THE CYBER TOOLCHAIN

NO. 148

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED NOVEMBER 23, 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

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every tool in the issue, in full.

all 52 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 9 tools

▾ // THE BRIEF

HIDE SAME ISSUE, SAME PROMPT, TWO WRITERS: **GPT-5.5** Opus 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 today: nono can record privileged sessions with tamper-evident replay, Prowler can fix selected cloud

vim ? help

findings immediately after scan, and Velero can restore changed block data in place instead of moving whole volumes around.

GOVERN

Prove what happened in privileged sessions

Tamper-evident recording with replay and transcript turns ephemeral admin access into reviewable evidence. Teams can investigate or audit a session without relying on screenshots, shell history, or a human's recollection.

nono

PROTECT

Remediate selected cloud findings as part of the scan

The new `--fixer` flag lets specified checks apply remediation immediately after findings are produced. That removes the handoff from scan output to a separate fix run for issues the team already trusts Prowler to correct.

Prowler

RECOVER

Restore Kubernetes volumes with less data movement

Changed-block tracking and in-place volume restores make recovery less dependent on copying full volumes back into place. For large persistent volumes, that can turn restore work from a bulk rebuild into a narrower resumption path.

Velero

RESPOND

Recover deleted Windows event logs during forensic collection

The EVTX carving plugin gives responders a way to pull back event-log evidence that no longer exists as normal files. That helps preserve host timelines when an attacker or cleanup process has removed the obvious logs.

Velociraptor

Does **GPT-5.5** read better? 👍 👎

◆ AI-Powered Offensive Security

xalgorix

Feedback ⓘ v4.6.205 NOTES ↗ RANK

> Xalgorix cut its main agent system prompt nearly in half with opt-in phase playbook delivery, narrowed its bundled skill catalog from 865 to 330 skills focused on web and API testing, and made build_plan rebuilds preserve task progress and discovered endpoints instead of discarding them.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 Phase playbook delivery via XALGORIX_PHASE_PLAYBOOKS

Not checked yet NEW 95

Setting `XALGORIX_PHASE_PLAYBOOKS=true` (off by default, configurable in Runtime settings or `~/.xalgorix.env`) cuts the main agent system prompt from ~114 KB to ~59 KB (48%) by deferring each large phase body until a plan task belonging to that phase is active or queued. With it enabled, a phase's playbook also surfaces earlier during active scans — when the phase has an active task, when the engine sees the agent testing one of the phase's vulnerability classes, or when the phase is one of the two lowest phases with ready tasks.

Persist the phase-playbooks setting across restarts by storing it in the env file xalgorix reads at startup.

```ini
XALGORIX_PHASE_PLAYBOOKS=true
```

Persist the phase playbook opt-in across restarts by adding it to the xalgorix environment file.

```ini
XALGORIX_PHASE_PLAYBOOKS=true
```

— Exact env var, sizes, percentages, config file and trigger conditions given v4.6.203 v4.6.204

02 Plan progress and endpoint merging across build_plan rebuilds

Not checked yet IMPROVED 75

Plan progress now survives a `build_plan` rebuild: completed and skipped tasks with matching id, vulnerability class, and endpoint are kept, re-validated, and reported as kept, reopened, or dropped. Endpoints discovered via observed traffic, forms, and

JavaScript/API extraction are now merged rather than replaced when `build_plan` runs, preserving the full endpoint surface across rebuilds.

— Clear before/after mechanism but no user-triggered command v4.6.205

---

03 **Skill catalog narrowed to web and API testing** Not checked yet

75

BREAKING

The bundled skill catalog is narrowed from 865 to 330 skills (14 MB to 5.1 MB), dropping 29 categories unrelated to web and API testing — including SOC operations, digital forensics, OT/ICS, endpoint hardening, compliance, containers, and mobile — to reduce noise in `search_skills` results and binary size. The `ssrf` alias now loads the web SSRF skill, and `saml_attacks` loads the SAML skill in `read_skill`; loading any of the 535 removed skills by name now has `read_skill` report it as not found.

— Exact before/after sizes and category list but no migration step v4.6.204

---

04 **Source-review skills gated to source-attached scans**

Not checked yet IMPROVED

65

Source-review skills (SAST rules, secret and dependency scanning, CI/CD and supply-chain review) are now offered only to scans that have source code attached; `search_skills`, `list_skills`, and `read_skill` all suppress them for black-box scans.

— Describes gating logic across three functions, no explicit toggle v4.6.204

---

05 **Live reload for auto-delegate and checklist settings**

Not checked yet IMPROVED

60

Changing `XALGORIX_DISABLE_AUTO_DELEGATE` or `XALGORIX_CHECKLIST` no longer requires restarting the engine — saving either now applies to the next new scan immediately.

— Names both env vars but no deeper mechanism described v4.6.205

└──▷ **BREAKING ON UPGRADE**

! If you load a removed skill by name, `read_skill` now reports it as not found — any workflow that relied on the 535 removed non-web skills will break.

WAS THIS USEFUL? 👍 👎

APPLICATION SECURITY

◆ Web Application Security

Wapiti

Feedback ⓘ 3.3.4 NOTES ↗ RANK

Wapiti is a web vulnerability scanner written in Python3

> Wapiti 3.3.4 replaces its crawler's dependency on mitmproxy with a built-in intercepting proxy.

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

01 **Built-in intercepting proxy replaces mitmproxy** Not checked yet

45

BREAKING

The crawler now uses a built-in intercepting proxy instead of mitmproxy, removing the external dependency. Installations that relied on mitmproxy as a dependency may need to adjust their environment after upgrading.

— Names the change and migration note but no config detail 3.3.4

└──▷ **BREAKING ON UPGRADE**

! mitmproxy is no longer used by the crawler; installations that relied on mitmproxy as a dependency may need to adjust their environment after upgrading.

WAS THIS USEFUL? 👍 👎

Cure53 DOMPurify

Feedback ⓘ 3.5.0 NOTES ↗ RANK

DOMPurify - a DOM-only, super-fast, uber-tolerant XSS sanitizer for HTML, MathML and SVG. DOMPurify works with a secure default, but offers a lot of configurability and hooks. Demo:

> DOMPurify 3.5.0 focuses entirely on hardening against DOM clobbering and tightening the guarantees and fail-closed behaviour of several existing security options.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 **Custom elements rejected in SVG/MathML** Not checked yet BREAKING 73

Custom element names that pass `CUSTOM_ELEMENT_HANDLING.tagNameCheck` are now rejected when encountered inside SVG or MathML, where they are not valid custom elements; previously such elements were kept.

– Named config key and exact before/after behaviour 3.5.0

02 **IN_PLACE clobbering hardening and fail-closed throw**

Not checked yet BREAKING 72

Hardens `IN_PLACE` against DOM clobbering attacks, including neutralizing rejected roots inside open shadow roots and re-sanitizing nodes that hooks or custom elements reinsert. `IN_PLACE` now throws instead of looping indefinitely (or silently returning the node) when a hook or custom element keeps reinserting a removed node.

– Mechanism and before/after given, but no code sample 3.5.0

03 **SAFE_FOR_TEMPLATES guarantee narrowed for DOM input**

Not checked yet BREAKING 70

`SAFE_FOR_TEMPLATES` now guarantees template-expression safety only for string input; for DOM node and `IN_PLACE` input it is documented as best-effort, because a tree the HTML parser did not build can change shape on reparse. This is a reduction from the prior guarantee, which covered DOM node and `IN_PLACE` paths too.

– Named config flag with clear before/after scope 3.5.0

THINNER COVERAGE BELOW

04 **sanitize() hooks no longer mutate outer call config**

Not checked yet BREAKING 55

sanitize() calls made from inside a hook no longer change how the rest of the outer call behaves; code that relied on the old side effect must now pass config explicitly.

— Clear migration note but no named flag or snippet `3.5.0`

---

05 **DOM node sanitization strips invalid element children**

`Not checked yet` 40

`BREAKING`

When a DOM node is passed in, a `textarea` or `title` element holding an element child, and a void element holding children, are now removed during sanitization.

— States the change but no mechanism or rationale given `3.5.0`

└──▷ **BREAKING ON UPGRADE**

! `SAFE_FOR_TEMPLATES` is no longer guaranteed for DOM node input or `IN_PLACE` — it is now documented as best-effort for those paths.

! Custom element names passing `CUSTOM_ELEMENT_HANDLING.tagNameCheck` are now removed when encountered inside SVG or MathML, where they were previously kept.

! When a DOM node is passed in, a `textarea` or `title` holding an element child and a void element holding children are removed.

! sanitize() called from inside a hook no longer mutates the outer call's config — code that relied on that side effect must now pass config explicitly.

! `IN_PLACE` now throws when a hook or custom element keeps reinserting a removed node, instead of returning the node or looping indefinitely.

WAS THIS USEFUL? 👍 👎

REVERSE ENGINEERING, MALWARE & FORENSICS

◆ Digital Forensics & Incident Response

Velocidex Velociraptor

Feedback ⓘ v0.77.4 NOTES ↗

CODE ↗ RANK

Velociraptor is a digging Deeper....

> Velociraptor v0.77.4 adds an EVTX carving plugin for recovering deleted event logs, 11 new forensic and server artifacts, functional/bitwise VQL functions, and secrets-backed SFTP uploads, while removing PGP encryption support and an older artifact.

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

01 New forensic and server artifacts

Not checked yet NEW

95

Adds the carve_evtx() VQL plugin exposed via the new `Windows.Carving.Evtx` artifact (recovers deleted or anti-forensically cleared Windows event logs from raw devices or files via an `output=` path producing a recoverable `.evtx` file), along with new artifacts `Windows.Forensics.BitLocker` (BitLocker status, policies, and lifecycle events), `Windows.Forensics.BITS` (parses the BITS job database `qmgr.db`), `Windows.Forensics.MPLog` (Microsoft Defender support logs), `Windows.Forensics.Recall` (Windows 11 Copilot Recall databases), `Windows.Forensics.CBS` (Windows servicing CBS/DISM logs), `Windows.Applications.WinGet` (WinGet logs), `Windows.Registry.TaskCache` (parses Task Scheduler cache keys and flags tasks hidden by tampering), `Generic.Applications.Chrome.Cache` (Chromium HTTP disk cache), `Linux.Sysinternals.Autoruns` (Linux autostart entries), `Server.Utils.StorageAnalysis` (analyses the Velociraptor data store), and `Server.Utils.ExportFlow` (exports a completed flow to a ZIP file).

Recover deleted or anti-forensically cleared event log entries directly from a live disk volume during incident response.

```
$ SELECT * FROM carve_evtx(filename='\\\\.\\C:',
accessor='raw_file') WHERE System.Channel = 'Security'
```

— Named plugin, flag, artifact, and a runnable query example All eleven artifacts named with purpose, no usage shown

v0.77.4

02 PGP encryption removed; FailedLogBeforeSuccess artifact dropped

Not checked yet BREAKING 70

pk_encrypt(), pk_decrypt(), `Server.Utils.CreateCollector`, and the offline collector no longer support PGP encryption; use X509 or a password instead. The artifact `Windows.Events.FailedLogBeforeSuccess` has been removed.

— Names affected functions/artifact and migration path v0.77.4

03 **Size and recency filters in FileFinder** Not checked yet IMPROVED 68

`Windows.Search.FileFinder` gains optional `MaxFileSize` and `ModifiedWithinDays` parameters to narrow file searches by size or recency; both are disabled by default.

— Named config parameters but no usage example given v0.77.4

04 **Mode parameter and new columns in RecycleBin artifact**

Not checked yet IMPROVED 68

`Windows.Forensics.RecycleBin` adds a `Mode` parameter (default `API`) selecting MFT-based recovery of emptied `$I` records, plus new `InUse` and `MFTId` output columns.

— Names parameter and columns but no run example v0.77.4

05 **Functional and bitwise VQL functions** Not checked yet NEW 64

Adds VQL functions `map`, `reduce`, `reverse`, and `bit_*` for functional and bitwise operations on query results, plus the ability to expose the byte size of a materialized `LET` member for memory-aware query design.

— Names several functions but no usage example v0.77.4

THINNER COVERAGE BELOW

06 **FlowId and ClientId parameters for SaveFavoriteFlow**

Not checked yet IMPROVED 55

`Server.Utils.SaveFavoriteFlow` now accepts `FlowId` and `ClientId` parameters so a favorite can be saved directly from an existing flow.

— Named parameters give a clear starting point v0.77.4

07 **Experimental distributed execution: redirected org and minion mode**

Not checked yet NEW 50

Adds experimental distributed execution modes: switching execution into a redirected org via the query() plugin, and a minion mode for running server artifacts on a remote

server.

— Names mechanism but no concrete syntax or config v0.77.4

08 **Secrets store authentication for upload_sftp** Not checked yet

IMPROVED 40

`upload_sftp` calls can now authenticate using Velociraptor's secrets store instead of inline credentials.

— Names the function but gives no setup detail v0.77.4

09 **macOS unified logs parser component** Not checked yet NEW

30

New macOS unified logs parser added as a VQL-level component; no artifact exposes it yet.

— Bare mention with no artifact or usage yet v0.77.4

└──▷ **BREAKING ON UPGRADE**

! pk_encrypt(), pk_decrypt(), `Server.Utils.CreateCollector`, and the offline collector no longer support PGP encryption; use X509 or a password instead.

! The artifact `Windows.Events.FailedLogBeforeSuccess` has been removed.

WAS THIS USEFUL? 👍 👎

NETWORK SECURITY

◆ Email Security

Rspamd

Feedback ⓘ 4.3.1 NOTES ↗ RANK

Rspamd is a rapid spam filtering system.

> Rspamd 4.3.x adds a new dynconf framework that lets scores, actions and plugin options be overridden at runtime via Redis or file backends, alongside ARC/SPF/phishing-scoring security hardening, new Lua APIs for ASCII85 decoding, SMTPUTF8 validation and URL redirect/query-target traversal, TLS-enabled Redis connections, a Redis-list metadata_exporter backend, per-scan symbol profiling with ClickHouse storage, and new RBL and DKIM tuning options.

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

01 dynconf runtime-override framework

Not checked yet BREAKING

100

A new dynconf framework lets scalar options of `chartable`, `dkim`, and `fuzzy_check`, plus runtime options of 41 Lua plugins, be overridden at runtime via Redis or file backends, managed through `rspamadm configdump --dynamic` and new controller endpoints `dump`, `set`, `reset`, and `reload`. Dynamic scores and actions move to dynconf: the legacy dynamic configuration file and Redis hash are imported once and the file renamed afterwards, and the `dynamic_conf` plugin is now a stub for the dynconf Redis backend.

Inspect all current runtime overrides — scores, actions, and plugin options — without restarting Rspamd

```
$ rspamadm configdump --dynamic
```

— Exact command and endpoints with migration behaviour 4.3.1

02 metadata_exporter Redis list backend and unix_time field

Not checked yet NEW

95

The `redis_list` backend (4.2.0+) exports message metadata to a Redis list, configured with `list_key` to name the list and an optional maximum-length setting that issues `LTRIM` after each `RPUSH` to drop the oldest entries when consumers fall behind; `unix_time` (4.2.3+) adds the message receipt time as a fractional-second GMT Unix timestamp to exported metadata.

Buffer outbound message metadata into a Redis list for a Logstash redis input plugin to consume with BLPOP, capping the list at 10 000 entries so it never grows unbounded.

```yaml
redis_list {
  list_key = "rspamd_metadata";
  max_length = 10000;
}
```

Include the message receipt time as a Unix timestamp in every exported record so downstream consumers can correlate events by exact arrival time.

```yaml
rules {
  MY_EXPORT {
    backend = "redis_list";
    list_key = "rspamd_metadata";
    fields = ["unix_time", "from", "rcpt", "subject", "action"];
  }
}
```

— Named config keys with runnable config file examples `product docs`

03 **New RBL tuning options** `Not checked yet` NEW 90

New RBL options: `hash_format = "rbase32"` enables RFC 4648 base32 encoding for RBL lookups, `selector_hash = false` lets selectors that already return a hash skip re-hashing, and `max_dns_requests` caps DNS requests per message per rule, adding a `<SYMBOL>_LIMIT` symbol with a zero score when work is dropped.

Cap RBL DNS traffic per message and avoid scoring noise from dropped work, using an RBL rule that already produces hashed selectors

```yaml
rbl {
  rule "MY_RBL" {
    rbl = "rbl.example.com";
    hash_format = "rbase32";
    selector_hash = false;
    max_dns_requests = 10;
  }
}
```

— Named config options with a runnable RBL rule example `4.3.1`

04 **Symbol options stored verbatim, selectively percent-encoded**

`Not checked yet` 85

BREAKING

Rspamd now stores symbol options verbatim, encoding only invalid characters as U+FFFD, so regexps, composites, and multimap see original values; outputs that join options (`X-Spamd-Result` , `X-Virus` , `metadata_exporter` , ClickHouse, and the symbols log line) now percent-encode only when an option contains `[]{};,` , also encoding `%` for unambiguous decoding. Options previously joined as comma-separated values by asn, reputation, rbl, dmarc, content rules, and `MULTIPLE_UNIQUE_HEADERS` are now inserted as separate options, which may require changes in rules or external tools parsing the old joined format.

— Names every affected output field, but no migration command product docs

05 **TLS connection table for Redis Lua API** Not checked yet NEW 85

Redis Lua API calls now accept a `connection` table with TLS fields `ssl` , `no_ssl_verify` , `ssl_ca` , `ssl_ca_dir` , `ssl_cert` , `ssl_key` , and `sni` ; when present, this table is authoritative over legacy flat connection fields.

Connect to Redis over TLS in a custom Lua plugin, supplying certificates and disabling hostname verification for a self-signed setup.

lua

```lua
local redis_params = {
  host = 'redis.example.com:6380',
  connection = {
    ssl = true,
    no_ssl_verify = false,
    ssl_ca = '/etc/ssl/certs/ca.pem',
    ssl_cert = '/etc/rspamd/redis-client.crt',
    ssl_key  = '/etc/rspamd/redis-client.key',
    sni      = 'redis.example.com'
  }
}
rspamd_redis.make_request(task, redis_params, callback, 'GET',
{'mykey'})
```

— Exact field names plus a runnable TLS connection example product docs

06 **DKIM extra_aligned_sigs option** Not checked yet NEW 80

New DKIM option `extra_aligned_sigs` (default 2) checks additional signatures aligned with From beyond the `max_sigs` limit, adding `DKIM_SIGS_TRUNCATED` when signatures are skipped.

Allow up to 4 From-aligned DKIM signatures to be verified beyond the normal max_sigs limit, so high-volume senders with multiple valid signatures are not silently truncated

```yaml
dkim {
  extra_aligned_sigs = 4;
}
```

— Named option, default, and symbol with runnable config example `4.3.1`

07 **New Lua utility functions: ASCII85 decode and SMTPUTF8 validation**

Not checked yet NEW 75

util.decode_ascii85(input[, size_limit]) decodes ASCII85-encoded data in Lua rules, with an optional `size_limit` to cap decoded output and a nil-plus-error return on failure; util.is_valid_smtputf8_address() validates SMTPUTF8 local-part/domain pairs against PRECIS and script-mixing rules.

Block messages that embed ASCII85-encoded payloads larger than 1 MB, to avoid decompression bombs in custom rules.

```lua
local util = require 'rspamd_util'
local decoded, err = util.decode_ascii85(part_content, 1048576)
if not decoded then
  rspamd_logger.infox(task, 'ascii85 decode failed: %s', err)
  return
end
```

Reject mail from SMTPUTF8 senders whose local part fails PRECIS or script-mixing rules before passing the message further.

```lua
local util = require 'rspamd_util'
local ok, rule = util.is_valid_smtputf8_address(local_part,
domain, false)
if not ok then
  task:set_pre_result('reject', string.format('invalid SMTPUTF8
address: %s', rule))
end
```

— Exact function signatures with runnable Lua examples `product docs`

08 **Per-scan symbol profiling with ClickHouse storage** Not checked yet

NEW 70

Read per-scan symbol timings through rspamd_config:register_profiled_symbols() and task:get_symbols_profile() using the new `lua_task_profile` library; the ClickHouse plugin now stores these per-scan symbol profile samples in new Profile columns (schema version 13).

— Named APIs and ClickHouse columns, no usage example given 4.3.1

09 **URL-mismatch scoring split from phishing scoring** Not checked yet

65

BREAKING

Rspamd now scores URL mismatches (where the displayed link differs from the real destination) separately from phishing scoring; link rewriting by trusted services uses explicit trust maps, and the stock configuration loads rewrite permissions from dynamic maps with conservative rollout scores. Local scores and any rules built on phishing symbols should be reviewed after upgrade.

— Mechanism described but no concrete map name or score value product docs

10 **ARC verification hardened (IMPROVED)** Not checked yet IMPROVED

65

A whitelisted forwarder is now trusted only when every later ARC instance is also whitelisted and verified, the newest ARC-Message-Signature must verify independently, and a DMARC adjustment requires the forwarder's own `dmarc=pass` with a `header.from` matching the From domain; additionally, Rspamd now parses ARC-Authentication-Results from the raw header value rather than after decoding encoded-words, preventing a crafted encoded-word in a comment from injecting a `dmarc=pass` result.

— Detailed trust mechanism but no setting to configure Security fix explained, nothing actionable for operators

product docs

11 **URL Lua API gains redirect and query-target traversal**

Not checked yet NEW

65

url:get_query_target(pool) follows an embedded HTTP/HTTPS/FTP URL through query parameters to its leaf, returning `{url|nil}` and stopping on ambiguity or depth exhaustion; url:get_redirected() retrieves a URL's redirect target or `nil` if not redirected; url:is_phished() returns the phished URL or `nil` when the URL is not phished or is redirected.

— Named methods with return semantics, no example code given product docs

12 **Strict time interval parsing in lua_util** Not checked yet

BREAKING

65

`lua_util` now parses time intervals strictly: `'500ms'` means milliseconds, `'1 d'` and `'1D'` both mean a day, and unknown units or trailing text are rejected.

— Exact parsing rules named, no migration command given `4.3.1`

THINNER COVERAGE BELOW

13 **Bare ESMTP keywords preserved in headers** Not checked yet IMPROVED

55

Rspamd now keeps bare ESMTP keywords without a value (such as `SMTPUTF8` or `REQUIRETLS`) in `X-Rspamd-Rcpt-Esmtp-Args` headers and in `rcpt_esmtp_args` in the v3 protocol, where they were previously dropped.

— Named fields but no action needed beyond upgrade `product docs`

14 **SPF term evaluation order fix** Not checked yet IMPROVED

50

Rspamd now evaluates SPF record terms in their original order rather than a broken sort order, so the first matching term wins as the RFC requires; qualifiers inside `include:` directives are now applied correctly.

— Clear fix described, nothing for reader to configure `product docs`

15 **Version output shows release base and git distance**

Not checked yet IMPROVED

50

`--version`, rspamd_version(), and the controller stat reply now show the release base and git distance from it.

— Named surfaces but minor, no example given `4.3.1`

16 **URL hash sets randomly seeded per process** Not checked yet

30

BREAKING

URL hash sets are seeded with a random value per process, so unsorted task URLs come out in a different order in each process.

— Brief behavioural note with no configuration surface `4.3.1`

└──▷ **BREAKING ON UPGRADE**

! Symbol options that were previously joined as comma-separated values by asn, reputation, rbl, dmarc, content rules, and `MULTIPLE_UNIQUE_HEADERS` are now inserted as separate options; rules and external tools that parse these joined options or the joined headers (`X-Spamd-Result`, `X-Virus`, metadata_exporter, ClickHouse, symbols log line) may need changes.

! URL-mismatch scoring is now reported separately from phishing scoring; review local scores and any rules that build on the phishing symbols.

! Dynamic scores and actions are moved to dynconf: the legacy dynamic configuration file and Redis hash are imported once and the file is renamed afterwards; the `dynamic_conf` plugin is now a stub for the dynconf Redis backend.

! `lua_util` now parses time intervals strictly: `'500ms'` means milliseconds, `'1 d'` and `'1D'` both mean a day, and unknown units or trailing text are rejected.

! URL hash sets are seeded with a random value per process, so unsorted task URLs come out in a different order in each process.

WAS THIS USEFUL?  

CLOUD & CONTAINER SECURITY

◆ CNAPP

Plerion

Feedback ⓘ See everything past end of life in one place:

NOTES ↗ RANK

Plerion is a cloud security platform that provides risk-based prioritization and remediation guidance across cloud infrastructure, helping teams identify and address critical vulnerabilities efficiently.

> Plerion introduced end-of-life tracking as a unified asset attribute, surfacing retirement status across AWS, Azure, Google Cloud, and Kubernetes resources and making it queryable through natural language.

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

01 **End-of-life coverage across AWS, Azure, GCP, Kubernetes**

Not checked yet NEW 75

Plerion now tracks end-of-life status as a filterable attribute usable in findings filters, workflows, and custom reports. Coverage includes AWS EKS clusters and node groups, RDS and Aurora databases, Lightsail databases, ElastiCache, OpenSearch, DocumentDB, Neptune, MSK, Amazon MQ, Glue jobs, EMR clusters, Fargate platform versions, CodeBuild images, and Bedrock agents and custom models built on retired foundation models; Azure Database for PostgreSQL and MySQL, App Service, Service Fabric, Azure Red Hat OpenShift, virtual machine scale sets, Batch pools, and SQL Server on Azure virtual machines; Google Cloud SQL, AlloyDB, Dataproc, Cloud Composer, Cloud Run, and Compute Engine instance templates and images; and Kubernetes nodes running a kubelet past upstream end of life. Users can also ask Pleri which assets are past end of life using natural language queries.

– Lists many named services and filter usage but no exact API or flag

See everything past end of life in one …

WAS THIS USEFUL? 👍 👎

DETECTION & RESPONSE

◆ Endpoint Security (EDR/XDR)

Trend Vision One

Feedback ⓘ

TrendAI Vision One™ Container Security for Amazon ECS version 2.2.15 (2026-11-13)

BLOG ↗ RANK

Trend Vision One is a unified cybersecurity platform that provides threat detection, investigation, and response across endpoints, networks, email, and cloud environments.

> Trend Vision One's AI Scanner gained multi-scan aggregation and finding-level review, correction, and deletion controls, while Container Security for Amazon ECS picked up a configurable scan image, SBOM spec pinning, and automatic rescans on policy changes.

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

01 **Review, correct, and delete AI Scanner results** Not checked yet

NEW 88

In the AI Scanner results edit drawer, findings can be marked as false positives or false negatives, with the Attack outcome column showing the correction alongside the original verdict in parentheses and a filter to sort by the new types. Severity can also be overridden in the edit drawer when the assigned severity is wrong, and corrections propagate into attack success rate, severity counts, results over time, compliance tiles, and every export format. A whole scan and its results can be removed with Delete Scan at the end of the scan row after confirmation. Access is gated by the AI Scanner Modify permission for editing findings and renaming, and the Delete permission for scan deletion, under AI Security → AI Application Security.

Override a scanner verdict to false positive so a known-safe probe no longer inflates your attack success rate.

📍 In AI Security → AI Application Security, open the AI Scanner results for your scan. Click the edit button in the finding row, select False positive in the edit drawer, and save. The Attack outcome column will show the correction with the original verdict in parentheses.

Remove a completed scan and all its data to keep your results view clean after a test campaign ends.

📍 In AI Security → AI Application Security, locate the scan row and click Delete Scan at the end of the row, then confirm the deletion. Requires the Delete permission.

— Exact UI steps, button names, and permission names given

Review, correct, and delete TrendAI Vis…

02 **Aggregate scan results across multiple scans in AI Scanner**

Not checked yet NEW 66

Trend Vision One AI Scanner now aggregates scan results across multiple scans of a target instead of showing per-scan summary tiles. Custom Time Range aggregation includes every completed scan that started within a specified date range, with the selector, tiles, and results table reflecting that range; Custom Selection aggregation lets you hand-pick individual completed scans from a target's scan list, each contributing its full set of results.

Aggregate AI scan results over a sprint or investigation window to see combined findings across all scans run during that period.

> 📍 In the console, go to AI Security → AI Application Security, select your target, open the aggregation selector, choose Custom Time Range, and set the start and end dates to cover the scans you want.

Compare findings from a specific subset of scans — for example, only pre- and post-remediation runs — without including unrelated scans in the summary tiles.

> 📍 In the console, go to AI Security → AI Application Security, select your target, open the aggregation selector, choose Custom Selection, and tick each individual completed scan you want to include.

– Clear UI navigation but no config or API surface named

Aggregate scan results across multiple …

03 **ECS container security scanning and SBOM updates** Not checked yet

IMPROVED 65

Container Security for Amazon ECS added `CloudScanJobImage` to the ECS protection stack parameters to control which image is used for cloud scan jobs. ECS CycloneDX SBOM output is now pinned to specification version 1.6. Modifying `fileScan` paths or the `imageScan` filter in the malware scan policy now triggers an automatic rescan. File lists are now excluded from Syft SBOM output to reduce SBOM size and storage cost.

– Names four config/parameter surfaces but no usage example

TrendAI Vision One™ Container Security …

WAS THIS USEFUL? 👍 👎

◆ Backup & Recovery

Velero

Feedback ⓘ v1.19.0 NOTES ↗ RANK

Velero is an open-source tool that backs up, restores, and migrates Kubernetes cluster resources and persistent volumes.

> Velero v1.19.0 adds block-level data movement with changed-block tracking, in-place volume restores, and ConfigMap-driven filtering for backups and restores, alongside several breaking changes to module layout, Kopia hashing, and credential handling.

└──▷ **WHAT SHIPPED · 10 FEATURES** most completely described first ? what's the number?

01 Block data mover for CSI snapshots

Not checked yet NEW

85

A new block data mover (`velero-block`) handles CSI snapshot data movement, enabling block-level backup and restore. It integrates with the Kubernetes CSI Changed Block Tracking (CBT) API to transfer only changed blocks during incremental operations.

— Names component and CBT mechanism but no sample command v1.19.0

02 In-place volume restore and existing-volume restore safeguards

Not checked yet NEW

80

Backup data from `velero-fs` , `velero-block` , or File System Backup can now be restored directly into existing volumes instead of new ones, with incremental restores skipping unchanged data; a related node-agent option, `restorePVC.ignoreDelayBinding: true` , works around stalled restores when a pod mounts both Immediate and WaitForFirstConsumer PVCs on a single-zone cluster. A new `AllowExistingVolumeMismatch` restore configuration setting also bypasses the same-volume-handle pre-flight check when restoring into existing volumes.

Work around a stalled restore when a pod mounts both Immediate and WaitForFirstConsumer PVCs on a single-zone cluster by setting the node-agent option to ignore delayed binding.

yaml

```yaml
restorePVC:
  ignoreDelayBinding: true
```

— Includes a concrete config example a reader can apply Exact flag named, usage context given but no sample

v1.19.0

03 **BackupStorageLocation credentials must use Secrets**

Not checked yet BREAKING 75

User-provided `spec.config.credentialsFile` values in BackupStorageLocation (BSL) and VolumeSnapshotLocation (VSL) are now ignored; credentials must be stored in a Kubernetes Secret and referenced via `spec.credential.name` and `spec.credential.key`.

– Exact fields named, clear migration step to Secret reference v1.19.0

04 **Fine-grained filtering and namespace label selection via ResourcePolicy ConfigMaps**

Not checked yet NEW 70

ResourcePolicy ConfigMaps now support fine-grained backup and restore filters with per-namespace and per-resource-kind rules, each with independent label selectors and resource name patterns. The same ConfigMap mechanism also allows namespace selection by labels, dynamically including or excluding namespaces for backups without maintaining explicit namespace lists.

– Names the ConfigMap mechanism but no exact key/field shown v1.19.0

05 **Go module path moved to velero-io org** Not checked yet BREAKING 70

The Go module path changed from `github.com/vmware-tanzu/velero` to `github.com/velero-io/velero`; downstream projects importing Velero as a library must update their imports and `go.mod`.

– Exact old and new import paths given, clear migration step v1.19.0

06 **Restic path for file system backup removed** Not checked yet 65

BREAKING

The Restic path for fs-backup is completely removed: restores from backups using the Restic path are marked `FailedValidation`, backup sync ignores backups with the Restic path, and deletion requests for existing Restic-path backups in the cluster will fail.

– Specific failure states named but no migration path given v1.19.0

07 **Volume policy enhancements** Not checked yet NEW 60

Volume policies gain a `custom` action type, letting third-party BackupItemAction plugins handle volume backups with plugin-specific parameters, alongside the existing snapshot and file system backup policy types; administrators can also now set default

volume backup rules for all backups through a shared global volume policy ConfigMap, with backup-specific policies taking precedence over the global defaults.

— Names the new action type but no plugin example given Describes behaviour but no concrete key or example

v1.19.0

THINNER COVERAGE BELOW

08 **Default prepare queue length for data movers** Not checked yet

55

BREAKING

A default prepare queue length of 5 is now enforced for data movers; previously there was no limit, so clusters relying on unlimited concurrency will now be capped.

— States the new limit but no config key to adjust it v1.19.0

09 **Kopia repository hashing algorithm changed** Not checked yet

55

BREAKING

The Kopia repository hashing algorithm changed from `BLAKE2B-256-128` to `HMAC-SHA256-128` for new repositories.

— Names old and new algorithms but no migration guidance v1.19.0

10 **Default restore resource modifier via shared ConfigMap**

Not checked yet NEW

50

A shared ConfigMap can now define a default restore resource modifier that automatically applies common resource transformations during restores; individual restores can override or skip it.

— Behaviour described but no concrete config example v1.19.0

└──▷ BREAKING ON UPGRADE

! The Go module path changed from `github.com/vmware-tanzu/velero` to `github.com/velero-io/velero`; downstream projects importing Velero as a library must update their imports and `go.mod`.

! Restic path for fs-backup is completely removed; restores from backups with the Restic path are marked `FailedValidation`, backup sync ignores backups with Restic path, and deletion requests for existing Restic-path backups in the cluster will fail.

! A default prepare queue length of 5 is now enforced for data movers; previously there was no limit, so clusters relying on unlimited concurrency will now be capped.

! Kopia repository hashing algorithm changed from `BLAKE2B-256-128` to `HMAC-SHA256-128` for new repositories.

! User-provided `spec.config.credentialsFile` values in BackupStorageLocation (BSL) and VolumeSnapshotLocation (VSL) are now ignored; credentials must be stored in a

Kubernetes Secret and referenced via `spec.credential.name` and `spec.credential.key`.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

ongrid

Feedback ⓘ v0.22.0 NOTES ↗ RANK

Ongrid is an ops AI Agent that understands your infrastructure, finds the root cause, and fixes it — right from Slack, Telegram, Lark or DingTalk.

> Ongrid improved its AI ops chat with context window management for long sessions and added visibility into worker tool steps during follow-ups.

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

01 **Chat context window management with compacted history recovery**

Not checked yet IMPROVED 33

Ongrid's AI ops chat now manages context windows and recovers compacted history to maintain continuity across long sessions.

— Describes behavior but no mechanism or limits given v0.22.0

02 **Worker tool step visibility during follow-ups** Not checked yet

NEW 25

Ongrid's chat now shows worker tool steps during and after follow-up messages.

— States what changed but no detail on how it's shown v0.22.0

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

▷ Subscribe Feedback on this issue