THE CYBER TOOLCHAIN NO. 001

Head

THE WEEK'S TOP PICKS

PUBLISHED JULY 13, 2026 · FRIDAYS

// SHARE THIS WEEK ▽ Top 5 ▽ Top Pick

EDITIONS tail | grep | **head** | diff | uniq

The best tooling updates that shipped this week.

> // HOW THESE PICKS ARE MADE
>
> Every feature release from the 874 tools on our watchlist goes into the daily newsletter. Once a week we read the whole field side by side and pick the few updates worth your attention, on two questions: **how deep and complete is the single best capability in the release**, and **how much it changes what you can actually do**.
>
> A tool is judged on everything it shipped that week, so a project that releases daily gets credit for the sum — and still only takes one slot. We would rather run a short list than a padded one.

// THE PICKS

01	**Dalfox**	Web Application Security
02	**trufflehog**	Secrets Detection & Management
03	**Headscale**	Secure Access (VPN / SASE / ZTNA)
04	**xsrfprobe**	Web Application Security
05	**ssh-audit**	Vulnerability Management

01 Dalfox

2 RELEASES · 2026-06-14 → 2026-06-21

Web Application Security

Dalfox is a powerful open-source XSS scanner and utility focused on automation.

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 5/5

Dalfox v3.1.0 ships a genuinely comprehensive XSS toolkit upgrade in one release: blind/OOB detection via interactsh, HAR file input from any proxy tool, AST-based external JS analysis, CSP/Trusted Types payload generation, WAF inspection-window bypass, four new attribute-decode mutations, and a global rate limiter — each of these alone would be noteworthy, together they make Dalfox a materially different tool for web-app testers than it was a week ago.

> Dalfox v3.1.0 adds out-of-band XSS detection via interactsh, HAR-based scanning, external JS/DOM analysis, CSP & Trusted Types aware payloads, several WAF-evasion techniques, and a global rate limiter, while v3.1.1 unifies the server/MCP `target` parameter, consolidates debug logging, and bounds request fan-out for safer large scans.

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

01 HAR file input for scans NEW

85

Adds `--input-type har` to accept HAR exports from Burp, Caido, ZAP, mitmproxy, or DevTools as a scan source; the `pipe` subcommand also accepts an explicit `-i har` input type for piping HAR captures directly into a scan.

Replay every request captured in a proxy session (Burp/Caido/ZAP) as a scan source, feeding all recorded targets at once.

```
$ dalfox scan export.har --input-type har --output json -o
results.json
```

Pipe a HAR capture through dalfox using an explicit input type — now correctly honoured by the `pipe` subcommand after the parity fix.

```
$ cat capture.har | dalfox pipe -i har --output results.json
```

— Names supported tools, flag and subcommand with runnable examples v3.1.0 v3.1.1

02 External JS bundle analysis NEW

85

Adds `--analyze-external-js` to fetch and run AST DOM-XSS analysis on a target's same-origin `<script src>` bundles, capped at 16 files / 512 KiB.

Cap aggregate outbound rate to 20 RPS across all workers while also pulling in external JS bundles for DOM-XSS analysis — useful for rate-sensitive bug-bounty targets.

```
$ dalfox scan https://target.example.com --rate-limit 20 --analyze-external-js
```

— Names flag, mechanism, and concrete size/file caps v3.1.0

03 **Global request rate limiting** NEW 80

Adds global rate limiting via `--rate-limit` / `-r` / `--rl` (requests-per-second token bucket, `0` = unlimited) shared across all workers and targets.

Cap aggregate outbound rate to 20 RPS across all workers while also pulling in external JS bundles for DOM-XSS analysis — useful for rate-sensitive bug-bounty targets.

```
$ dalfox scan https://target.example.com --rate-limit 20 --analyze-external-js
```

— Named flags, algorithm, default, and a runnable example v3.1.0

04 **Blind/out-of-band XSS detection** NEW 70

Adds `--blind-oob` for out-of-band (OAST) XSS detection via interactsh, catching stored and async sinks that never reflect back to the response.

Detect stored/async XSS in a target that never reflects payloads back — catch blind execution via an interactsh OOB callback.

```
$ dalfox scan https://target.example.com --blind-oob
```

— Names flag and mechanism but not callback verification details v3.1.0

05 **Retry handling for transient errors** NEW 70

Adds `--retries` / `--retry-delay` for optional exponential-backoff retries on HTTP 5xx and transient errors (HTTP 429 is always retried).

— Flags and backoff behavior named, no example command v3.1.0

06 **Unified target parameter for server/MCP** IMPROVED 70

Server and MCP modes now accept a unified `target` parameter for scan targets, while the REST API retains `url` as a backward-compatible alias.

Debug a scan against a REST API server using the unified `target` parameter (replacing the old `url` field for server/MCP calls).

```
$ curl -s -X POST http://localhost:6664/scan -H 'Content-Type: application/json' -d '{"target": "https://example.com/search?q=test"}'
```

— Named field change with a runnable API example `v3.1.1`

---

07 **CSP and Trusted Types aware payloads** NEW 65

Adds CSP & Trusted Types awareness: emits `strict-dynamic` / nonce gadget payloads and adapts payload generation when a Trusted Types policy is detected on the target.

— Mechanism named but no flag to invoke it manually `v3.1.0`

---

08 **Attribute-decode WAF-bypass mutations** NEW 65

Adds four attribute-decode WAF-bypass mutations — `KeywordEntityEncode`, `SchemeBreak`, `EntityScheme`, `MultiSlash` — exploiting HTML tokenizer entity decoding to evade WAF filters.

— All four mutation names given but no user-facing toggle `v3.1.0`

---

09 **Outdated front-end library detection** NEW 60

Adds `--detect-outdated-libs` (opt-in) to flag known-vulnerable front-end library versions as informational findings.

— Flag named but no detail on detection source or library list `v3.1.0`

---

10 **Structured metadata in output formats** IMPROVED 60

Structured output metadata — version, targets, duration, request/finding counts, per-target WAF info — is now included in SARIF, Markdown, and TOML output formats.

— Names formats and fields but no flag or output sample `v3.1.0`

---

11 **Logging and output consistency improvements** IMPROVED 60

Consolidates all debug output through a single stderr `dbg_log!` macro, keeping JSON/JSONL/SARIF stdout clean under `--debug`, and aligns OOB/blind XSS lifecycle messages with the standard timestamped log format (`INF` / `WRN` / `OOB`) for consistency across scan modes.

— Names macro and log tags but no direct user action `v3.1.1`

---

12 **Scan reliability and request-fan-out bounds** IMPROVED

60

Adds a per-parameter payload safety cap and a recall-preserving DOM-phase early-exit to bound request fan-out and prevent runaway HTTP requests on wide parameter sets; also bounds previously-unbounded task spawning in parameter mining and cuts server hot-path lock-hold and allocations to improve reliability under load.

— Mechanism and scope described but no user-facing control v3.1.1

THINNER COVERAGE BELOW

13 **Filter-aware payload synthesis** NEW

55

Adds filter-aware payload synthesis that computes exact JS breakout sequences from the observed script prefix, including escaped-quote and nested-context cases.

— Mechanism described but no control surface to trigger it v3.1.0

14 **WAF inspection-window bypass** NEW

55

Adds size-limited WAF inspection-window bypass: detects WAFs that inspect only the first N bytes of a request and positions payloads past that inspection window to evade detection.

— Mechanism explained but automatic, no configurable surface named v3.1.0

15 **Insecure TLS mode** NEW

55

Adds `--insecure` TLS flag to make certificate validation configurable for `scan`, `server`, and `mcp` modes.

— Flag and applicable modes named, minimal further detail v3.1.0

16 **Whole-scan timeout budget** NEW

55

Adds `scan_timeout` whole-scan budget for `server` and `mcp` modes to bound total duration of concurrent async jobs.

— Config key named but no example or default value given v3.1.0

17 **Additional DOM-XSS sink coverage** IMPROVED

30

Extends DOM-XSS sink coverage to include Document.parseHTMLUnsafe() and window.open().

— Sink names given but no mechanism or usage detail v3.1.0

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

02 trufflehog

1 RELEASE · 2026-07-09

Secrets Detection & Management

Find, verify, and analyze leaked credentials

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 4/5

TruffleHog adds seven new validated secret detectors (OpenRouter, New Relic, Duffel, Shippo, IPinfo, Lob, HashiCorp Vault batch tokens) and an --exclude-archived flag for org-wide GitHub scans — a well-rounded drop-in upgrade for any secrets-scanning pipeline this week.

> TruffleHog v3.95.9 expands its default detector set with seven new secret detectors and adds an option to skip archived repositories during GitHub org scans.

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

01 Exclude archived repos from GitHub org scans NEW

70

Adds `--exclude-archived` flag to skip archived repositories when scanning a GitHub org, reducing noise and scan time in large orgs.

Scan a GitHub org while skipping archived repos to reduce noise and scan time in large orgs.

```
$ trufflehog github --org=<your-org> --results=verified --exclude-archived
```

— Exact flag and runnable command provided v3.95.9

02 Seven new secret detectors added NEW

65

Adds detectors for OpenRouter (`sk-or-v1-` prefixed keys, verified against the OpenRouter API), New Relic Insights Insert Key (`NRII-` prefixed keys, verified via the Events API), Duffel Token (Duffel travel API), Shippo (Shippo shipping API keys), IPinfo (now included in the default detector set), an improved Lob detector (also now in the default detector set), and HashiCorp Vault batch tokens.

— Names all seven detectors and prefixes but no run commands v3.95.9

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

03 Headscale

1 RELEASE · 2026-06-17

Secure Access (VPN / SASE / ZTNA)

An open source, self-hosted implementation of the Tailscale control server

// WHY IT MADE THE LIST

DEPTH 5/5 IMPACT 5/5

Headscale v0.29.0 brings SSH check-mode auth (OIDC or CLI approval before SSH access), policy test blocks (including sshTests), Tailscale Grants for Taildrive and traffic steering, nodeAttrs capability distribution, and HA subnet router health probing — making a self-hosted Tailscale control plane meaningfully closer to feature parity with the SaaS offering for practitioners who rely on it.

> Headscale v0.29.0 adds SSH check-mode approval via a new `headscale auth` CLI, beta policy testing with `tests` / `sshTests` blocks, and new `grants` / `nodeAttrs` policy blocks for Taildrive and node capability distribution — alongside a run of breaking changes to ACL semantics, the upgrade path, and server config.

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

01 SSH check-mode approval with `headscale auth` CLI NEW

88

A new `headscale auth` CLI command group adds `approve`, `reject`, and `register` subcommands to manage SSH check-mode and web auth approval flows. Headscale now supports Tailscale SSH `"action": "check"` rules, prompting OIDC or CLI approval before granting SSH access to a node, e.g. `headscale auth approve --auth-id <id>`.

Approve a pending SSH check-mode or web auth request identified in server logs, granting the user access without re-keying.

```
$ headscale auth approve --auth-id <id>
```

— Named CLI subcommands and a runnable example command. v0.29.0

02 Policy tests and sshTests blocks (beta) NEW

88

Headscale evaluates a `tests` block in policy files (beta) asserting reachability between sources and destinations, and an `sshTests` block (beta) asserting SSH reachability with `accept`, `deny`, and `check` user lists per source/destination pair; failing tests block a policy write before it is applied. Policies can be validated with `headscale policy check`.

Validate a policy file's `tests` and `sshTests` blocks before pushing it live, catching reachability regressions without affecting running nodes.

```
$ headscale policy check
```

— Named policy blocks and a validation command. v0.29.0

03 **Wildcard ACL semantics replaced by autogroup:danger-all** BREAKING

80

Wildcard (`*`) in ACL sources and destinations now resolves to the Tailscale CGNAT range (`100.64.0.0/10`) and ULA range (`fd7a:115c:a1e0::/48`) instead of all IPs. Policies relying on `*` to match every address must switch to the new source-only `autogroup:danger-all` target, which resolves to `0.0.0.0/0` and `::/0`, or use explicit CIDR ranges for destinations.

— Exact before/after ranges and migration target given. v0.29.0

04 **MagicDNS hostname sanitisation and naming changes** BREAKING

77

Hostname sanitisation now follows Tailscale's MagicDNS rules, supporting hostnames with apostrophes, special characters, and non-ASCII characters; empty or all-non-ASCII hostnames now fall back to the literal `node` instead of `invalid-<rand>`. The `GivenName` collision suffix changed from a random 8-character hash (e.g. `laptop-abc12xyz`) to a monotonic numeric suffix (`laptop` , `laptop-1` , `laptop-2` , ...), matching Tailscale SaaS; MagicDNS names may change on upgrade for nodes previously assigned a random-suffix label.

— Explains before/after naming behavior and upgrade impact. v0.29.0

05 **nodeAttrs policy block for node capabilities** NEW

74

A new `nodeAttrs` policy block distributes Tailscale node capabilities to matched nodes, including `disable-captive-portal-detection` , `nextdns:<profile>` , `drive:share` , and `drive:access` . Taildrive file-sync is configurable through these `drive:share` / `drive:access` nodeAttrs together with `tailscale.com/cap/drive` grants.

— Names the block, capability keys, and links to grants. v0.29.0

06 **randomizeClientPort moved from server config to policy file**

72

BREAKING

The `randomize_client_port` server-config key was removed and headscale now refuses to start if the old key is present. The toggle must be moved to the policy file as the new top-level `randomizeClientPort` field.

— Names old and new keys and the exact failure behavior. v0.29.0

07 **ICMP and autogroup:self ACL rule changes** BREAKING 65

`proto:icmp` now covers only ICMPv4 (protocol 1); ICMPv6 must be specified explicitly via `proto:ipv6-icmp` or protocol number `58`. Separately, `autogroup:self` destinations now reject tags, hosts, and IPs as sources, matching Tailscale's own validation.

— Two named rule changes but no worked example. v0.29.0

THINNER COVERAGE BELOW

08 **Tailscale Grants policy block** NEW 59

A new `grants` block in policy files enables application-level controls such as Taildrive file sharing and traffic steering via tagged subnet routers or exit nodes.

— Names the block and its use cases, no config example given. v0.29.0

09 **Tailnet-wide client auto-update default** NEW 51

A new `auto_update.enabled` config option sets the tailnet-wide default for client auto-update, distributed via CapMap.

— Names the config key but gives little mechanism detail. v0.29.0

10 **Strict sequential minor-version upgrade path** BREAKING 45

Headscale now blocks skipping minor versions during upgrade (e.g. 0.27 → 0.29 is rejected) and blocks downgrading to a previous minor version; upgrades must proceed one minor version at a time.

— States the rule but no override or migration mechanism. v0.29.0

11 **Minimum supported Tailscale client raised to v1.80.0** BREAKING 35

Headscale now requires connecting Tailscale clients to be at least version v1.80.0.

— Bare version requirement with no further detail. v0.29.0

└──▷ BREAKING ON UPGRADE

! MagicDNS `Gi` `ve` collision `lap` `top`) to monotonic `l` `a` , `l` `a` , `l` `a` , ...); MagicDNS names change on upgrade for

`nName` changed from 8-char random hash (`-abc12xyz` numeric suffix (`ptop` `ptop-1` `ptop-2` any node whose previous label was a random-suffix form

! Empty or all-non-ASCII hostnames now fall back to the literal `node` instead of `invalid-<rand>`

! Wildcard (`*`) in ACL sources and destinations now resolves to the Tailscale CGNAT range (`100.64.0.0/10`) and ULA range (`fd7a:115c:a1e0::/48`) instead of all IPs (`0.0.0.0/0` and `::/0`); policies relying on `*` to match all IP addresses must switch to `autogroup:danger-all` (sou or ex CIDF (des

! Policies using tags, hosts, or IP addresses as sources for `autogroup:self` destinations now fail validation

! `proto:icmp` now only includes ICMPv4 (protocol 1); ICMPv6 must be specified explicitly as `proto:ipv6-icmp` or protocol number `58`

! Skipping minor versions (e.g. 0.27 → 0.29) is blocked; upgrades must proceed one minor version at a time, and downgrading to a previous minor version is blocked

! The `randomize_client_port` server-config key was removed; headscale refuses to start when the old key is present — the toggle must be moved to the policy file as the top-level `randomizeClientPort` field

! Minimum supported Tailscale client version is now v1.80.0

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

04 xsrfprobe

1 RELEASE · 2026-07-12

Web Application Security

The Prime Cross Site Request Forgery (CSRF) Audit and Exploitation Toolkit.

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 4/5

XSRFProbe v3 is a ground-up rebuild with 25 named CSRF checks covering token tampering, Origin/Referer bypass, SameSite gadget exploits (including client-side redirect and sibling-subdomain XSS paths), headless Firefox for browser-dependent tests, and machine-readable JSON output — a well-structured CSRF toolkit that goes well beyond what the previous version offered.

> XSRFProbe v3 rebuilds its detection engine around 25 categorized CSRF checks and adds headless-browser SameSite bypass testing with auto-validated proof-of-concepts, a bounded crawler, and machine-readable JSON reporting.

└──▷ WHAT SHIPPED · 6 FEATURES most completely described first ? what's the number?

01 25-check CSRF detection engine NEW

83

XSRFProbe v3 ships a full battery of 25 named checks (`D1` – `A1`) covering token tampering, method/Content-Type overrides, Origin/Referer bypasses, SameSite cookie analysis, and token strength, each with a unique ID, severity rating, and exploitability notes. Token-tampering checks (`T2` – `T8`) actively probe GET↔POST method switching, token omission, empty values, cross-session replay, double-submit cookies, and custom-header token forgery. Origin/Referer bypass checks (`R0` – `R2c` , `O1` – `O3`) test header removal, regex tricks, subdomain tricks, and `Origin: null` against real-world validation implementations. SameSite bypass checks (`S2` – `S4`) probe `SameSite=Strict` via client-side redirect gadgets and sibling-subdomain XSS, and `SameSite=Lax` via cookie-refresh/OAuth flows.

— Names every check ID and technique; core scan behaviour, not a distinct flag v3.0.0

02 Headless-browser SameSite bypass and PoC auto-validation NEW

83

Adds optional headless Firefox (Selenium) integration via `--browser` for browser-dependent SameSite bypass tests and auto-validation of generated PoCs; generates tailored, auto-validated exploitable CSRF proof-of-concept pages per finding. `--auto-validate-poc` runs that validation in the browser, and `--geckodriver-path` points to a geckodriver binary that isn't on PATH.

Run the full CSRF audit including browser-dependent SameSite bypass tests and get a machine-readable JSON report for pipeline consumption.

```
$ xsrfprobe --url https://target.example.com --browser --json
```

Auto-validate generated PoCs in headless Firefox and point to a non-PATH geckodriver binary for a custom test environment.

```
$ xsrfprobe --url https://target.example.com --browser --auto-validate-poc --geckodriver-path /opt/geckodriver/geckodriver
```

— Concrete flags plus a runnable command in the examples v3.0.0

03 **Bounded crawler with scope-limiting flags** NEW 80

New bounded crawler with configurable `--max-urls`, `--max-depth`, and `--crawl-timeout` flags to control scan scope, e.g. capping URL count, crawl depth, and wall-clock time for faster CI runs.

Limit crawler scope to a shallow pass on a large site so CI finishes quickly — cap URLs, depth, and wall-clock time.

```
$ xsrfprobe --url https://target.example.com --max-urls 50 --max-depth 2 --crawl-timeout 120 --json
```

— Names all three flags with a runnable example command v3.0.0

04 **JSON report output** NEW 75

Adds a `--json` flag for machine-readable JSON reports; each finding includes `test_id`, `severity`, and an `exploitability` note, suitable for pipeline consumption.

Run the full CSRF audit including browser-dependent SameSite bypass tests and get a machine-readable JSON report for pipeline consumption.

```
$ xsrfprobe --url https://target.example.com --browser --json
```

— Names flag and exact output fields with a runnable example v3.0.0

THINNER COVERAGE BELOW

05 **crt.sh subdomain enumeration for SameSite testing** NEW 40

Supports optional subdomain enumeration via crt.sh to discover sibling domains used by the sibling-subdomain XSS SameSite bypass check.

— Names the data source but no flag or config given v3.0.0

06 **Response diffing to cut false positives** IMPROVED 30

Adds a response diffing/benchmark engine that compares responses to reduce false positives before declaring an endpoint vulnerable.

— Mechanism named only briefly, no flag or metric given v3.0.0

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

05 ssh-audit

1 RELEASE · 2026-07-04

Vulnerability Management

SSH server & client security auditing (banner, key exchange, encryption, mac, compression, compatibility, security, etc)

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 4/5

ssh-audit v3.9.0 adds post-quantum warnings on all non-PQC key exchanges, built-in hardening guides accessible offline via --get-hardening-guide, UNIX socket scanning for containerized environments, and SOCKS5 proxy support — a practical set of additions for practitioners auditing SSH posture across modern and air-gapped infrastructure.

> ssh-audit v3.9.0 adds post-quantum key-exchange warnings, offline hardening guides, UNIX socket and SOCKS5 scanning, and expands its algorithm and policy databases, while dropping SSHv1 and older Python support.

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

01 Built-in offline hardening guides NEW

90

New `--list-hardening-guides` and `--get-hardening-guide` flags surface built-in hardening guides for common platforms (Debian, Ubuntu, Rocky Linux, OpenSSH) without requiring internet access; new guides were added for Debian 13, Rocky Linux 10, and Ubuntu 26.04.

Retrieve the built-in hardening guide for a specific platform to apply recommended SSH configuration without visiting an external site.

```
$ ssh-audit --list-hardening-guides
ssh-audit --get-hardening-guide "Debian 13"
```

— Named flags plus a runnable example command v3.9.0

02 UNIX socket scanning NEW

80

Supports scanning SSH servers over UNIX sockets (e.g. `unix:///path/to/socket`), enabling auditing in containerized and CI/CD environments.

Audit an SSH daemon exposed only via a UNIX socket inside a container or CI/CD pipeline — no TCP port required.

```
$ ssh-audit unix:///run/ssh-unix-local/socket
```

— Named socket syntax with a runnable command example v3.9.0

03 **SOCKS5 proxy scanning** NEW 80

New `--socks5 host:port` flag enables scanning through a SOCKS5 proxy, and implies `--skip-rate-test`.

Scan an internal host through a SOCKS5 jump proxy — useful when direct connectivity to the target is unavailable.

```
$ ssh-audit --socks5 proxy.example.com:1080 target-host.internal
```

— Named flag and behavior with a runnable example v3.9.0

04 **Policy engine additions for OpenSSH 10.x** NEW 65

Adds built-in policies for OpenSSH 10.0 through 10.4, and a new `allow_hostkey_subset_and_reordering` policy option letting policy definitions accept targets with a stricter or differently-ordered host key list.

— Names the policy option and versions but no example usage v3.9.0

05 **Expanded algorithm identification** NEW 60

Identifies 2 new post-quantum key exchanges (`mlkem768nistp256-sha256`, `mlkem1024nistp384-sha384`), 11 new host key types including ML-DSA variants (`mldsa-44`, `mldsa-65`, `mldsa-87`, and OpenSSH-namespaced equivalents) and `webauthn-sk-ecdsa-sha2-nistp256-cert-v01@openssh.com`, and 2 new ciphers (`AEAD_CAMELLIA_128_GCM`, `AEAD_CAMELLIA_256_GCM`).

— Names every new algorithm but no user-facing action v3.9.0

06 **Batch mode no longer implies verbose** BREAKING 60

Batch mode (`-b`) no longer implicitly enables verbose mode; scripts or workflows that depended on that behavior must now explicitly add `-v`.

— Names the flags and the required workaround v3.9.0

THINNER COVERAGE BELOW

07 **Post-quantum warnings on key exchanges** NEW 50

ssh-audit now warns on all key exchanges lacking post-quantum protection, flagging exposure to Harvest Now, Decrypt Later attacks.

— Explains the risk flagged but no specific command shown v3.9.0

08 **SSHv1 support removed** BREAKING 35

SSHv1 support has been removed entirely, so any scans or configurations relying on SSHv1 scanning will no longer function.

— States the removal but no migration guidance v3.9.0

09 **Python 3.8 and 3.9 support dropped** BREAKING 30

Support for Python 3.8 and Python 3.9 has been dropped, so setups running ssh-audit on those versions will no longer work.

— States the version cutoff without a migration path v3.9.0

└──▷ **BREAKING ON UPGRADE**

! Support for Python 3.8 has been dropped, so setups running ssh-audit on Python 3.8 will no longer work.

! Support for Python 3.9 has been dropped, so setups running ssh-audit on Python 3.9 will no longer work.

! SSHv1 support has been removed, so any scans or configurations relying on SSHv1 scanning will no longer function.

! Batch mode (`-b`) no longer automatically enables verbose mode, so scripts or workflows that depended on batch mode implying verbose output will see less output unless `-v` is explicitly added.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

// END OF TRANSMISSION

The Cyber Toolchain · This Week's Highlights · July 13, 2026
The full firehose — every feature release, every day — is in the newsletter.

▷ Subscribe