1.1.0 NOTES ↗ RANK

PassDetective is a command-line tool that scans shell command history to detect mistakenly written passwords, API keys, and secrets. Using regular expressions, it helps prevent accidental exposure of sensitive information in your command history.

> PassDetective 1.1.0 adds support for scanning fish shell history and broadens its regex-based detection of secret patterns.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Fish shell history scanning** NEW 35

PassDetective can now scan fish shell command history for accidentally typed passwords, API keys, and secrets, in addition to previously supported shells.

– Named shell support but no mechanism or command details 1.1.0

02 **Expanded secret detection regexes** IMPROVED 25

PassDetective now detects a wider range of secret patterns through expanded secret regexes, improving its ability to catch mistakenly written passwords and API keys in command history.

– States improvement but no specific patterns or regex examples given 1.1.0

WAS THIS USEFUL? 👍 👎

◆ Code Security

Mend.io Mend Platform

Feedback ⓘ

Version 26.10.2 (15-November-2026) NOTES ↗ RANK

Mend Platform is a traditional application security misses AI risk. Standalone AI security tools miss your code.

> Mend Platform's November release focuses on AI-assisted workflows, adding a Claude Connector for its MCP Server and a guided onboarding wizard for SAST scanning, while removing the Forge tab.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 **Guided SAST onboarding wizard** NEW 68

A new guided onboarding wizard activates SAST scanning for repositories directly from the Mend Platform, covering quick repository selection, pilot scan execution, and team onboarding via email invitations. To use it, go to SAST, click `Activate SAST`, select a repository, run a pilot scan, then use `Invite via email` to delegate onboarding to a colleague who owns the repository.

Activate SAST scanning for a repository and invite a colleague to complete the setup when you do not own the repository.

> 📍 In the Mend Platform, go to SAST, click `Activate SAST`, select a repository, run a pilot scan, then use `Invite via email` to delegate onboarding to a colleague.

— Describes each wizard step with exact UI navigation Version 26.10.2 (15-November-2026)

02 **Claude Connector for Mend MCP Server** NEW 67

A new Claude Connector enables one-click installation and discovery of the Mend MCP Server within the Claude environment, with permission-gated access to Mend findings and fixes. Installation is done from within Claude by going to Connectors, searching for `Mend`, and clicking Install on the Mend MCP Server connector.

Connect the Mend MCP Server to the Claude assistant without manual configuration to scan repositories and review findings in one place.

> 📍 In the Claude environment, go to Connectors, search for `Mend`, and click Install on the Mend MCP Server connector.

— Clear mechanism and exact UI steps, but no detail on permission scopes

Version 26.10.2 (15-November-2026)

THINNER COVERAGE BELOW

03 **Forge tab removed** BREAKING 10

The Forge tab has been removed from the platform. No further detail on its prior function or migration path is given.

– Only a bare statement of removal, no context or mechanism

Version 26.10.2 (15-November-2026)

└──▷ **BREAKING ON UPGRADE**

! The Forge tab has been removed from the platform.

WAS THIS USEFUL? 👍 👎

Mend.io Mend SCA Feedback ⓘ

Version 26.10.2 (15-November-2026) NOTES ↗ RANK

Mend SCA is a traditional application security misses AI risk. Standalone AI security tools miss your code.

> Mend SCA added a Claude Connector for its Mend MCP Server, letting users install and scan repositories directly from within Claude.

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

01 **Claude Connector for Mend MCP Server** NEW 50

Mend SCA now adds a Claude Connector for the Mend MCP Server, giving one-click installation and discovery within Claude for repository scanning, remediation, and organization-wide security posture reporting without manual configuration.

– Names integration and capabilities but no setup steps or config.

Version 26.10.2 (15-November-2026)

WAS THIS USEFUL? 👍 👎

◆ Software Supply Chain Security

Socket

Feedback ⓘ New alert covering dual use tools (2026-11-13)

BLOG ↗ RANK

Socket fights vulnerabilities and provides visibility, defense-in-depth, and proactive supply chain protection for JavaScript, Python, and Go dependen...

> Socket's biggest addition this window is Full Application Reachability analysis, extended across Ruby/Bundler, .NET/NuGet, Go, JVM, Python, and JavaScript to narrow CVE lists down to exploitable call chains, alongside a new `socket fix` auto-remediation command and SCA Patches for code-level fixes. Socket Firewall also gained webhook/Splunk decision streaming, Redis-backed verdict caching, metadata filtering, Cooldown mode for newly published packages, DNS-override and Registry Mode routing, plus smaller platform updates to policy propagation speed, liveness checks, the Threat Feed API, and PR scan search.

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

01 Full Application Reachability analysis across ecosystems NEW

100

Socket adds Full Application Reachability analysis, which builds a cross-dependency call graph to identify which CVEs have an exploitable call chain from application code to the vulnerable function, across Ruby/Bundler, .NET/NuGet, Go Modules, JVM (Gradle/Maven/Scala/sbt), Python (Poetry/uv), and JavaScript (npm/pnpm/Yarn) projects. Run it with `socket scan create --reach .`, or locally without uploading via `socket scan reach`, which saves output to `.socket.facts.json`. Tune large scans with `--reach-analysis-timeout` (default `10m`), `--reach-analysis-memory-limit` (default `8GB`, 16–32 GB recommended for large projects), and `--reach-concurrency`; skip workspaces with no source via `--reach-continue-on-no-source-files`, and keep the intermediate facts file with `--reach-retain-facts-file`. For builds without a lock file, resolve the dependency graph per build with `--dynamic-sbom-inference` (scoped with `--dynamic-sbom-inference-ecosystems`), or generate manifests independently with `socket manifest dynamic-sbom-inference`, `socket manifest gradle`, `socket manifest maven`, or `socket manifest scala`, or via CycloneDX using `socket cdxgen -t gradle|maven|sbt -o socket-gradle.cdx.json --install-deps --lifecycle build`. Set the org with `--org <org-slug>` or `socket login`, and authenticate headlessly via `SOCKET_SECURITY_API_TOKEN`; run the same analysis in CI via `socketcli --reach --reach-analysis-memory-limit --reach-analysis-timeout`. Source code and dependency graphs are analyzed locally, not uploaded. In the Dashboard, filter results by reachability status or CVE Reachability to compare Reachable, Unreachable, and Potentially reachable counts, view the CLI's

`Noise%` column (`((Unreachable + NoSup + NoReach) / Total Vulns) × 100`), and override unreachable low-impact CVEs to suppress them from future scans.

Compare baseline vs. reachability scan to see how many CVEs are actually exploitable in your Ruby app.

```
$ socket scan create .
# note total CVE count, then:
socket scan create --reach .
```

Run reachability analysis offline on a large monorepo with extended timeout and memory, without uploading results.

```
$ socket scan reach
# results saved to .socket.facts.json
```

Tune reachability analysis for a large Bundler project where the default 10-minute timeout and 8 GB memory limit are insufficient.

```
$ socket scan create --reach \
  --reach-analysis-timeout 20m \
  --reach-analysis-memory-limit 32GB \
  --reach-concurrency 4 \
  .
```

Scan a .NET solution in CI and get a noise-reduced CVE list showing only exploitable paths, with the dependency graph resolved per build.

```
$ socket scan create --reach --dynamic-sbom-inference .
```

Tune a large NuGet monorepo scan that exceeds the default memory and timeout limits.

```
$ socket scan create --reach \
  --reach-analysis-timeout 20m \
  --reach-analysis-memory-limit 32GB \
  --reach-concurrency 4 \
  .
```

Enable lock file generation in a .csproj so Socket can produce the most accurate dependency graph.

```xml
<PropertyGroup>
  <RestorePackagesWithLockFile>true</RestorePackagesWithLockFile>
</PropertyGroup>
```

Establish a baseline CVE count, then run a reachability scan to filter out non-exploitable findings in a Go Modules project.

```
$ socket scan create .
# note baseline CVE count, then:
socket scan create --reach .
```

Analyze a large Go monorepo locally without uploading, with extended timeout and higher memory and concurrency limits.

```
$ socket scan create --reach \
  --reach-analysis-timeout 30m \
  --reach-analysis-memory-limit 32GB \
  --reach-concurrency 4 .
```

Run reachability analysis offline and inspect results before deciding whether to upload.

```
$ socket scan reach
```

Run a full reachability scan on a Gradle or Maven project to identify only the CVEs that are actually reachable from your application code.

```
$ socket scan create --reach --dynamic-sbom-inference .
```

Analyze reachability locally without uploading a scan result — useful for sensitive codebases where you want to inspect `.socket.facts.json` before sharing.

```
$ socket scan reach --dynamic-sbom-inference
```

Tune reachability analysis on a large multi-module project to avoid timeouts and memory exhaustion.

```
$ socket scan create --reach --dynamic-sbom-inference --reach-analysis-timeout 20m --reach-analysis-memory-limit 32GB --reach-concurrency 4 .
```

Identify which CVEs in a Poetry project are actually reachable from your application code, to cut triage noise before a release.

```
$ socket scan create --reach --reach-analysis-memory-limit 32GB --reach-concurrency 4 .
```

Run reachability analysis on a uv project in CI without uploading results, then inspect the output file before promoting to a full scan.

```
$ socket scan reach
```

Authenticate in a headless CI environment using an API key instead of interactive login.

```
$ export SOCKET_SECURITY_API_TOKEN=sktsec_<your_key>
socket scan create --reach --org <your-org-slug> .
```

Compare total CVE count before and after enabling reachability to measure noise reduction in a large npm project.

```
$ npm install
socket scan create .
# note baseline CVE count, then run with reachability:
socket scan create --reach --reach-analysis-memory-limit 32GB --reach-analysis-timeout 20m --reach-concurrency 4 .
```

Run reachability analysis locally on a pnpm workspace without uploading results, then inspect the output file before committing.

```
$ pnpm install
socket scan reach
cat .socket.facts.json
```

Authenticate via API key in CI and run a reachability scan for a Yarn project without an interactive login.

```
$ export SOCKET_SECURITY_API_TOKEN=sktsec_<your_key>
socket scan create --reach --org <your-org-slug> .
```

Scan an npm app with full reachability to collapse `Potentially reachable` CVEs into definitive Reachable/Unreachable verdicts, isolated on its own branch so it does not pollute the Alerts baseline.

```
$ npm ci
socket scan create --org=your-org --repo="your-app" --
branch="socket-reach" --reach --no-interactive .
```

Scan a large app in CI with raised memory and timeout limits to avoid out-of-memory or timeout failures, running two parallel analysis workers.

```
$ socket scan create --org=your-org --repo="your-app" --
branch="socket-reach" --reach --no-interactive \
  --reach-analysis-memory-limit 32GB \
  --reach-analysis-timeout 30m \
  --reach-concurrency 2 .
```

Run reachability in a CI pipeline using socketcli, constrained to the default branch only so pull requests stay on the faster standard scan.

```
$ socketcli --target-path . --reach --reach-analysis-memory-limit
16GB --reach-analysis-timeout 30m
```

— Every flag, default, and command named across all six ecosystems. product docs

02 **SCA Patches apply code-level CVE fixes** NEW 95

Socket adds SCA Patches, which applies code-level CVE fixes without changing package versions or lockfiles. Automate reapplication after every `npm install` or `npm ci` with `socket patch setup --yes`, which adds `"postinstall": "socket patch apply --silent --ecosystems npm"` to `package.json`; query reachable, patchable CVEs across all repos via `GET https://api.socket.dev/v0/orgs/your-org/historical/alerts` with `filters.cvePatchStatus=patch_available` and `filters.alertReachabilityType=reachable`; and add `--offline` to `socket patch apply` to fail rather than fetch a missing patch from `.socket/blobs/`. Socket Patch CLI 5.1.0 adds recursive monorepo scanning via `socket-patch scan` run from the repo root, Go workspace support, and rebuilds gems with native extensions (e.g.

nokogiri) after patching, with hosted mode replacing already-installed gems while keeping Gemfile declarations intact.

Find all reachable, patchable CVEs across your org's repos in the last 30 days to prioritize which packages to patch first.

```
$ curl -s -H "Authorization: Bearer $SOCKET_CLI_API_TOKEN"
"https://api.socket.dev/v0/orgs/your-org/historical/alerts?
range=-30d&per_page=100&filters.cvePatchStatus=patch_available&fi
lters.alertReachabilityType=reachable" | jq -r '[.items[]? | "\
(.repoSlug) \(.artifact.name)@\(.artifact.version) \
(.alert.props.cveId)"] | unique[]'
```

Scan a project for available patches and apply one by name without changing the manifest or lockfile.

```
$ cd /path/to/your-app && npm ci && socket patch scan --json >
patches.json && socket patch get minimist && npm test
```

Gate CI so it fails if patches were not applied — catching cases where the installed version no longer matches a saved patch.

```
$ out=$(socket patch apply --json)
echo "$out" | jq -e '.status == "success" and (.patchesApplied +
.alreadyPatched) > 0' >/dev/null \
|| { echo "Socket patch apply: $(echo "$out" | jq -c '{status,
patchesApplied, alreadyPatched, failed}')"; exit 1; }
```

Scan an entire monorepo recursively from the root rather than running socket-patch per workspace.

```
$ socket-patch scan
```

— Named endpoint, flags, and commands across both releases.

Socket Patch CLI 5.1.0 (2026-11-11)

03 `socket fix` automated remediation and PR status NEW

95

Socket adds a `socket fix` command that calculates the smallest set of upgrades to clear a vulnerability — including bumping a parent package when needed — and applies them to the manifest and lockfile; it supports npm, pnpm, Yarn, uv, Poetry, Gradle,

`dotnet`, Cargo, Bundler, and Maven. Skip major-version bumps with `--no-major-updates`, or omit it for CVEs where a major upgrade is the only path, and enable auto-merge on fix PRs with `--autopilot` so they merge once required checks pass. The Open Socket Fix PR button is now disabled for unsupported ecosystems — npm, Maven, NuGet, Go, Cargo, PyPI, and RubyGems are supported — and Owners/Admins see a setup notice with a Settings link when Fix PRs aren't yet enabled for the org, while other members see no PR button until it's turned on.

Audit what `socket fix` would change across all vulnerabilities before touching any files — useful for a first-pass review in a new project.

```
$ socket fix --all --no-apply-fixes --output-file fixes.json
jq '.fixes | length' fixes.json
```

Schedule weekly automated fix PRs in CI, capped at 5 per run and restricted to package versions older than one week to reduce supply-chain risk.

yaml

```yaml
name: socket-fix
on:
  schedule:
    - cron: '0 6 * * 1'
permissions:
  contents: write
  pull-requests: write
jobs:
  socket-fix:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v4
        with:
          node-version: '22'
      - run: npm ci
      - name: Run Socket Fix
        env:
          SOCKET_CLI_API_TOKEN: ${{ secrets.SOCKET_CLI_API_TOKEN
}}
          SOCKET_CLI_GITHUB_TOKEN: ${{
secrets.SOCKET_FIX_GITHUB_TOKEN }}
        run: npx --yes socket fix --pr-limit 5 --minimum-release-
age 1w
```

Fix a single reachable CVE locally, verify tests still pass, then commit — keeping the blast radius to one advisory at a time.

```
$ socket fix --id GHSA-xxxx-xxxx-xxxx
npm ci
npm test
git diff
```

— Flags, supported ecosystems, and UI logic all named.

Clearer Availability and Status for Soc…

04 **Webhook delivery of firewall decisions** NEW 95

Socket Firewall adds a webhook to POST package decisions as JSON or NDJSON to any HTTPS endpoint. Enable it with `WEBHOOK_ENABLED=true` and `WEBHOOK_URL` in `.env`, with an optional `WEBHOOK_AUTH_HEADER` sent as the `Authorization` header; toggle which decision types are forwarded with `WEBHOOK_ON_BLOCK`, `WEBHOOK_ON_WARN`, `WEBHOOK_ON_IGNORE`, and `WEBHOOK_ON_MONITOR`. Enable batched NDJSON delivery (`application/x-ndjson`, one event per line) with `WEBHOOK_BATCH_ENABLED=true` and `WEBHOOK_BATCH_SIZE`, and disable TLS verification for local testing with `WEBHOOK_SSL_VERIFY=false`. Documented routing targets include Microsoft Sentinel or Defender XDR (via an Azure Function/Logic App), syslog-only SIEMs (via Logstash), and ticketing systems such as Jira or ServiceNow (filtering on `decision` = `blocked`).

Forward all firewall decision types to a SIEM or log collector so your SIEM can show allowed installs, warnings, and blocks, not just blocks alone.

```ini
WEBHOOK_ENABLED=true
WEBHOOK_URL=https://collector.example.com/socket-firewall
WEBHOOK_AUTH_HEADER=Bearer your_receiver_token
WEBHOOK_ON_BLOCK=true
WEBHOOK_ON_WARN=true
WEBHOOK_ON_IGNORE=true
WEBHOOK_ON_MONITOR=true
WEBHOOK_SSL_VERIFY=true
```

Apply the new `.env` webhook settings — environment variables are only read at container startup, so a force-recreate is required.

```
$ docker compose up -d --force-recreate
```

— Every env var and routing target named. product docs

05 **Metadata filtering removes blocked versions from listings**

NEW

95

Socket Firewall adds metadata filtering, which strips blocked versions from package metadata before the client resolves a version range. Enable it with `METADATA_FILTERING_ENABLED=true` in `.env`, or configure it via a top-level `metadata_filtering:` block in `socket.yml` with `enabled`, `filter_warn` (default `false`), `max_versions` (default `100` newest versions), `include_unchecked_versions`, and `cache_ttl`. Block unscanned versions from both listings and downloads with `socket.fail_open_unscanned: false` or `SOCKET_FAIL_OPEN_UNSCANNED=false`. Coverage spans npm, Go, RubyGems, Cargo, Conda, OpenVSX, NuGet, PyPI, and Maven, each filtered at the appropriate per-version or per-file granularity, and filter activity is logged as decision events with `event_category: "proactive"` and `decision: "blocked"`.

Turn on metadata filtering for a fresh firewall deployment so blocked versions are pruned from registry responses before npm resolves a semver range.

ini

```
METADATA_FILTERING_ENABLED=true
```

Verify that a blocked version is absent from the firewall's metadata listing before testing an install.

```
$ diff <(curl -s https://registry.npmjs.org/is-odd | jq -r
'.versions | keys[]') \
    <(curl -sk https://sfw.example.com/npm/is-odd | jq -r
'.versions | keys[]')
```

— Every config key, default value, and covered ecosystem named. product docs

06

Redis-backed distributed verdict caching

NEW

95

Socket Firewall adds Redis-backed distributed verdict caching so allow/warn/block decisions are shared across instances and survive restarts. Enable it with `REDIS_ENABLED=true`, `REDIS_HOST`, `REDIS_PORT`, and `REDIS_PASSWORD` in `.env`; control freshness with `cache.ttl` (default 600s) and `redis.ttl` (default 86400s), and set re-check behavior for expired allow verdicts with `cache.confirm_allow_mode` (default `wait`). Enable TLS for managed Redis (ElastiCache, Azure Cache for Redis) with `REDIS_SSL=true` and `REDIS_SSL_VERIFY=true`, plus optional `REDIS_SSL_CA_CERT`, `REDIS_SSL_CLIENT_CERT`, `REDIS_SSL_CLIENT_KEY`, and `REDIS_SSL_SERVER_NAME`, or equivalently a `redis:` block in `socket.yml` using `ssl_ca_cert`, `ssl_client_cert`, `ssl_client_key`, and `ssl_server_name`. The cache holds verdicts only — package bytes always stream from the upstream registry — and multiple instances share verdicts by pointing at the same `REDIS_HOST`. With

Redis enabled, restarting the firewall no longer clears verdicts; run `FLUSHDB` via redis-cli to force fresh verdicts after a policy change or waiver.

Add Redis to a Socket Firewall deployment so verdicts survive restarts and are shared across all firewall instances.

```env
REDIS_ENABLED=true
REDIS_HOST=redis
REDIS_PORT=6379
REDIS_PASSWORD=change_me_to_a_long_random_string
```

Connect a Socket Firewall instance to a managed Redis service (ElastiCache, Azure Cache for Redis) with mutual TLS.

```env
REDIS_ENABLED=true
REDIS_HOST=redis.example.com
REDIS_PORT=6379
REDIS_PASSWORD=change_me_to_a_long_random_string
REDIS_SSL=true
REDIS_SSL_VERIFY=true
REDIS_SSL_CA_CERT=/etc/redis-tls/ca.pem
REDIS_SSL_CLIENT_CERT=/etc/redis-tls/client.pem
REDIS_SSL_CLIENT_KEY=/etc/redis-tls/client-key.pem
REDIS_SSL_SERVER_NAME=redis.example.com
```

Force fresh verdicts after a policy change or waiver when Redis persistence is active, since a firewall restart alone no longer clears the cache.

```
$ rc() { docker compose exec -T redis redis-cli --no-auth-warning
-a "$(grep '^REDIS_PASSWORD=' .env | cut -d= -f2-)" "$@"; }
rc FLUSHDB
```

— Every env var, default TTL, and TLS option named. product docs

07 **Cooldown mode for recently published package versions** NEW

95

Socket Firewall adds Cooldown mode: set a Recently Published Alert Threshold in Dashboard → Settings → Alerts → Scans to auto-block new package versions until they age past it, holding back newly published versions for a configurable number of days. Allow recent versions through for selected ecosystems by adding `recently_published_enabled_ecosystems` to `socket.yml`; listed ecosystems receive a `200` with `x-socket-block-reason` suppressed, while unlisted

ecosystems stay blocked. Script cooldown waivers via the Policies API — `POST https://api.socket.dev/v0/orgs/{org}/alert-policies/{id}/rules` or `POST https://api.socket.dev/v0/orgs/{org}/alerts/resolutions` — using tokens scoped to `alert-policy:list`, `alert-policy:create`, or `alert-resolution:create`. In SIEM pipelines, count cooldown blocks on the `decision` field rather than `action`, since ecosystems allowed via `recently_published_enabled_ecosystems` log `decision: allowed` with `action: error`.

Allow recent npm packages through the cooldown while keeping other ecosystems blocked — add this to your existing `socket.yml` and recreate the container.

yaml

```yaml
socket:
  api_url: https://api.socket.dev
  fail_open: true
  recently_published_enabled_ecosystems:
    - npm
```

Programmatically grant a one-version waiver for a specific package in the new Policies system, so a pinned release clears the firewall's 403 without disabling the cooldown for other versions.

```
$ curl -s -X POST -H "Authorization: Bearer $SOCKET_API_KEY" -H
'Content-Type: application/json' \
  "https://api.socket.dev/v0/orgs/$ORG/alert-
policies/$DEFAULT_ID/rules" \
  -d '{"name":"Cooldown waiver rollup 4.25.0",
       "vigil_selector":
{"artifact.type":"npm","artifact.name":"rollup","artifact.version
":"4.25.0",

"finding.alertType":"recentlyPublished"},
       "action":"ignore",
       "note":"Waiver TICKET-123, approved by NAME"}'
```

Confirm Socket flags a freshly published package with a `recentlyPublished` alert before testing it through the firewall.

```
$ curl -s -X POST "https://api.socket.dev/v0/purl?alerts=true" \
  -H "Authorization: Bearer $SOCKET_API_KEY" \
  -H 'Content-Type: application/json' \
  -d '{"components":[{"purl":"pkg:npm/rollup@4.25.0"}]}' \
  | jq -c 'select(.name!=null)|[.alerts[]?
|select(.type=="recentlyPublished")|.action]'
```

— Dashboard path, config key, and scoped API endpoints all given. `product docs`

08 **Registry Mode Docker firewall proxy for npm and PyPI** NEW

95

Socket Firewall's Registry Mode runs as a Docker container (`socketdev/socket-registry-firewall:latest`) that proxies npm and PyPI requests and returns HTTP 403 for blocked packages. Configure per-ecosystem routes in `socket.yml` via `path_routing.routes` entries, each with a `path` , `upstream` registry URL, and `registry` type (`npm` or `pypi`); set `fail_open: true` / `false` to control behavior when the Socket API is unreachable, and `path_routing.use_incoming_domain: true` to preserve the client's host/port in rewritten download URLs. Authenticate with `SOCKET_SECURITY_API_TOKEN` (requires `packages:list` and `entitlements:list` scopes, distinct from the CLI/CI token), tune verdict-cache lifetime with `cache.ttl` , and control log verbosity with `SOCKET_IGNORE_LOG_LEVEL` . The quickstart guide adds `Configure registry: dev` and `Configure registry: CI/CD` instructions for pointing developer machines and CI/CD pipelines/Docker builds at the firewall.

Bootstrap the firewall for npm and PyPI in a strict posture so any Socket API outage blocks installs rather than allowing them through.

```yaml
socket:
  api_url: https://api.socket.dev
  fail_open: false
  ports:
    http: 8080
    https: 8443
  cache:
    ttl: 600
  path_routing:
    enabled: true
    domain: sfw.example.com
    rewrite_scheme: https
  routes:
    - path: /npm
      upstream: https://registry.npmjs.org
      registry: npm
    - path: /pypi
      upstream: https://pypi.org
      registry: pypi
```

Set required environment variables, including the firewall-specific API token, before starting the container.

```env
SOCKET_SECURITY_API_TOKEN=your_api_key_here
SOCKET_FAIL_OPEN=true
SOCKET_IGNORE_LOG_LEVEL=warn
```

Start the firewall and confirm it can reach Socket and load verdicts before routing real traffic through it.

```
$ docker compose up -d
for i in $(seq 1 30); do curl -sk https://sfw.example.com/health
&& break; sleep 2; done
docker compose logs socket-firewall | grep 'Resolved org slug
from API'
```

— Docker image, config keys, and token scopes all named. product docs

09 **SFW registry CI/CD integration via package-manager env vars**

NEW 90

Socket adds a CI/CD integration guide for the SFW registry, routing npm, pnpm, Yarn 1, and pip installs through the firewall proxy by setting `NPM_CONFIG_REGISTRY` to `https://sfw.example.com/npm/` and `PIP_INDEX_URL` to `https://sfw.example.com/pypi/simple/` in CI environment variables. Trust the firewall's CA for Node via `NODE_EXTRA_CA_CERTS` (adds to Node's existing trust list) and for pip via `PIP_CERT` (replaces pip's entire bundle, so point it at the system bundle `/etc/ssl/certs/ca-certificates.crt` after adding the firewall CA with `update-ca-certificates`); bake these into a container image with `ENV NPM_CONFIG_REGISTRY`, `PIP_INDEX_URL`, `NODE_EXTRA_CA_CERTS`, and `PIP_CERT`. Verify the integration by installing `lodahs` on a test branch and confirming the build fails with `E403 ... Blocked by Security Policy`.

Wire a GitHub Actions job to block malicious packages at install time for both npm and pip without changing your existing install steps.

yaml

```yaml
jobs:
  build:
    runs-on: ubuntu-latest
    env:
      NPM_CONFIG_REGISTRY: https://sfw.example.com/npm/
      PIP_INDEX_URL: https://sfw.example.com/pypi/simple/
    steps:
      - uses: actions/checkout@v4
      - name: Install firewall CA
        run: |
          printf "%s\n" "$SFW_CA_PEM" | sudo tee
/usr/local/share/ca-certificates/sfw-ca.crt > /dev/null
          sudo update-ca-certificates
          echo "NODE_EXTRA_CA_CERTS=/usr/local/share/ca-
certificates/sfw-ca.crt" >> "$GITHUB_ENV"
          echo "PIP_CERT=/etc/ssl/certs/ca-certificates.crt" >>
"$GITHUB_ENV"
        env:
          SFW_CA_PEM: ${{ secrets.SFW_CA_PEM }}
      - run: npm ci
```

Bake SFW registry and CA trust into a custom CI container image so every pipeline that uses it is protected with no per-job configuration.

```dockerfile
COPY sfw-ca.pem /usr/local/share/ca-certificates/sfw-ca.crt
RUN update-ca-certificates
ENV NODE_EXTRA_CA_CERTS=/usr/local/share/ca-certificates/sfw-ca.crt \
    PIP_CERT=/etc/ssl/certs/ca-certificates.crt \
    NPM_CONFIG_REGISTRY=https://sfw.example.com/npm/ \
    PIP_INDEX_URL=https://sfw.example.com/pypi/simple/
```

— Exact env vars, CA handling, and a runnable verification step. product docs

10 **Firewall decision-event schema and log routing** NEW

80

Socket documents the firewall decision-event data model, routing events to Splunk HEC, a webhook (Sentinel, Defender XDR, Elastic, ticketing workflows), or container stdout via environment variables. Every event is JSON with `event_type: firewall_package_encountered` and fields including `timestamp`, `client_ip`, `registryFqdn`, `repo`, `purl`, `artifact_purl`, `action`, `decision`, `block_reason`, `warn_reason`, `blocked_alerts`, `request_id`, and `cached`. Log every decision — including clean allows hidden by default — by adding `SOCKET_IGNORE_LOG_LEVEL=warn` to `.env`, then filter with `docker compose logs socket-firewall | grep SOCKET_DECISION`. Filter `event_category: "proactive"` events out of install-block counts, since each version listing generates one proactive event per version up to `max_versions`. Use `decision` rather than `action` to determine what the client actually received (some allowed downloads carry `action: error` under Cooldown). Detect the real requester behind a load balancer via `client_ip.header` and `client_ip.trusted_proxies` in `socket.yml`. Dedupe PyPI events on `purl` plus `client_ip`, since pip downloads produce two events with the same `purl` but different `source_path`.

Capture every firewall decision including clean allows to stdout so you can ship the full install-decision stream to your SIEM.

```
$ # Add to .env
SOCKET_IGNORE_LOG_LEVEL=warn

# Recreate the container and tail decision events
docker compose up -d --force-recreate
docker compose logs socket-firewall | grep SOCKET_DECISION
```

— Full field list and log-level variable given, no new endpoint. product docs

11 **Domain-based DNS override routing** NEW

80

Socket Firewall adds domain-based DNS override routing so clients reach the firewall without any package manager reconfiguration. Configure it in `socket.yml` using `registries[].domains` and `registries[].upstream` keys so the firewall answers directly for `registry.npmjs.org`, `pypi.org`, and `files.pythonhosted.org`. Generate a TLS certificate covering all registry hostnames with `-addext 'subjectAltName=DNS:registry.npmjs.org,DNS:pypi.org,DNS:files.pythonhosted.org'`, and prove it out on one machine by adding firewall IP entries to `/etc/hosts` before rollout. Track client coverage by matching the `client_ip` field in firewall decision events against MDM, DHCP, or VPN records.

Switch Socket Firewall from path routing to domain-based routing so npm and pip clients need no reconfiguration — back up the existing config first.

```
$ cp socket.yml socket.yml.path-routing
```

Domain-based routing config that makes the firewall intercept npm and PyPI traffic transparently using real registry hostnames.

```yaml
socket:
  api_url: https://api.socket.dev
  fail_open: true
  ports:
    http: 8080
    https: 8443
  registries:
    npm:
      domains: [registry.npmjs.org]
      upstream: https://registry.npmjs.org
    pypi:
      domains: [pypi.org, files.pythonhosted.org]
      upstream: https://pypi.org
```

Validate that a single test machine is fully covered before org-wide rollout — no real registry addresses should appear.

```
$ getent ahosts registry.npmjs.org
```

— Config keys and a test command both given. product docs

12 **DNS-aware liveness endpoint for Kubernetes** NEW 80

Socket Registry Firewall adds a `GET /health/live` endpoint that checks DNS resolution across every nginx worker, so Kubernetes can restart the firewall when a worker fails its DNS check.

Wire the new DNS-aware endpoint into a Kubernetes liveness probe so the pod restarts automatically when any nginx worker loses DNS resolution.

```yaml
livenessProbe:
  httpGet:
    path: /health/live
    port: 80
  initialDelaySeconds: 10
  periodSeconds: 15
```

— Exact endpoint and a ready-to-use probe config.

New: DNS-Aware Liveness Checks in Socke…

13 **`displayName` field in Threat Feed API** NEW 65

Threat Feed API responses and webhooks now include a `displayName` field containing the registry-native spelling of a package name (for example, the crates.io spelling of a Cargo crate), making it easier to match a threat feed entry to the package under investigation; the field may be `null` for Cargo, browser extension, and ClawHub entries and should be treated as optional.

When consuming the Threat Feed API, read `displayName` to get the exact crates.io (or other registry) spelling of a flagged package, then cross-reference it directly in your registry or dependency manifest.

```
$ curl -s -H 'Authorization: Bearer <token>'
'https://api.socket.dev/v0/threat-feed' | jq '.[] | {displayName:
.displayName, name: .name, severity: .severity}'
```

— Field name and nullability conditions given, no new endpoint.

New: displayName in Socket threat feed …

14 **Search and date filters for PR Scans** NEW 60

Socket adds search by repository, branch, PR number, title, or author on the Scans > PRs tab (case-insensitive, minimum two characters), and a `Filter → Created` option that replaces the old `Last 90 days` dropdown with presets and custom date ranges. Repositories → Labels can now be filtered by `Security policy` or `License policy` status (Enabled or Disabled), and search/filter state is reflected in the page URL for bookmarking and sharing.

Find all PR scans by a specific author within a custom date range and share the resulting view with a teammate.

> 📍 In the Socket dashboard, go to Scans › PRs. Type the author name in the search box, then open Filter → Created and set a custom start and end date. Copy the URL from the browser address bar and share it.

Quickly audit which labels in your org have Security policy disabled so you can triage policy gaps.

> 📍 In the Socket dashboard, go to Repositories → Labels. Set the Security policy filter to Disabled to list all labels where the policy is off.

— UI navigation given but no API or config surface.

New: Find PR Scans Faster with Search a…

THINNER COVERAGE BELOW

15 **Stale `.socket.facts.json` uploaded as SBOM** BREAKING

55

If `.socket.facts.json` is present when `socket scan create` is run without `--reach`, Socket uploads it as an SBOM carrying stale reachability results, affecting NuGet, Go, Python, and JavaScript scans; delete the file first to avoid this.

— Clear cause and fix, but no mechanism beyond that. product docs

16 **Faster policy propagation to Socket Firewall** IMPROVED

45

Hosted Socket Firewall now propagates security policy, license policy, and triage rule changes in about 5 seconds, down from a 10-minute cache for decisions and up to an hour for package version listings.

— Concrete before/after timing but nothing for a reader to do.

New: Policy changes reach Socket Firewa…

NEW DETECTION Dual-use functionality alert

└──▷ BREAKING ON UPGRADE

! Delete `.socket.facts.json` before running `socket scan create` without `--reach`; if the file is present it is uploaded as an SBOM carrying stale reachability results.

! Delete `.socket.facts.json` before running `socket scan create` without `--reach`; otherwise the file is uploaded as an SBOM carrying stale reachability results.

! If `.socket.facts.json` is present and you run `socket scan create` without `--reach`, the file is uploaded as an SBOM with stale reachability data — delete it first to avoid this.

! Delete `.socket.facts.json` before running `socket scan create` without `--reach`; if the file is present it is uploaded as an SBOM and carries stale reachability results.

! With Redis enabled, restarting the firewall no longer clears verdicts — run `FLUSHDB` via redis-cli to force fresh verdicts after a policy change or waiver.

> New alert covering dual use tools (2026-11-13) BLOG ↗ LATEST STABLE

WAS THIS USEFUL? 👍 👎

REVERSE ENGINEERING, MALWARE & FORENSICS

◆ Digital Forensics & Incident Response

ElcomSoft

Feedback ⓘ changelog · Nov 16, 2026 CHANGELOG ↗

ElcomSoft is a try professional password recovery, data decryption, mobile and cloud forensic tools from a manufacturer with 30+ years of expertise, providing tools and training to law enforcement, financial and intelligence agencies.

NEW SUPPORTED PLATFORM Low-Level Extraction of the Apple Watch S4/S5

WAS THIS USEFUL? 👍 👎

WithSecure chainsaw

Feedback ⓘ v2.18.0 NOTES ↗ RANK

Chainsaw is a rapidly Search and Hunt through Windows Forensic Artefacts

> Chainsaw v2.18.0 expands windash expansion with permutation support and pulls in a newer upstream EVTX library that introduces breaking changes.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Permutation support in windash expansion** NEW 40

The sigma-to-tau parser's windash expansion now supports permutations, broadening detection coverage for Windows command-line variations.

— Explains mechanism but no concrete command/flag examples v2.18.0

02 **Upstream EVTX library update** BREAKING 35

Chainsaw updates its upstream EVTX library to a newer version, which introduces breaking changes (see issue #248); existing integrations or workflows depending on EVTX parsing behaviour may require adjustment.

— Notes breaking change and issue reference, lacks specifics on behaviour change v2.18.0

└──▷ **BREAKING ON UPGRADE**

! Updating the upstream EVTX library introduces breaking changes (see issue #248); existing integrations or workflows depending on EVTX parsing behaviour may require adjustment.

WAS THIS USEFUL? 👍 👎

◆ Malware Analysis

CAPEv2 Feedback ⓘ changes since 2026-11-13 CODE ↗

CAPEv2 is an open-source malware sandbox that automatically executes and analyzes suspicious files to extract configurations, payloads, and behavioral indicators.

CAPEv2 adds a bulk report download API, payload dump-trigger attribution, and a Linux process dump option

└──▷ TRY IT

Download reports for a range of tasks as a SHA256-named ZIP archive for bulk triage or archival.

```
$ curl -s 'https://<cape-host>/apiv2/tasks/get/reports/bulk/101-105/json/' -o reports_101_105.zip
```

Enable bulk report downloads and raise the per-request cap in the API config before exposing the endpoint.

conf/default/api.conf.default

```
[taskreport]
enabled = yes
bulk_download = yes
bulk_max_reports = 10
```

Download multiple analysis reports at once via the new `/apiv2/tasks/get/reports/bulk/` endpoint (and `/apiv2/tasks/get/reports/bulk/<task_ids>/<report_format>/`), accepting task ID ranges or lists (e.g. `101-103` or `101,102,105`), a `report_format` of `json`, `litereport`, `html`, `htmlsummary`, `pdf`, `maec`, `maec5`, `metadata`, `protobuf`, or `parti`, and an optional `package` filter; results are streamed as a `.zip` archive with entries named `<sha256>.<ext>`.

Gate bulk report downloads with the new `bulk_download` and `bulk_max_reports` keys in `conf/default/api.conf.default` (defaults: `bulk_download = no`, `bulk_max_reports = 3`), alongside the existing `[taskreport] enabled` guard.

CAPEv2 now shows the specific debugger breakpoint (`bp0`–`bp3` / `br0`–`br3` / `action0`–`action3`) or Dump-on-API trigger that produced each payload via the new extract_dump_triggers() and CAPE._resolve_dump_trigger() functions, displayed as a Dump Trigger row in the Payloads tab when `file.dump_trigger` is present.

CAPEv2 now supports `procdump` as an analysis option for Linux guests.

WAS THIS USEFUL? 👍 👎

NETWORK SECURITY

◆ Network Analysis & Monitoring

Ettercap Feedback ⓘ CVE-2026-129432 CVE ↗

Ettercap is an open-source network security tool for man-in-the-middle attacks, enabling live packet sniffing, filtering, and protocol analysis on local networks.

`CVE-2026-129432`: Heap out-of-bounds read in the mountd dissector's remote-directory copy (CVSS 6.5, affects `<= 0.8.5.1`)

WAS THIS USEFUL? 👍 👎

Netcap

Feedback ⓘ v0.16.7 NOTES ↗ CODE ↗

Netcap is a framework for secure and scalable network traffic analysis - https://netcap.io

Netcap v0.16.7 adds `--pcapng-merge-sections`, per-run local API authentication, `directoryMatches` in Hosts.json, and a 3x speed improvement for large UDP-flow workloads.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.16.7 https://github.com/dreadl0ck/netcap.git
# already have the repo? check out this version:
$ git checkout v0.16.7
```

Enable continuous-interface PCAPNG processing across concatenated sections with `--pcapng-merge-sections` (or env var `NC_PCAPNG_MERGE_SECTIONS`)

Offline web UI is now opt-in; per-run authentication, strict Host/Origin checks, same-origin CORS, an HttpOnly SameSite cookie for browser bootstrap, and MCP backend authentication protect the local API

Print the local web UI access token to stderr at startup when the web UI server starts

Add `directoryMatches` field to `Hosts.json` entries, listing LDAP computer objects whose name equals a host's observed name

Index service-flow membership to reduce processing time for large UDP-flow workloads, cutting the 40k-UDP-flow case from 3.21 s to 1.09 s

+1 more

UPDATED DETECTION c2.beacon corroboration annotation

UPDATED DETECTION policy.http-on-443 false-positive reduction

NEW SUPPORTED PLATFORM USB link types 189, 220, 249

NEW SUPPORTED PLATFORM DLT 108, 50, 117, 119 link types

NEW DETECTION NBNS node-status decoding

UPDATED DETECTION DNS binary NULL tunnel detection

└──▷ **BREAKING ON UPGRADE**

! Offline capture runs (`--read`) no longer start the web UI by default; pass `--http <addr>` explicitly to enable it.

! Invalid or non-capture additional positional arguments now return a fatal error (`unexpected capture argument`) instead of being silently skipped.

! The `--gen-config` flag now returns an error ('capture `--gen-config` is not implemented; use NC_* environment variables or command flags') instead of printing a stub message.

WAS THIS USEFUL? 👍 👎

CLOUD & CONTAINER SECURITY

◆ CNAPP

Plerion

Feedback ⓘ See the tools your agents define: NOTES ↗

RANK

Plerion is a cloud security platform that provides comprehensive risk management, vulnerability detection, and threat prioritization across cloud environments and workloads.

> Plerion's Agent security now surfaces the tools an Amazon Bedrock AgentCore runtime actually defines, and adds an Access tab showing reachability across runtimes, gateways, and agents.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 Tool-definition visibility for AgentCore runtimes NEW

60

Agent security now shows the tools an Amazon Bedrock AgentCore runtime defines in its code for agents built with LangChain, LangGraph, or Strands. This is read automatically from the code zips and container images Plerion already scans, requiring no additional setup — select a runtime and open the Tools view to see every defined tool.

Review what tools an AgentCore runtime exposes before approving it for production, without any additional setup if Plerion already scans its code.

📍 In Agent security, select an AgentCore runtime and open the Tools view to see every tool defined in its LangChain, LangGraph, or Strands code.

— Names frameworks and source but no limits or output format

See the tools your agents define:

02 Access tab for AgentCore runtimes, gateways, and agents NEW

60

Every AgentCore runtime, gateway, and Bedrock agent in Agent security now has an Access tab showing what each one can reach and what can reach it, displayed as a graph or a list, with IAM and resource policies applied.

Audit the blast radius of a Bedrock agent — what it can reach and who can reach it — using IAM and resource policies.

📍 In Agent security, select any AgentCore runtime, gateway, or Bedrock agent, then open the Access tab and toggle between the graph and list views.

— Clear UI path and mechanism, but no CLI/API surface See the tools your agents define:

WAS THIS USEFUL? 👍 👎

◆ Container & Image Security

Mend.io Mend Container Feedback ⓘ

Version 26.10.1 (01-November-2026) NOTES ↗

Mend Container is a traditional application security misses AI risk. Standalone AI security tools miss your code.

NEW VULNERABILITY CHECK Binary package type vulnerability scanning

WAS THIS USEFUL? 👍 👎

DETECTION & RESPONSE

◆ Detection Engineering

Tenzir Feedback ⓘ Tenzir Skills v4.1.0 (2026-11-15) BLOG ↗ RANK

Tenzir is a data pipeline engine for security teams that collects, transforms, enriches, and routes telemetry.

> Tenzir shipped a small update to Tenzir Skills, mirroring agent documentation locally for offline parity with the Platform.

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

01 **Local mirroring of agent map and docs** IMPROVED 45

The `tenzir` skill now mirrors the agent map and documentation pages published on tenzir.com, so agents using the skill locally read the same documentation as agents connected to the Tenzir Platform via MCP.

– Explains what changed but no config key or command given

Tenzir Skills v4.1.0 (2026-11-15)

WAS THIS USEFUL? 👍 👎

◆ SIEM

graylog2-server Feedback ⓘ CVE-2023-41044 CVE ↗ STABLE

Graylog2-server is a free and open log management

`CVE-2023-41044` : Partial path traversal vulnerability in Support Bundle feature (affects `>= 5.1.0`)

▾ **1 more release in this issue** · 2023-07-05

> CVE-2023-41041 CVE ↗ STABLE
>
> `CVE-2023-41041` : User session is still usable after logout (affects >= 1.0 (?))

WAS THIS USEFUL? 👍 👎

EXPOSURE & VULNERABILITY MANAGEMENT

◆ Endpoint Management (UEM/RMM)

Microsoft Intune

Feedback ⓘ

Week of October 26, 2026 (Service release 2610) NOTES ↗

Microsoft Intune is a cloud-based unified endpoint management solution built on Zero Trust. Empower IT to manage, secure and govern devices, apps and endpoints.

Intune 2610 adds STIG audit baselines, certificate connector health monitoring, Apple Enforce Routes VPN support, Administrator protection settings, and faster Windows script delivery via push notifications.

NEW COMPLIANCE FRAMEWORK Audit Windows devices against STIG recommendations in commercial tenants using the new STIG audit baselines feature, which evaluates per-setting and per-device compliance without changing device settings — requires Intune Advanced Analytics.

NEW SUPPORTED PLATFORM RETIRED SUPPORTED PLATFORM Configure Apple VPN profiles on iOS/iPadOS and macOS to now support Apple's Enforce Routes capability, with routing behavior defined via `Include all networks` and `Exclude local networks` settings.

View certificate issuance health signals directly in the Microsoft Intune admin center via the Certificate Connector, with indicators for CA connectivity failures, service account permission issues, and PKCS template mismatches.

Configure two new Administrator protection settings — Administrator Approval Mode type and elevation prompt behavior — in endpoint security Account protection policies for supported Windows 11 versions.

Configure the same two Administrator protection settings in the settings catalog for Windows 11 versions 24H2 and 25H2.

+3 more

NEW INTEGRATION MEFERI OEMConfig app support for Android Enterprise

NEW INTEGRATION New protected apps available for Intune

NEW SUPPORTED PLATFORM AOSP referenced app type for Android

└──▷ BREAKING ON UPGRADE

! Setup Assistant (legacy) authentication for iOS/iPadOS and macOS automated device enrollment with user affinity is no longer available when creating or editing profiles; existing profiles are automatically migrated to Setup Assistant with modern authentication.

! Legacy Apple MDM software update workloads have been removed from Intune following Apple's retirement of the underlying commands and payloads in OS 27; admins must switch to Apple declarative device management update policies.

WAS THIS USEFUL? 👍 👎

IDENTITY, ACCESS & DATA PROTECTION

◆ Secure Access (VPN / SASE / ZTNA)

Twingate

Feedback ⓘ changelog · Nov 16, 2026 CHANGELOG ↗ RANK

Twingate is an identity-based access for users, services, and AI agents that deploys in minutes, scales to every resource, and finally lets you retire your VPN.

> Twingate expanded its Privileged Access Management offering with SSH Resource registration for servers and VMs and added support for protecting internal HTTP web apps.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **SSH Resource registration in PAM** NEW 71

Servers and VMs can now be registered as SSH Resources in Twingate, letting users authenticate via their identity provider while the Twingate Gateway handles certificate-based authentication. Every session is recorded to stdout in asciicast format for SIEM forwarding or audit replay.

— Names mechanism and format but no exact command or UI path changelog-20261116-0fce4e7f

THINNER COVERAGE BELOW

02 **HTTP web app protection in PAM** NEW 56

Twingate Privileged Access Management now supports HTTP web apps, enabling MFA, JIT access, security policies, and access logging for internal web apps, data portals, CI/CD infrastructure, and AI-coded business apps, without requiring users to re-authenticate at the app.

— Lists capabilities and use cases but no config or endpoint details

changelog-20261116-0fce4e7f

WAS THIS USEFUL? 👍 👎

GOVERNANCE, RISK & COMPLIANCE

◆ Integrated & IT Risk Management

CISO Assistant

Feedback ⓘ v4.3.2 NOTES ↗ RANK

CISO Assistant is a one-stop-shop GRC platform for Risk Management, AppSec, Compliance & Audit, TPRM, BIA, Privacy, and Reporting. It supports 200+ global frameworks with automatic control mapping, including ISO 27001, NIST CSF, SOC 2, CIS, PCI DSS, NIS2, DORA, GDPR, HIPAA, CMMC, and more.

> CISO Assistant's only update this window is a set of new MCP tools for managing documents directly through the MCP interface.

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

01 **Document management via MCP tools** NEW 60

Documents can now be managed through the MCP interface using three new tools: `get_documents`, `create_document`, and `update_document`.

— Names exact tool calls but no detail on parameters or behavior v4.3.2

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI Security Posture & Discovery

Mend.io Mend AI

Feedback ⓘ changelog · Nov 16, 2026

CHANGELOG ↗ RANK

Mend AI is a traditional application security misses AI risk. Standalone AI security tools miss your code.

> Mend AI's reporting layer gained two new report types—one that traces chat AI system prompts to their code location and an expanded AI inventory covering the full agent/tool/MCP stack—plus the ability to run any report at the organization level.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 **AI Inventory report expanded and renamed** BREAKING 55

The AI Inventory report now includes one row per detected AI entity across models, system prompts, agents, agent configurations, tools, and MCP servers. The previous AI Inventory report, which covered AI models only, has been renamed to Models Inventory.

– Names entity types and the renamed report but no migration steps.

changelog-20261116-605c5559

02 **Conversational System Prompts report** NEW 50

A new report type that locates the system prompts behind chat-based AI applications and shows exactly where each one is defined in code.

– Describes mechanism and output but no UI path or command given.

changelog-20261116-605c5559

03 **Organization-scope report generation** NEW 35

All Mend AI reports can now be generated at the organization scope, in addition to the existing application and project scopes.

– States the new scope but no further detail on usage. changelog-20261116-605c5559

└──▷ **BREAKING ON UPGRADE**

! The previous AI Inventory report, which covered AI models only, is renamed to Models Inventory.

WAS THIS USEFUL? 👍 👎

◆ AI Red Teaming & Testing

Giskard

Feedback ⓘ giskard-scan/v1.0.2 NOTES ↗ RANK

Giskard tests and evaluates LLM applications for performance, bias, and security issues.

> Giskard shipped a modular v3 scanner that replaces the old v2 LLM scan and RAGET, adding OWASP-aligned red-teaming, RAG knowledge-base quality evaluation, and synchronous execution helpers for agent testing.

└──▷ **WHAT SHIPPED · 8 FEATURES** most completely described first ? what's the number?

01 v3 LLM vulnerability scanner with custom generators NEW

88

Giskard v3 introduces a modular vulnerability scanner that runs automated red-teaming across OWASP LLM Top-10 threat categories — prompt injection, harmful content, stereotypes, and misinformation — via vulnerability_scan(target, description, languages) from `giskard.scan`. Scans can be extended with custom adversarial generators by passing `ScenarioGenerator` instances to `generate_suite`, or by registering them on `vulnerability_suite_generator_registry`.

Red-team a customer support agent against OWASP LLM Top-10 threat categories including prompt injection and harmful content.

python

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

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

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

asyncio.run(main())
```

— Full function signature, threat categories and runnable example included

giskard-scan/v1.0.2

02 **v2 LLM scan and RAGET deprecated in favor of v3** BREAKING

78

Giskard v2 is no longer actively maintained; v3 (`giskard-scan`) replaces the v2 LLM scan and RAGET for agentic systems. The v2-only automatic tabular scan (`giskard.Model` + `giskard.Dataset`) and the `giskard.testing` ML test suite are not planned for v3 — pin `pip install 'giskard[llm]>2,<3'` to keep using them.

— Names deprecated surfaces and gives migration pin command `giskard-scan/v1.0.2`

03 **Synchronous run_sync() for scenarios and suites** NEW

70

Call run_sync() directly on a scenario or suite to run evals from a plain script or CI pipeline without managing an event loop or calling asyncio.run().

Run a Giskard scenario in a plain script or CI pipeline without needing asyncio.run() or an existing event loop.

python

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

def get_answer(inputs: str) -> str:
    return "Paris"

scenario = (
    Scenario("test_france_capital")
    .interact(inputs="What is the capital of France?",
outputs=get_answer)
    .check(
        Groundedness(
            name="answer is grounded",
            context="France is in Western Europe. Its capital is
Paris.",
        )
    )
)
result = scenario.run_sync()
result.print_report()
```

— Runnable example demonstrates exact method usage `giskard-core/v1.0.4`

04 **Scanner installation and provider extras** NEW

60

Install the scanner with `pip install 'giskard[scan]'` or `pip install giskard-scan` ; LLM provider extras such as `giskard[openai]` or `giskard[anthropic]` are available separately.

– Exact install commands given, nothing further to explain giskard-scan/v1.0.2

05 **Telemetry opt-out via environment variables** NEW 60

Opt out of telemetry by setting `DO_NOT_TRACK=1` or `GISKARD_TELEMETRY_DISABLED=1` as environment variables or keys in a `.env` file in the working directory.

– Exact variable names and config location given giskard-scan/v1.0.2

THINNER COVERAGE BELOW

06 **RAG quality evaluation with quality_scan and KnowledgeBase**

NEW 50

Giskard adds `quality_scan` and `KnowledgeBase`, the v3 successor to RAGET, for evaluating RAG knowledge-base quality in agentic systems.

– Names the new function and class but shows no usage giskard-scan/v1.0.2

07 **Agent prompt template validation** NEW 45

Validate agent prompt templates before execution with the new `validate_template` function in `giskard.agents`.

– Names function but no example or mechanism given giskard-core/v1.0.4

08 **Graceful handling of unavailable Hub checks** IMPROVED

35

Checks that are unavailable on the Hub now warn and skip rather than failing the entire run.

– Describes behavior change without naming a surface giskard-core/v1.0.4

└──▷ **BREAKING ON UPGRADE**

! Giskard v2 is no longer actively maintained; v3 (`giskard-scan`) replaces the v2 LLM scan and RAGET for agentic systems. The v2-only automatic tabular scan (`giskard.Model` + `giskard.Dataset`) and `giskard.testing` ML test suite are not planned for v3 — pin `pip install 'giskard[llm]>2,<3'` to keep them.

WAS THIS USEFUL? 👍 👎

◆ AI Runtime Defense & Guardrails

doberman-core

Feedback ⓘ v0.21.0 NOTES ↗ RANK

> doberman-core's v0.21.0 release introduces Windows Hello as a TPM-backed 2FA factor for AI agent action approvals, replacing TOTP secrets, alongside two breaking packaging and hook changes.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 Windows Hello 2FA mode for agent approvals NEW

90

Windows Hello can now be enrolled as the sole 2FA factor with `doberman 2fa hello enroll` (run from an admin terminal), which deletes the TOTP secret and pins a TPM-backed key so no 2FA secret is stored on disk. Approval and loosening commands now accept a Windows Hello gesture instead of a TOTP code, and each approval is stored as a Hello-key signature tied to the specific action for later audit. `doberman doctor` now reports Windows Hello enrollment status alongside other health signals.

Replace TOTP-based 2FA with a Windows Hello gesture on a machine with a TPM, so no 2FA secret is ever stored on disk.

```
$ doberman 2fa hello enroll
```

Confirm that Windows Hello mode is active and check overall guardrail health after enrolling.

```
$ doberman doctor
```

— Full enroll command, mechanism and audit behaviour all named. v0.21.0

02 Cursor hook timeout updated to 660 seconds BREAKING

75

After upgrading, users must re-run `doberman install-hooks --host cursor` so the Cursor hook picks up the new 660-second wait timeout; hooks installed before the upgrade keep using the old value until reinstalled.

— Exact command and new timeout value given, no old value. v0.21.0

THINNER COVERAGE BELOW

03 winhello pip extra now empty BREAKING

50

The `winhello` pip extra is now empty; any install that pinned `doberman-core[winhello]` pulls no extra dependencies.

— States the change but not what replaces the removed dependency. v0.21.0

└──▷ **BREAKING ON UPGRADE**

! Cursor users must re-run `doberman install-hooks --host cursor` after upgrading, or the hook will use the wrong timeout (the new value is 660 seconds).

! The `winhello` pip extra is now empty; any install that pinned `doberman-core[winhello]` pulls no extra dependencies.

WAS THIS USEFUL? 👍 👎

◆ Model & Agent Protection

Minimal Software Research minimal

Feedback ⓘ

release-32b1f303 NOTES ↗ RANK

Minimal is a secure, declarative runtimes for Claude Code, Gemini, Codex, Pi and more, so you can run multiple agents safely on your stack from a verified software supply chain.

> This release hardens box security with sealed-box loadout restrictions and a hash-chained audit trail, adds a GitHub App-based sign-in flow for `min login`, and introduces a daily update check reported via `min`.

└──▷ WHAT SHIPPED · 4 FEATURES most completely described first ? what's the number?

01 **GitHub sign-in via `min login`** NEW 85

`min login` is now the GitHub sign-in entry point, completing sign-in on a real page with repository choice. Sign-in happens under a published GitHub App, with keys held in the host keychain, the Secure Enclave (macOS), TPM (Linux), or Secret Service (Linux hosts with no usable TPM).

— Names exact command and the platform-specific key-storage backends. release-32b1f303

02 **Hash-chained audit trail and JSON spec inspection for boxes** NEW 85

`min box audit --parent mybox` reviews the full hash-chained audit trail for a box, merging its parent chain, to verify no decisions were edited. `min box show -o json mybox` inspects a box's declared spec and egress in machine-readable JSON for scripted policy checks.

Review the full hash-chained audit trail for a box, merging its parent chain, to verify no decisions were edited.

```
$ min box audit --parent mybox
```

Inspect a box's declared spec and egress in machine-readable form for scripted policy checks.

```
$ min box show -o json mybox
```

— Exact commands and flags make this immediately runnable. release-32b1f303

03 **Loadout restrictions on sealed boxes** NEW 65

A sealed box now refuses a named loadout, a loadout that sets `loadout_mode`, and a contradicted constraint; a sealed box reads no default loadouts.

— Names exact rejected conditions but no runnable example. release-32b1f303

THINNER COVERAGE BELOW

04 **Daily update check via `min`** NEW 35

minimal now checks once a day whether the channel has a newer release and reports it via `min`.

— Describes behavior but no command or config shown. release-32b1f303

WAS THIS USEFUL?  

OTHER / UNCATEGORIZED

◆ UNCATEGORIZED

vigolium

Feedback ⓘ v0.5.7 NOTES ↗ RANK

Vigolium - High-fidelity vulnerability scanner fusing agentic AI with native speed, modularity, and precision

> Vigolium v0.5.7 adds safe-mode scanning, scan resume, baseline diffing and OAST session persistence, while reworking redirect handling, finding identity, and numerous CLI command defaults with breaking changes across fuzzing, replay, and export workflows.

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

01 **Scan control flags** NEW 88

Adds new scan control flags: `--safe` limits scans to read-only HTTP methods (GET/HEAD), skips forms, uploads and logins, caps request pace and enforces a 15-minute budget (with a method allow-list settable via the `scope.methods` config key); `--resume-scan <uuid>` resumes a scan from its coverage ledger, avoiding re-crawling already-covered paths; `--max-time` or the `scanning_pace.scan_max_duration` config key caps total scan duration (defaults to 1 hour); `--baseline` and `--ignore-file` (pointing to a `.vigolium-ignore.yaml` file) suppress pre-existing findings, with `--fail-on` and the new `scan diff` command counting only new, un-ignored findings for CI gating; and `--require-scope` or the `VIGOLIUM_REQUIRE_SCOPE=1` environment variable forces a scan to refuse running against a scope that names no hosts.

Run a low-impact scan during business hours that stays read-only and times out automatically — safe for production targets.

```
$ vigolium scan https://app.example.com --safe --max-time 10m
```

Pick up a scan that was interrupted by a timeout or Ctrl-C without re-crawling already-covered paths.

```
$ vigolium scan --resume-scan a1b2c3d4-e5f6-7890-abcd-ef1234567890
```

In CI, compare a feature branch scan against a saved baseline so only regressions fail the build.

```
$ vigolium scan https://staging.example.com --baseline
baseline.jsonl --ignore-file .vigolium-ignore.yaml --fail-on high
```

— Named flag, config key, and exact behavioral limits given. Flag and mechanism named, no detail on ledger format. Flag, config key and default value given, no mechanism beyond cap. Multiple named flags, file format, and CI use case specified. Flag and env var named, behavior is simple.

v0.5.7

---

02 **Credential redaction in reports and raw views** NEW

86

Credentials can be redacted in reports and notifications via the `output.redact_evidence` config key (suppressed with `--no-redact`), and masked in raw request/response views under `-j` using `VIGOLIUM_REDACT` or `--redact` / `--no-redact`; the JSON envelope now carries a `redacted` field reflecting this state.

— Multiple named config keys, env var, and envelope field given. v0.5.7

---

03 **Redirect-mode default scoped to in-scope hosts** BREAKING

72

`--redirect-mode` now defaults to `in-scope`; cross-host redirects no longer carry session or credential headers, and raw requests no longer follow redirects.

— Exact flag and before/after behavior named. v0.5.7

---

04 **Scan commands attack only direct input** BREAKING

70

`scan-url`, `scan-request`, and `run -t` now attack only their direct input; pass `--include-stored` to include stored records as before.

— Commands and flag named with before/after behavior. v0.5.7

---

05 **Header fuzzing scope control** IMPROVED

68

Header fuzzing now defaults to only headers that can sink a vulnerability class via `--header-fuzz gated`; passing `--header-fuzz all` restores fuzzing of all headers.

— Exact flag values given, limited further explanation. v0.5.7

---

06 **`fuzz` requires explicit insertion point** BREAKING

68

`fuzz` now requires an explicit insertion-point position and defaults to `--rate-limit 100` and `--max-requests 10000`.

— Command, flags and defaults all named. v0.5.7

---

07 **Dry-run preview for mutating commands** NEW — 64

The `--dry-run` flag previews any mutating or traffic-sending command without sending requests, including `agent autopilot` and `agent swarm`.

— Flag and affected commands named, no further mechanism. v0.5.7

08 **`replay -i` sends each record once** BREAKING — 62

`replay -i` now sends only once; URL positionals and multi-record matches without `--all-matches` are refused.

— Flags and refusal conditions named, limited mechanism. v0.5.7

09 **Finding identity v2 schema (v6)** BREAKING — 60

Finding identity now keys on location and insertion point under schema v6; older vigolium binaries cannot open databases written by this version.

— Schema version and migration risk named, no migration path given. v0.5.7

THINNER COVERAGE BELOW

10 **OAST session persistence after scan ends** NEW — 58

OAST sessions now stay alive after a scan completes, and late out-of-band callbacks can be retrieved afterward with `oast poll`.

— Names the subcommand but little mechanism detail. v0.5.7

11 **`--intensity` replaces strategy/profile flags** BREAKING — 58

`--strategy` and `--scanning-profile` are moved to Advanced presets; `--intensity` becomes the public dial for scan behavior.

— Flags named, no detail on intensity levels. v0.5.7

12 **Default `db export` format changes to JSONL** BREAKING — 56

`db export --format nuclei` was the previous default output; the new default format is `--format jsonl`.

— Command and both formats named explicitly. v0.5.7

13 **OAST public fallback disabled by default** BREAKING — 54

`oast.fallback_to_default` now defaults to `false`; public OAST fallback is no longer active unless explicitly enabled.

— Config key and new default named, little else. v0.5.7

14 **JSON envelope `ok` field tied to exit code** BREAKING 48

`ok` in the JSON envelope is now true exactly when the exit code is 0, 3, or 4.

— Exact field and codes named, no further context. v0.5.7

15 **`--ci-output-format` hidden and errors** BREAKING 46

`--ci-output-format` is now hidden from help output and exits with code 2 when used.

— Thin: names the flag and exit code only. v0.5.7

16 **Unreadable config file now fatal** BREAKING 46

An unreadable config file is now fatal, raising `config_invalid`, rather than being silently ignored as before.

— Error code named, minimal mechanism. v0.5.7

└──▷ BREAKING ON UPGRADE

! `--redirect-mode` now defaults to `in-scope`; cross-host redirects no longer carry session or credential headers, and raw requests no longer follow redirects.

! Finding identity v2 now keys on location and insertion point (schema v6); older vigolium binaries cannot open databases written by this version.

! `--ci-output-format` is hidden and now exits 2.

! `fuzz` now requires an explicit insertion-point position and defaults to `--rate-limit 100` and `--max-requests 10000`.

! `replay -i` now sends only once; URL positionals and multi-record matches without `--all-matches` are refused.

! `ok` in the JSON envelope is now true exactly when the exit code is 0, 3, or 4.

! An unreadable config file is now fatal (`config_invalid`) rather than silently ignored.

! `scan-url`, `scan-request`, and `run -t` now attack only their direct input; pass `--include-stored` to include stored records.

! `oast.fallback_to_default` defaults to `false`; public OAST fallback is no longer active unless explicitly enabled.

! `--strategy` and `--scanning-profile` are moved to Advanced presets; use `--intensity` as the public dial.

! `db export --format nuclei` is the old default output; the new default format is `--format jsonl`.

WAS THIS USEFUL? 👍 👎

Sandbox0 s0

Feedback ⓘ v0.10.6 NOTES ↗ RANK

s0 is the open-source CLI for Sandbox0, a hosted execution environment for long-running AI agents.

> s0 introduced optional preauthorized memory burst for sandboxes, letting sandboxes in the `ali-ue1` region expand memory by up to 50% on demand without restarting processes.

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

01 **Dynamic memory burst for sandboxes** NEW 91

Sandboxes in the `ali-ue1` region can now be given preauthorized memory burst at creation by setting `config.resources.memory_burst_percent`, allowing up to 50% additional memory (e.g. a 4 GiB base can expand to 6 GiB) without restarting processes. The feature is disabled by default (`resources.memory_burst_percent` defaults to 0), is limited to `ali-ue1`, and base plus burst must fit the platform maximum. Billing follows the effective memory limit at each point in time (4 GiB before expansion, 6 GiB while expanded, 4 GiB after shrinking), while CPU remains based on the original base memory specification.

— Names exact config key, region, limits and billing mechanics v0.10.6

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The Cyber Toolchain · curated daily · November 16, 2026
Releases & features from the open-source and commercial cyber tools that practitioners actually run.

▷ Subscribe Feedback on this issue