ROB VIPER

ILLUSTRATED USER MANUAL

Know what runs.
Decide what connects.
Understand what needs attention.

A practical guide to monitoring, execution rules, network controls, local threat scans and Mac hardening.



VERSION 0.8.1 / BUILD 18

November 2026 / Development preview

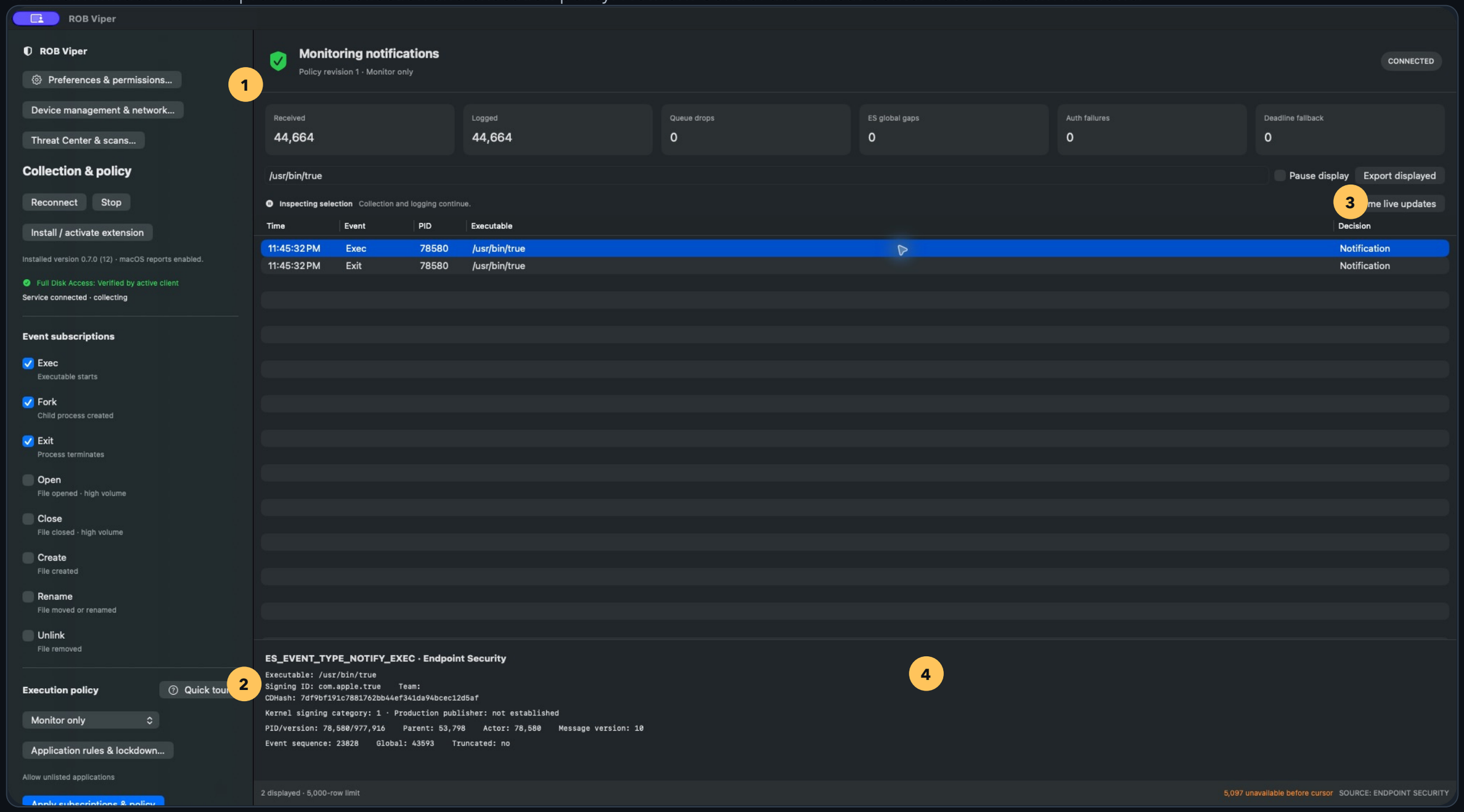
Your route through Viper

Click any entry to jump to that page. Use Preview thumbnails or PDF bookmarks to return.

The app at a glance	03
Set up permissions	04
Inspect an event	05
Monitor, Audit and Enforce	06
Rehearse an execution rule	07
Publisher, app and build rules	08
Enforce carefully; recover clearly	09
Activate Network Control	10
Read a connection	11
Build a precise network rule	12
Allow only reviewed websites	13
Answer connection approvals	14
Profiles and blocklists	15
Use the offline destination map	16
Prepare the scanning tools	17
Run a malware scan	18
Review dependency vulnerabilities	19
Use reviewed YARA-X rules	20
Read and export scan outcomes	21
Schedule a scoped check	22
Investigate behavior signals	23
Run a system hardening check	24
Understand FileVault	25
Manage a pilot fleet	26
Troubleshoot by visible state	27
Privacy and daily reference	28
Browse platform CVE reports	29
Search the application index	30
Inspect tests and patches	31
Prepare a RedAgent test	32
Run and reopen an inventory	33
Filter the full application list	34
Sources and evidence	35

The app at a glance

The main window controls process/file collection and execution policy. Network Control and Threat Center have their own windows.



1 Setup and tools

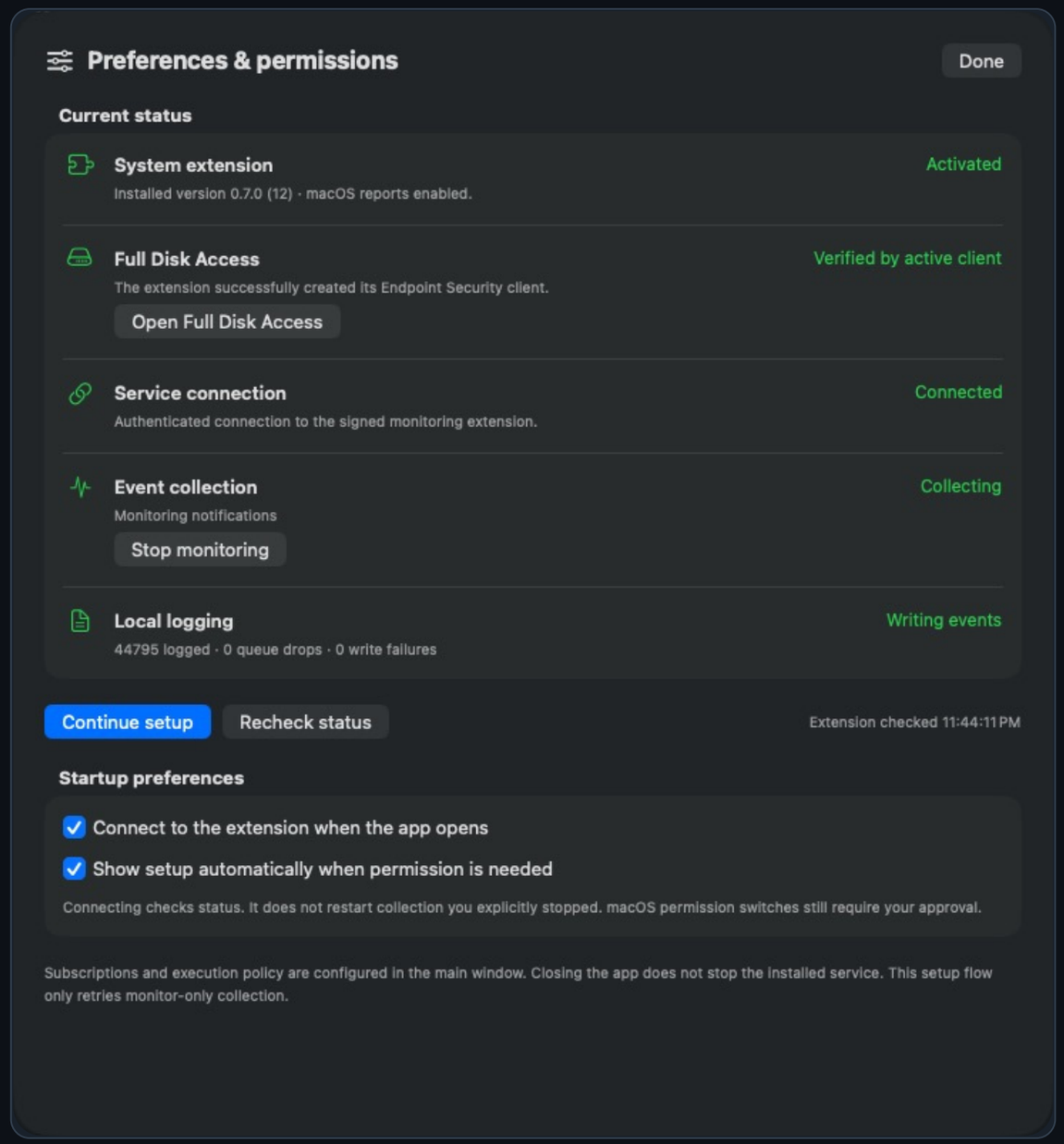
2 Execution policy

3 Hold / resume inspection

4 Selected-event identity

Set up permissions

Start with the signed development build already installed on this Mac. A public production installer is still pending.



Actual setup panel. Green states show separate successful checks; one green state does not establish all the others.

1. **Open the setup panel**

   Click **Preferences & permissions...** in the main window.

2. **Follow the visible setup state**

   Use **Continue setup**. If needed, approve the named extension in macOS, then grant Full Disk Access to the extension through the provided button.

3. **Return and verify**

   Click **Recheck status**. Look for Activated, Verified by active client, Connected, Collecting and Writing events.

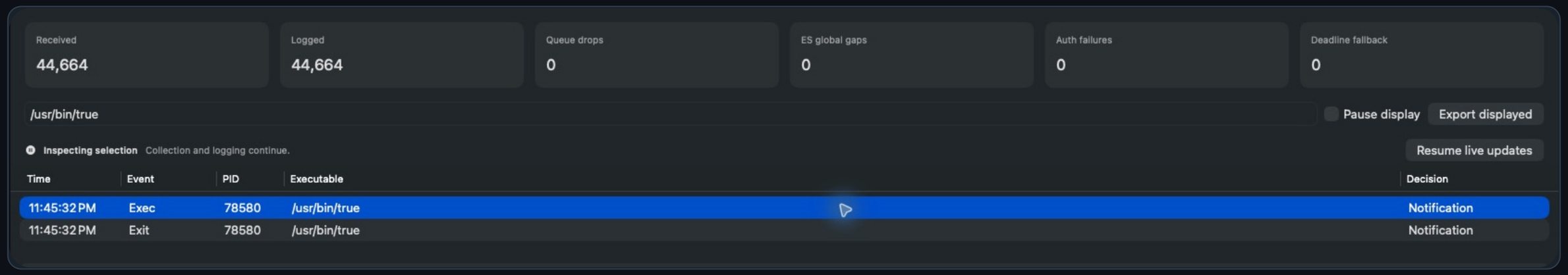
4. **Start with notifications**

   Choose **Monitor only**, select Exec, Fork and Exit, then **Apply subscriptions & policy**. Additional file events can create much more data.

> **Requirements**
>
> Host: macOS 13+. Publisher trust and Lockdown enforcement: macOS 26+. Scanner engines have separate installation and compatibility requirements.

Inspect an event without losing it

Selection holds the table and inspector while the extension continues collecting and logging.



Detail: search, status counters and the selected Exec row.



Detail: the executable identity. The pictured /usr/bin/true is protected macOS software, used here only to illustrate inspection.

A harmless first observation

Run `/usr/bin/true` in Terminal, then filter for that path. Do not use a system utility as your first blocking target.

1. **Filter, then select**

   Type a process name, path, PID or event. Click the row text once; the display should say **Inspecting selection**.

2. **Read the identity**

   Executable is the program. Signing ID and Team describe its signature. CDHash identifies that code build. An Object path on a file event is the affected file.

3. **Return to live data**

   Use **Resume live updates**. Filtering or holding the display does not stop service logging.

Three modes. Three different promises.

The active mode reported by the service is the source of truth. Editing a draft is a separate step.

OBSERVE

Monitor only

Collects your selected notifications. Viper does not request execution authorization and does not block launches.

REHEARSE

Audit execution rules

Evaluates an execution rule and records what it would do. Every launch is still allowed. A matching deny is shown as **Would deny**.

CONTROL

Enforce execution rules

Evaluates the same policy and can deny a new launch. Applying enforcement needs its own confirmation. Existing processes are not terminated.

1. Prepare a draft

   Select mode, subscriptions, rules and default action. The active service has not changed yet.

2. Apply deliberately

   Use **Apply subscriptions & policy**. Read the resulting active mode, revision and Active subscriptions.

3. Verify the outcome

   In Audit, see Would deny while the disposable program still runs. In Enforce, verify both the event and the failed launch.

Independent systems

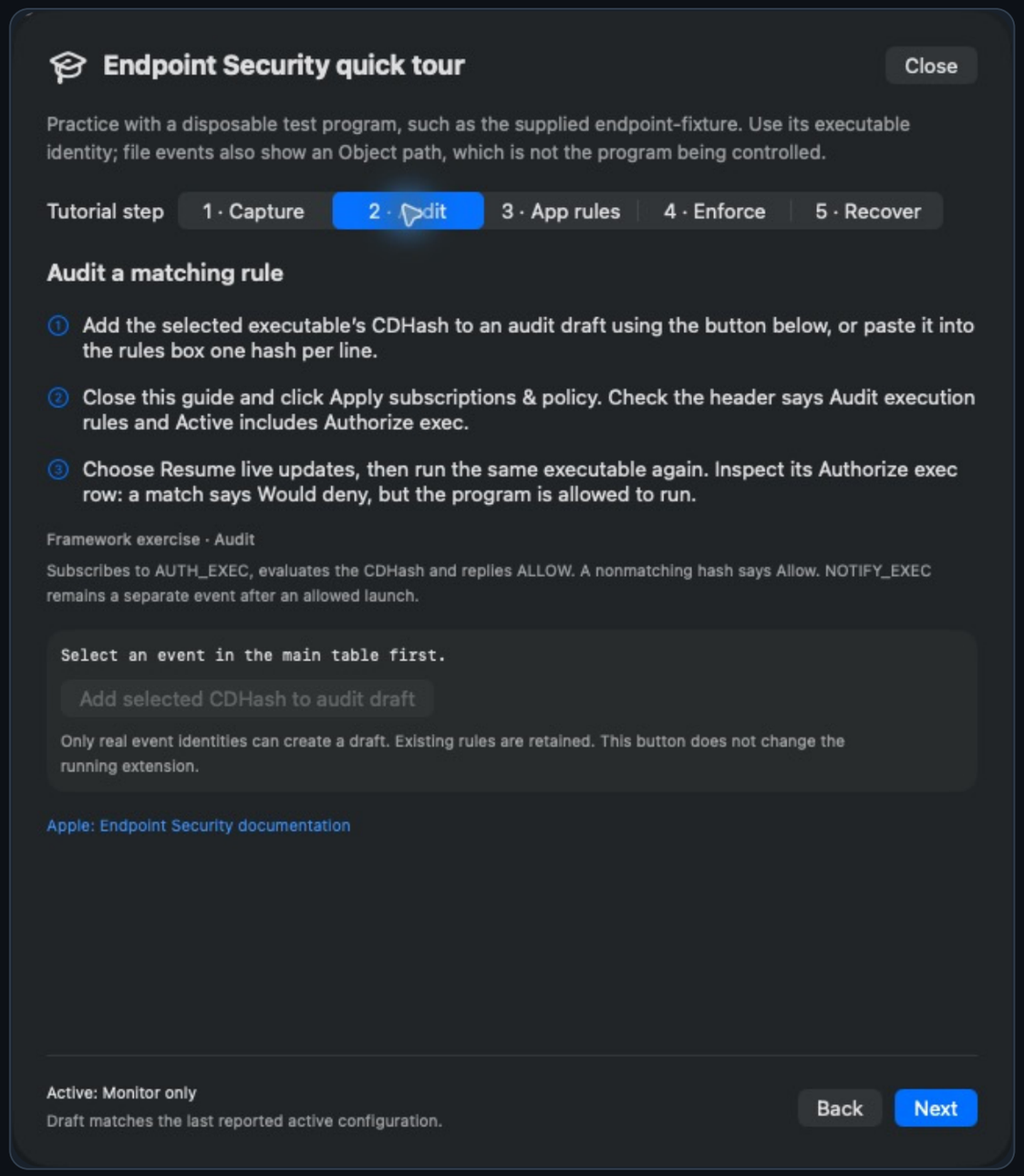
Execution mode does not enable the network filter. Network mode does not run a malware scan. Threat findings do not automatically create a blocking rule.

Platform protection

macOS platform binaries and verified Viper recovery components remain allowed. A filename, app icon or self-signed Team ID claim does not grant publisher trust.

Rehearse a rule in Audit

Use a disposable executable. The tutorial describes actions; merely reading it does not turn on enforcement.



Audit tutorial shown with no selected target. The disabled draft button becomes available only for a suitable real event.

1 Capture the exact executable

Keep Monitor only and Exec active. Run the disposable test program. Filter for its path and select the real Exec row.

2 Prepare the Audit draft

Open **Quick tour** and choose **2 - Audit**. Use **Add selected CDHash to audit draft** for the disposable program.

3 Apply and repeat

Close the tour, Apply, and check Audit plus Authorize exec in Active. Resume live updates and run the same build again.

4 Confirm the rehearsal

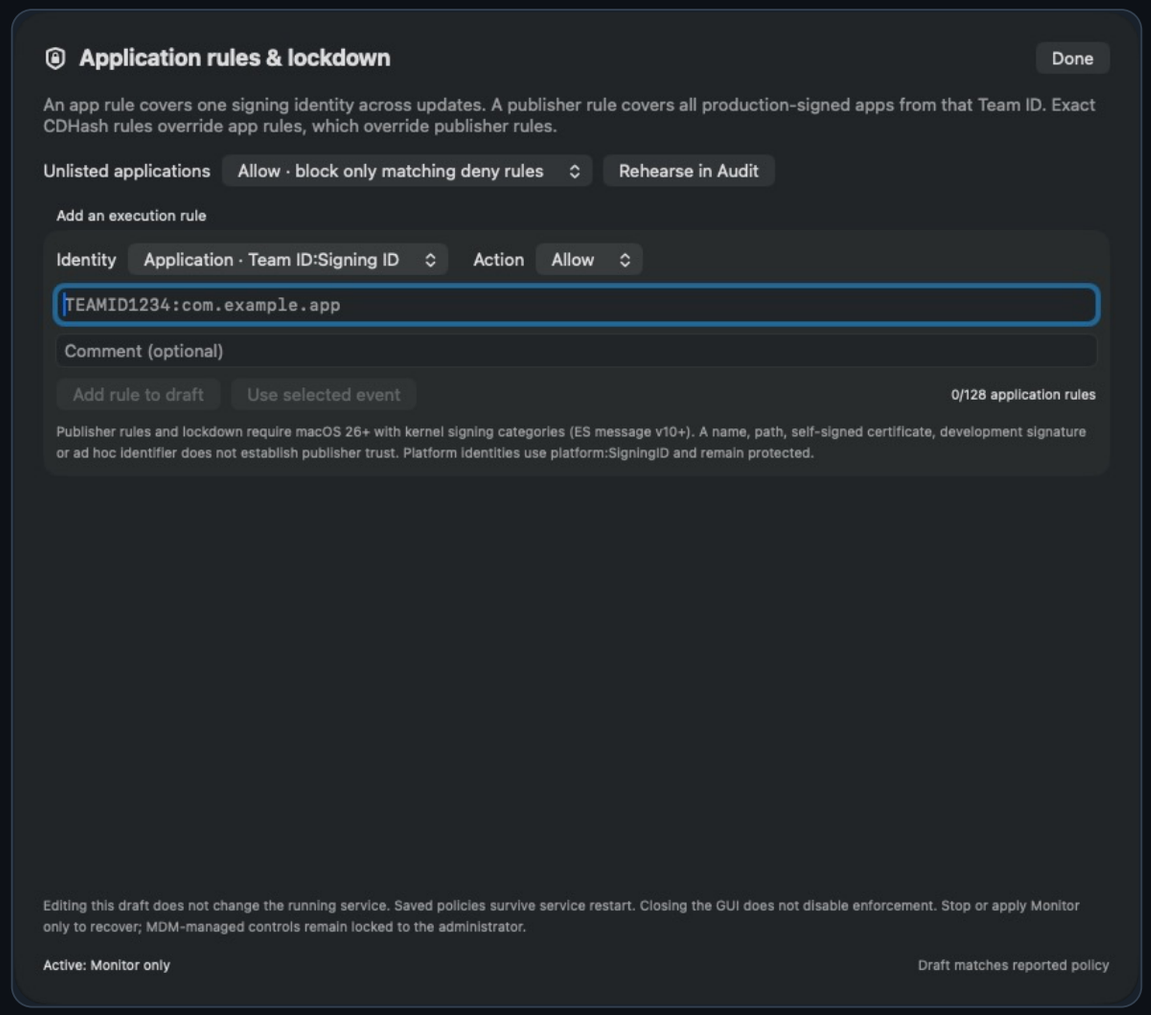
The Authorize exec row should say **Would deny**, and the program should still finish successfully. Return to Monitor only and Apply when finished.

Optional developer fixture: run these from the repository folder with Xcode tools installed. No arguments means process-only.

```
mkdir -p build/manual-lab
xcrun clang Tests/Fixtures/endpoint_fixture.c \
  -o build/manual-lab/endpoint-fixture
./build/manual-lab/endpoint-fixture
```

Choose the right identity scope

Open Application rules & lockdown beside the execution-mode picker.



An empty rule editor. Adding a rule or choosing Rehearse in Audit prepares a draft; Apply is still required.

1 One build: CDHash

A 40-character hexadecimal CodeDirectory hash. Rebuilding, re-signing or changing architecture can change it. It is not a whole-file SHA-256 checksum.

2 One app across updates

Use **Team ID:Signing ID** for a trusted production-signed app. This is narrower than trusting all software from the publisher.

3 One publisher

A Team ID rule covers all matching production-signed executables from that team. Development and ad hoc builds need exact-build handling.

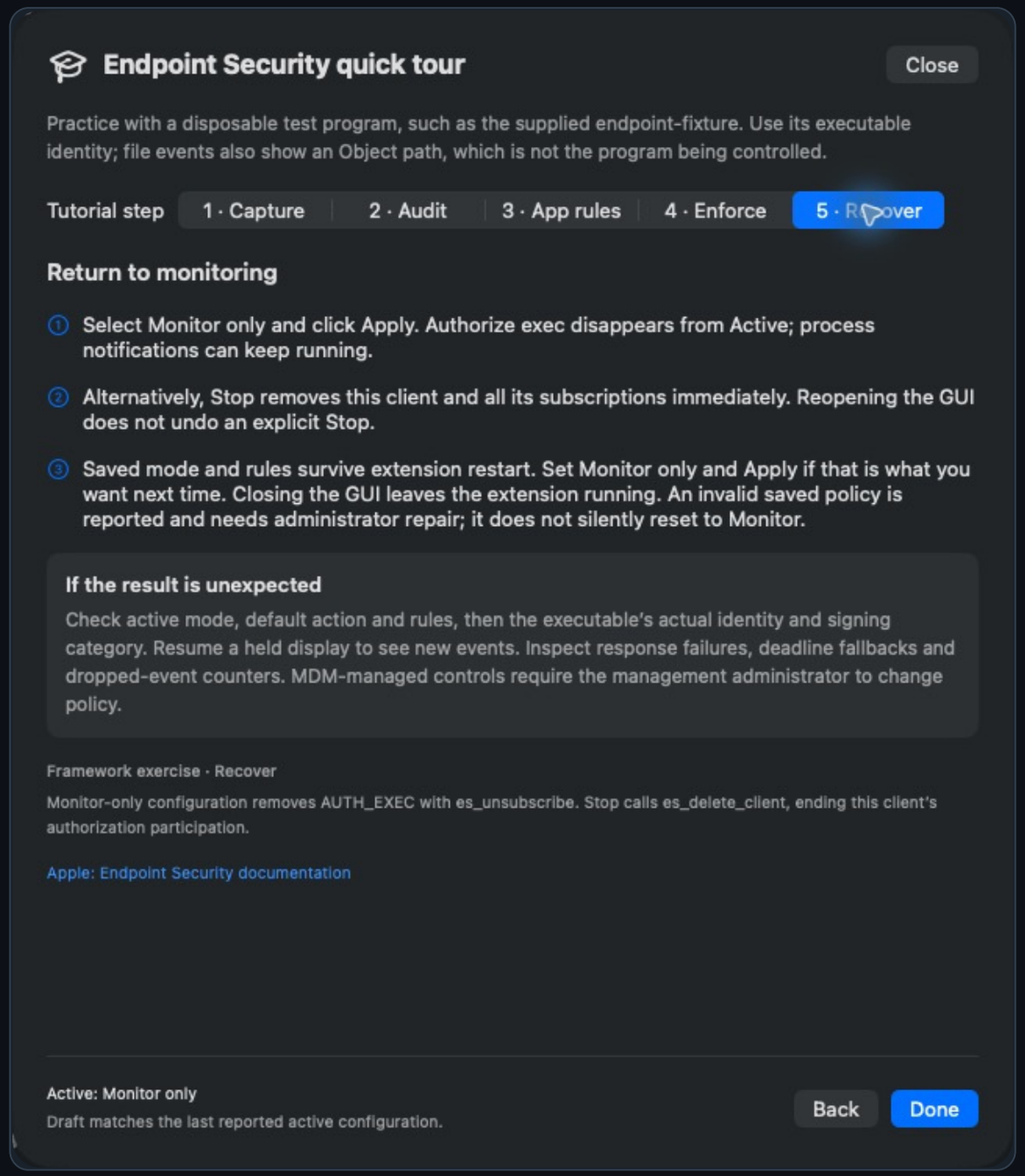
4 Set the default explicitly

Unlisted = Allow creates a deny-list policy. Unlisted = Deny creates application Lockdown. Rehearse the complete workload in Audit first.

Precedence: protected platform/recovery identities, then exact CDHash, application identity, publisher, default. Exact-build rules are most specific. The combined execution-rule budget is 128.

Know the recovery path first

Closing the host app is not a reliable way to stop an installed execution policy.



Recovery is part of the built-in tour. Forced MDM policy may lock local Apply and Stop.

1. **Make a scoped first test**

   First verify an Audit match for a disposable program. Keep Unlisted applications = Allow. Never begin with a shell, Cerebro, Viper or an everyday app.

2. **Enforce only the reviewed rule**

   Choose Enforce, Apply and confirm. Re-run that exact test build. Check Deny, response result 0, and the program failing to launch.

3. **Return to monitoring**

   Choose **Monitor only** and Apply. Authorize exec should disappear from Active. Remove test rules from the draft and apply the intended final configuration.

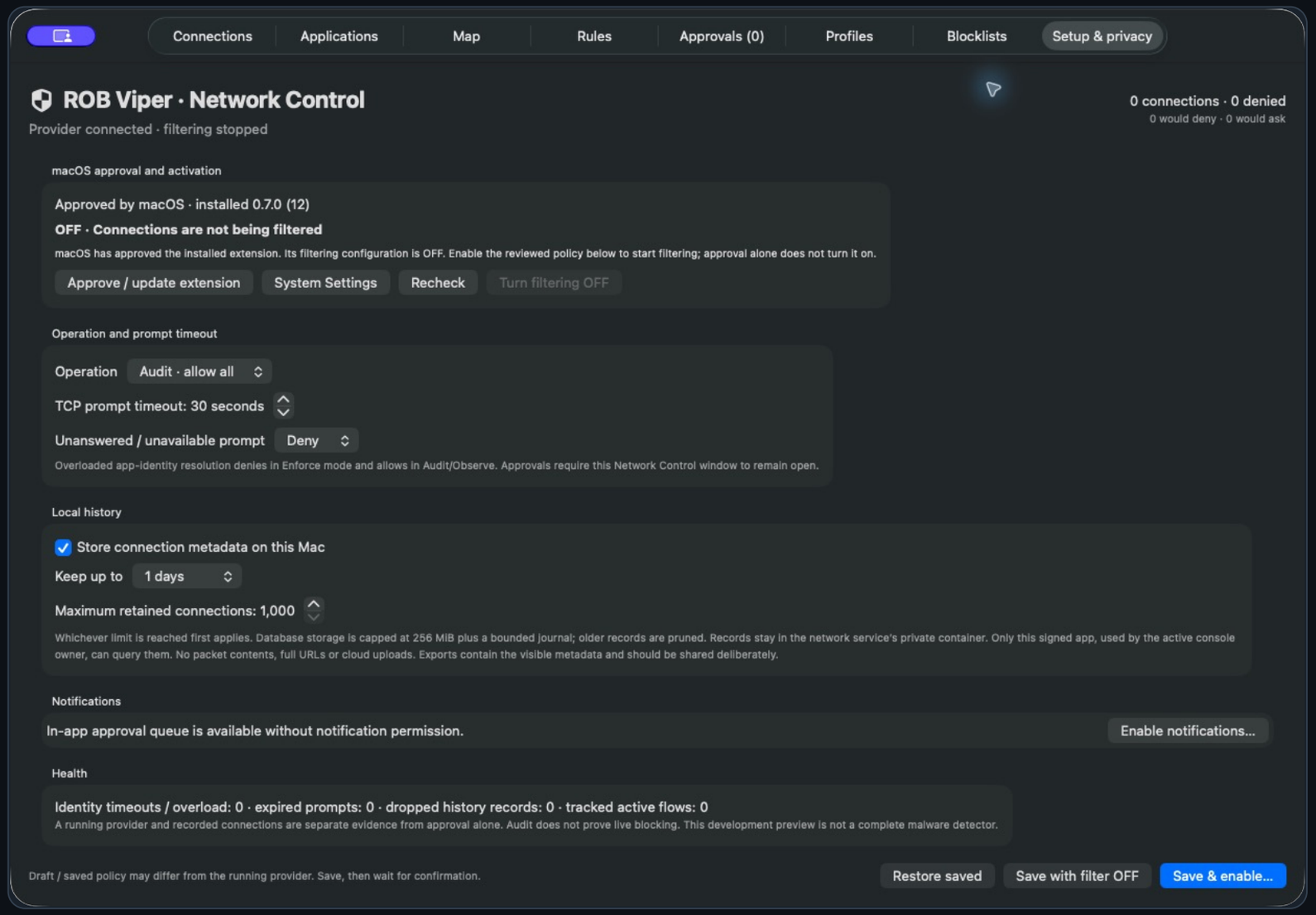
4. **Understand Stop**

   Stop removes this ES client. Saved policy can return when macOS starts a new extension process. Apply Monitor first if that is the desired next-start state.

> **Preview boundary**
>
> Live execution blocking, restart recovery and managed-fleet rollout still require production qualification. A test event by itself is not proof of a blocked launch.

Approval is not activation

Open Network Control with Command-Shift-N. Read Setup & privacy before changing a rule.



Setup & privacy. The screenshot intentionally shows filtering stopped; no network mode was enabled for this manual.

1 Check macOS approval

Approve or update the signed network extension if needed. Recheck after returning from System Settings.

2 Read the separate filter state

Approved + OFF means macOS permits the extension, but traffic filtering is stopped. This is the state used for the manual captures.

3 Begin with Audit

Choose **Audit - allow all**, review history/privacy settings, then Save & enable and confirm the intended Audit operation.

4 Confirm a running provider

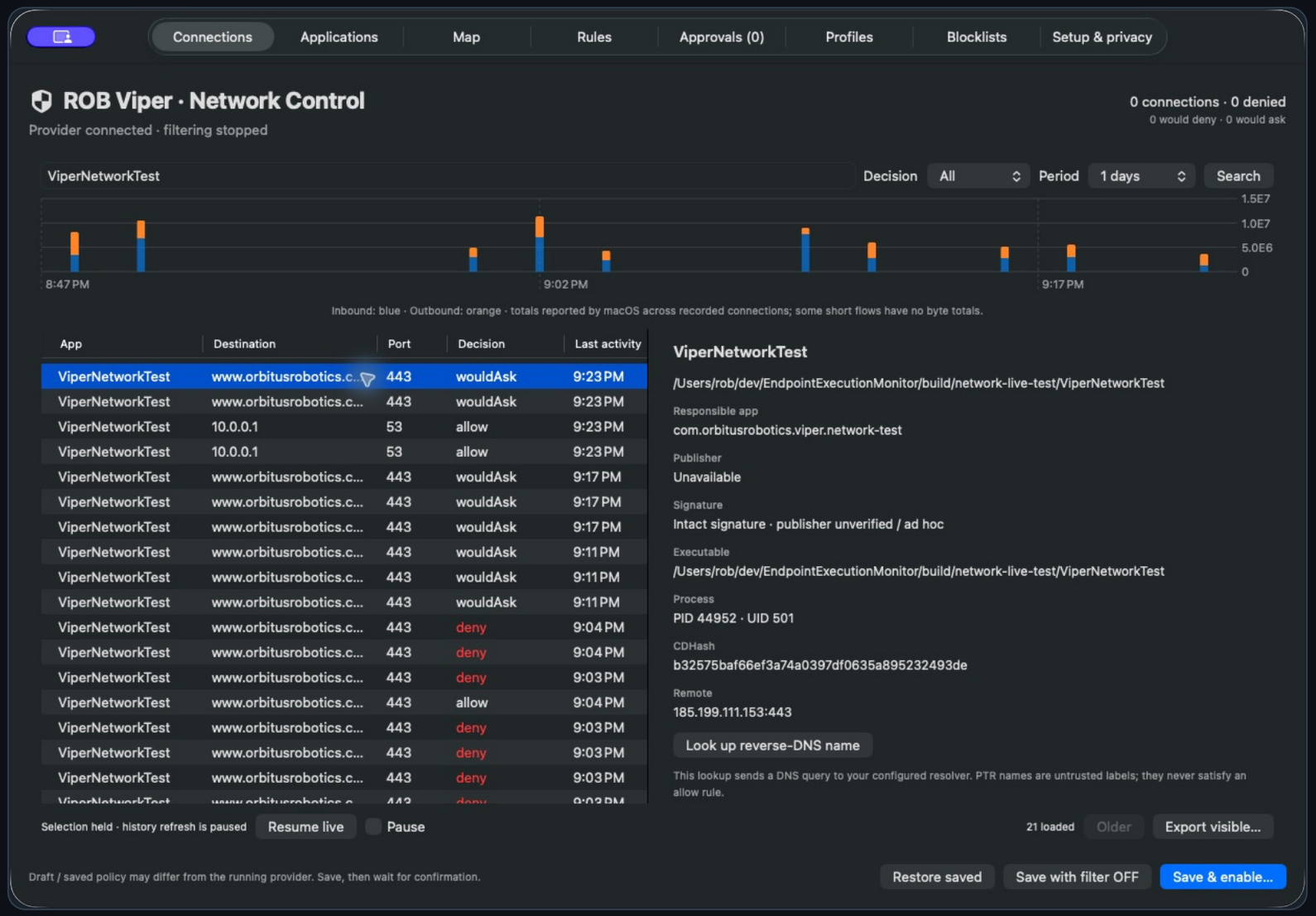
Look for a running provider and new connection evidence. An approval badge alone does not establish active inspection.

Two different save actions

Save with filter OFF stores the reviewed policy while leaving filtering off. Save & enable / Apply policy activates the selected operation after confirmation.

Read a connection, then decide

Connections and Applications summarize stored metadata. They do not display packet payloads or full browsing URLs.



Historical ViperNetworkTest records only. The filter is currently OFF. The chart aggregates the period; no byte total for a short flow means unavailable, not proven zero.

1 Search the history

Choose a period and decision filter, enter an app, signing ID, hostname or IP, then Search. Older loads more retained records.

2 Hold one connection

Select the row text. The inspector holds the responsible app, actual process, signature, destination, rule reason and state.

3 Treat DNS as a clue

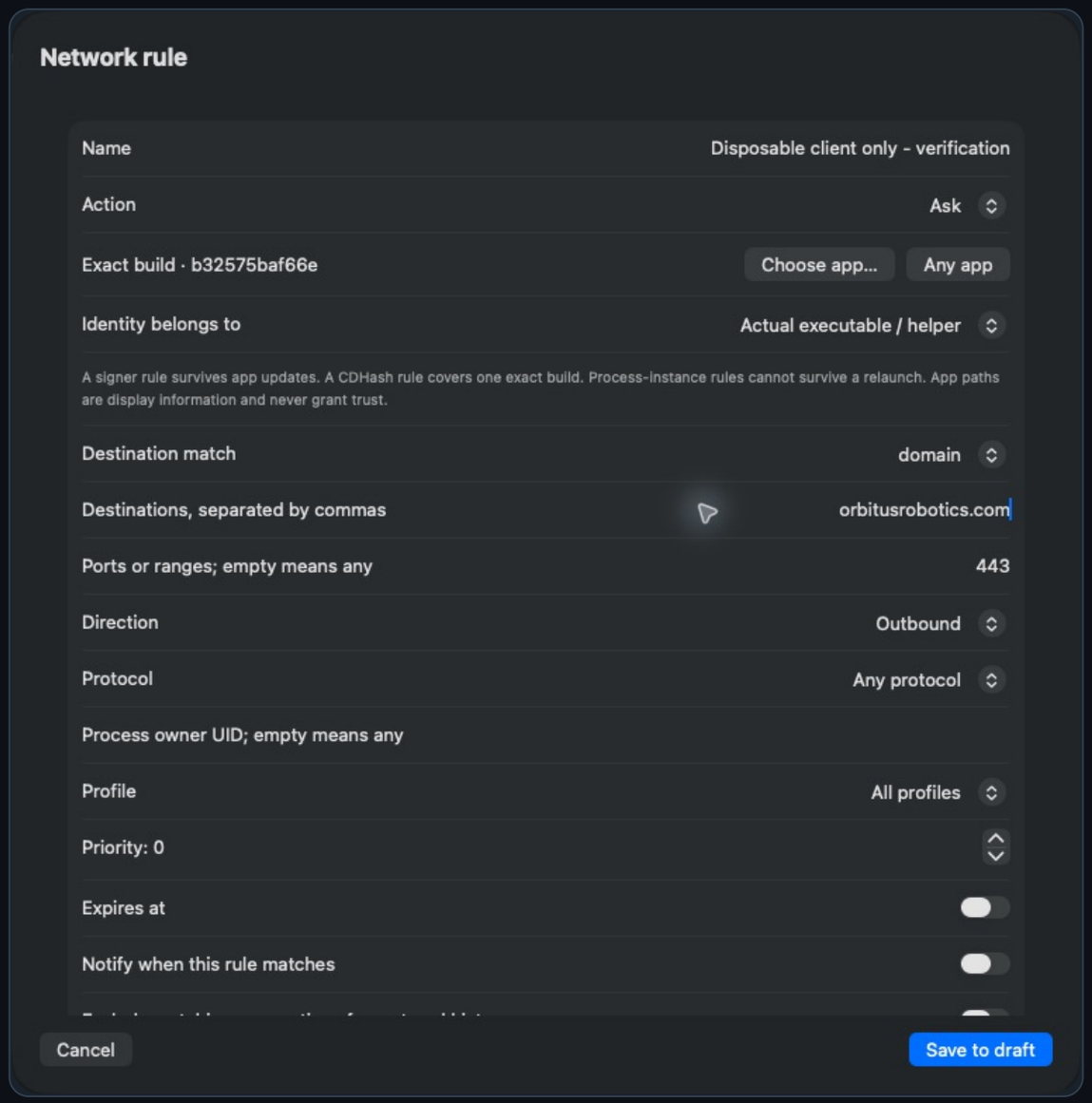
Look up reverse-DNS name asks your configured resolver for a PTR name. The answer is an untrusted label; it never authorizes traffic.

4 Resume or export

Resume live clears the hold. Export visible exports loaded rows. The graph covers recorded traffic in the selected period, not just the filtered app.

Build a precise network rule

Rules can match the responsible app or the actual executable/helper, plus destination, direction, protocol and port.



Unsaved illustration: the earlier disposable-client rule with a domain match entered. It was cancelled after capture; no active policy changed.

1 Start from observed identity

Select a connection and choose Allow rule or Deny rule, or use Rules > Add rule > Choose app. Paths are display information, not trust.

2 Choose the match carefully

Exact matches one IP or hostname. Domain matches that domain and its subdomains. Network matches a CIDR range. Ports can be individual numbers or ranges.

3 Keep the scope narrow

Set the identity source, Outbound/Inbound/Both, TCP/UDP, profile and optional expiry. An exact-build rule must be reviewed after the app changes.

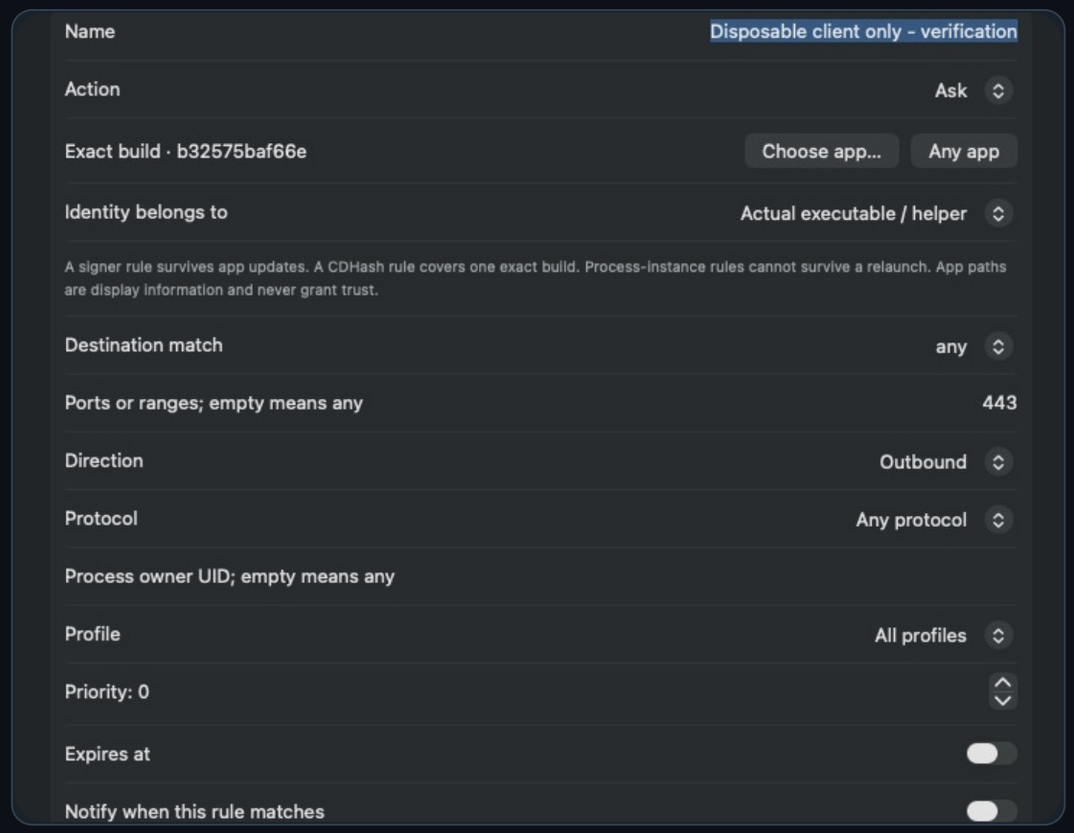
4 Save in two stages

Save to draft closes the editor. The main window Save / Apply action changes the saved or running policy. Confirm the provider revision afterward.

Rule order: higher priority first, then greater specificity; Deny wins a tie. A rule with priority above zero can override a subscribed blocklist. Review exceptions deliberately.

Allow only reviewed websites

A destination allowlist is useful, but it is not a certificate-pinning service or a complete URL-content filter.



Rule editor detail. Create your own reviewed Allow rules; do not copy this test client's Ask rule.

1 Build a pilot profile

In Profiles, create a pilot profile with Default = Deny. This affects unmatched inbound and outbound flows. Review both directions and essential services before enforcement.

2 Add explicit Allow rules

For one exact host, use Destination match = exact, for example www.orbitusrobotics.com. Use domain only when you intend to include subdomains. Add required ports and trusted app identity.

3 Include real dependencies

Review DNS, identity/login, software updates, MDM, CDN and other dependencies seen in Audit. A web page may need several destinations. No essential-service exceptions are inserted automatically.

4 Rehearse before Enforce

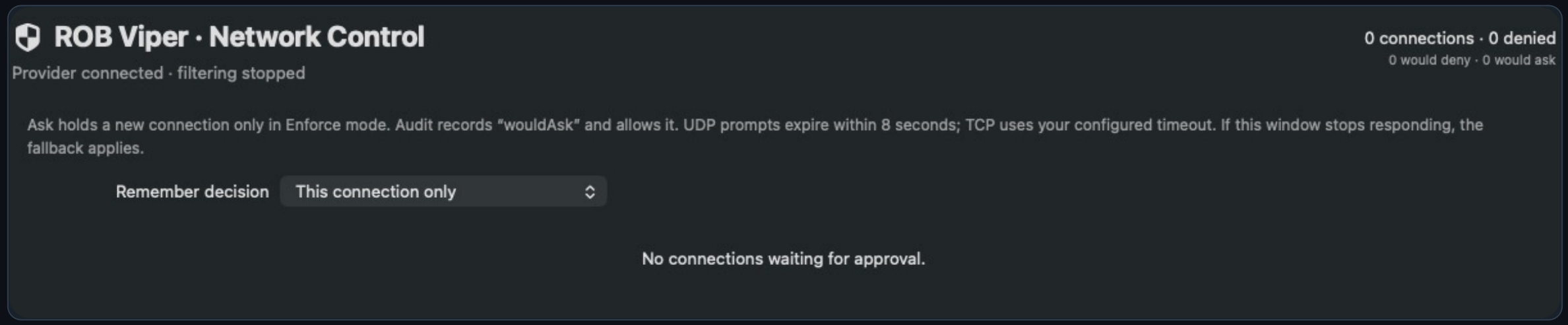
Keep Operation = Audit. Save & enable, exercise normal tasks and review wouldDeny. Only then review enforcement and a tested recovery path.

Hostname and shared-IP limits

Hostname matching depends on names supplied by macOS. Raw IP connections may have no usable hostname. Allowing a shared IP can allow other sites on that IP. A permitted domain can still host malicious content.

Answer connection approvals

Ask can pause a matching new flow in Enforce. In Audit, wouldAsk is recorded and the connection is allowed.



Approval controls with an empty queue. No pending request or Allow/Deny button has been fabricated for this guide.

1. **Keep Network Control open**

   The approval queue needs this window to remain responsive. UDP requests expire within eight seconds; TCP uses the configured timeout.

2. **Read the app and destination**

   When a request arrives, verify identity, remote destination, port and context before choosing Allow or Deny.

3. **Choose the smallest useful lifetime**

   This connection only, this process instance, until filter/policy changes, an hour, a day, or a permanent rule. Temporary decisions can end earlier when the policy or filter changes.

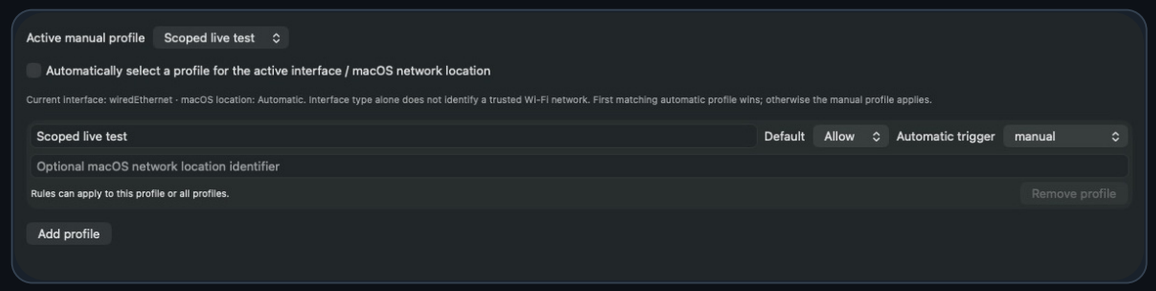
4. **Know the fallback**

   Setup & privacy defines the unanswered/unavailable prompt action. A missing response is not an implicit approval.

> **Do not mistake a rehearsal for a prompt**
>
> Seeing wouldAsk in history while Audit is active is expected. Nothing was held and there may be no approval to answer.

Profiles and blocklists

Use profiles for context, and blocklists for reviewed domain/IP feeds. Neither replaces testing the resulting policy.



Profiles detail: active profile, default action and automatic trigger.



Blocklists detail: the add-subscription entry point; no feed is configured in this capture.

1. **Define a profile**
   Choose a default action, assign rules to the profile or All profiles, and save the reviewed draft.

2. **Use automatic selection cautiously**
   Triggers can use interface type or macOS network location. Wi-Fi alone does not identify a trusted network. The first matching automatic profile wins.

3. **Review the feed source**
   Add a trusted HTTPS plain-text domain/IP or hosts list. Save the subscription before Update now or importing entries.

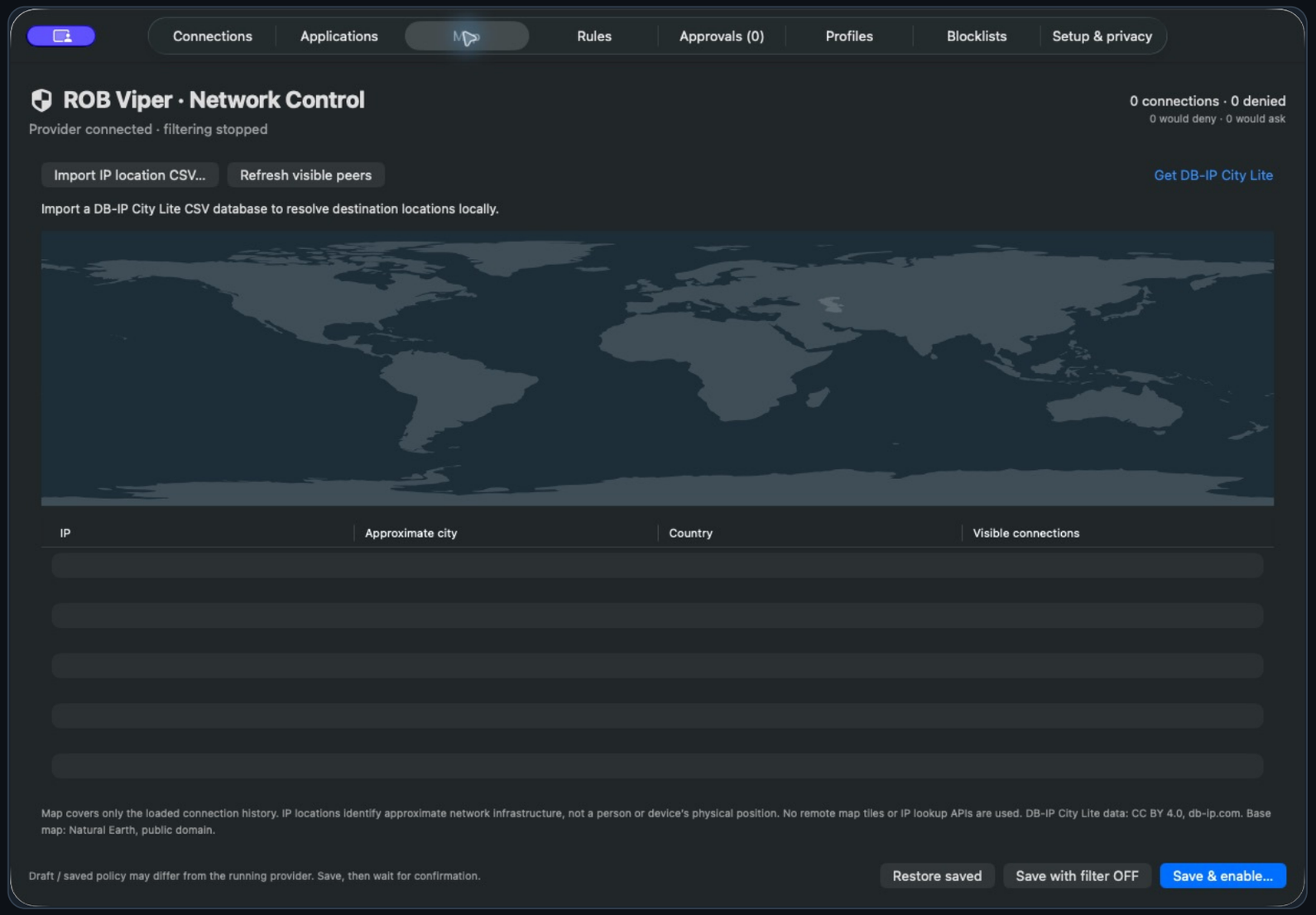
4. **Check update status**
   Invalid updates retain the last working list. Each feed is limited to 8 MiB and 100,000 entries; the preview permits up to 16 subscriptions.

Privacy and precedence

Downloading a feed reveals your public IP to its host, but does not upload browsing history. A high-priority exception can override the list. Review rules and current feed status together.

Use the offline destination map

The map is an investigation aid for network infrastructure, not a locator for people or devices.



Actual empty map before a GeoIP database is imported. Viper uses no remote map tiles or IP-lookup API for this view.

1. **Import the location data**

   Use Import IP location CSV with a compatible DB-IP City Lite CSV database. No bundled database is implied by the map outline.

2. **Load the history you need**

   The map uses currently loaded connections. Refresh visible peers after changing that set.

3. **Read approximate locations**

   Cities and countries describe IP infrastructure estimates. VPNs, CDNs and hosting can make that different from the user or company location.

4. **Expect missing points**

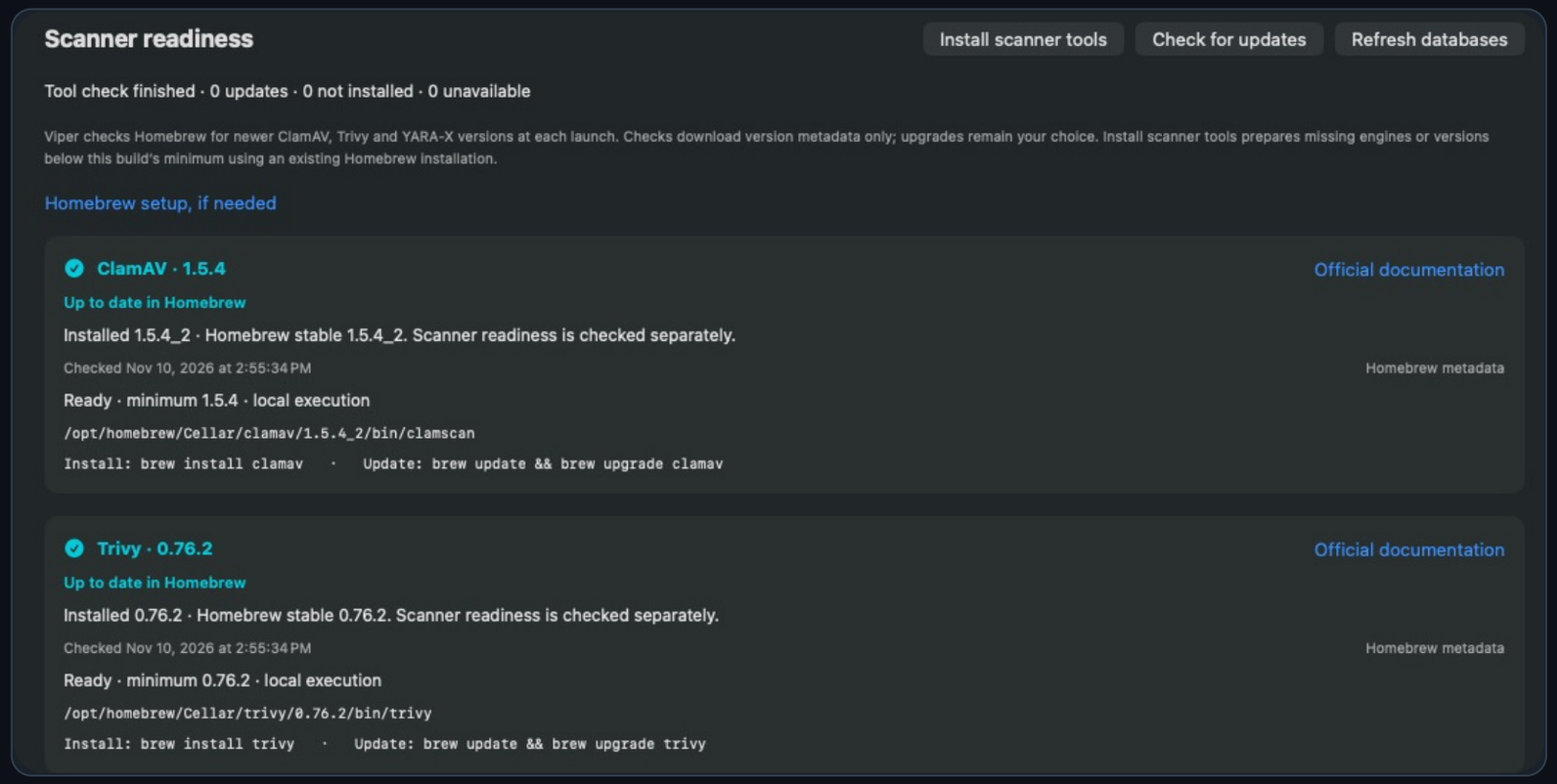
   An empty table can mean no imported database or no matching loaded peers. It does not prove that no network activity occurred.

Map sources

DB-IP City Lite data uses CC BY 4.0. Natural Earth base-map data is public domain. Keep the imported dataset current.

Prepare the scanning tools

Open Threat Center with Command-Shift-T. Scanners run outside real-time execution and network decision callbacks.



November 10 detail crop: installed and Homebrew versions, source links and timestamps. YARA-X continues below the visible area.

Choose a scope

Use a specific file or folder you intend to inspect. No whole-disk or broad home-folder scan is selected by default. Normal account file permissions still apply.

1 Check the engines

Each launch checks ClamAV, Trivy and YARA-X against Homebrew stable versions and package revisions. Use **Review tool updates** from the monitor or **Check for updates** here.

2 Install or update separately

Install scanner tools prepares missing or below-minimum engines. Use the displayed Homebrew upgrade command for a newer release. Checks never install an update.

3 Refresh advisory data

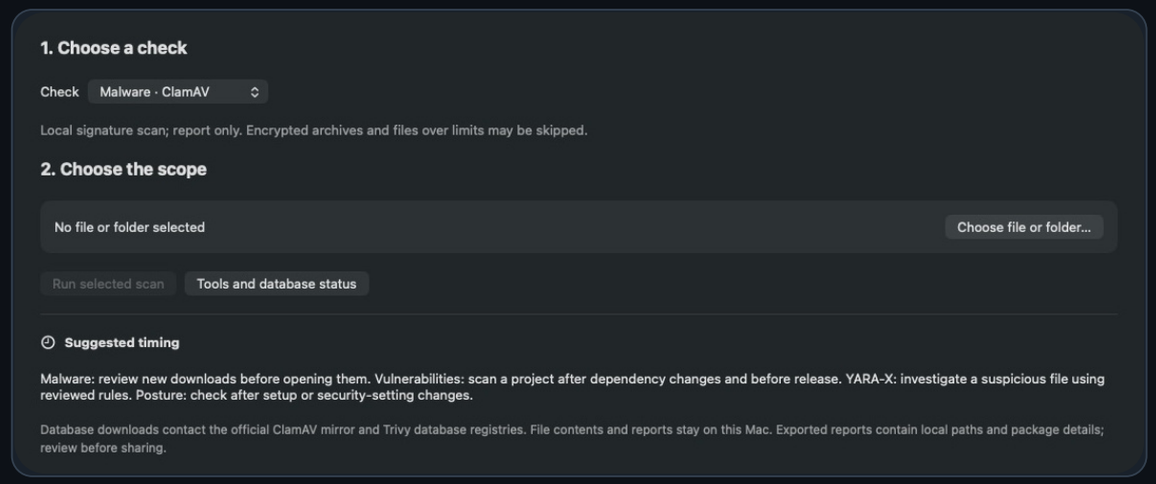
Refresh databases downloads official ClamAV signatures and the Trivy database. The local scan does not upload files or package inventories.

4 Resolve not-ready states

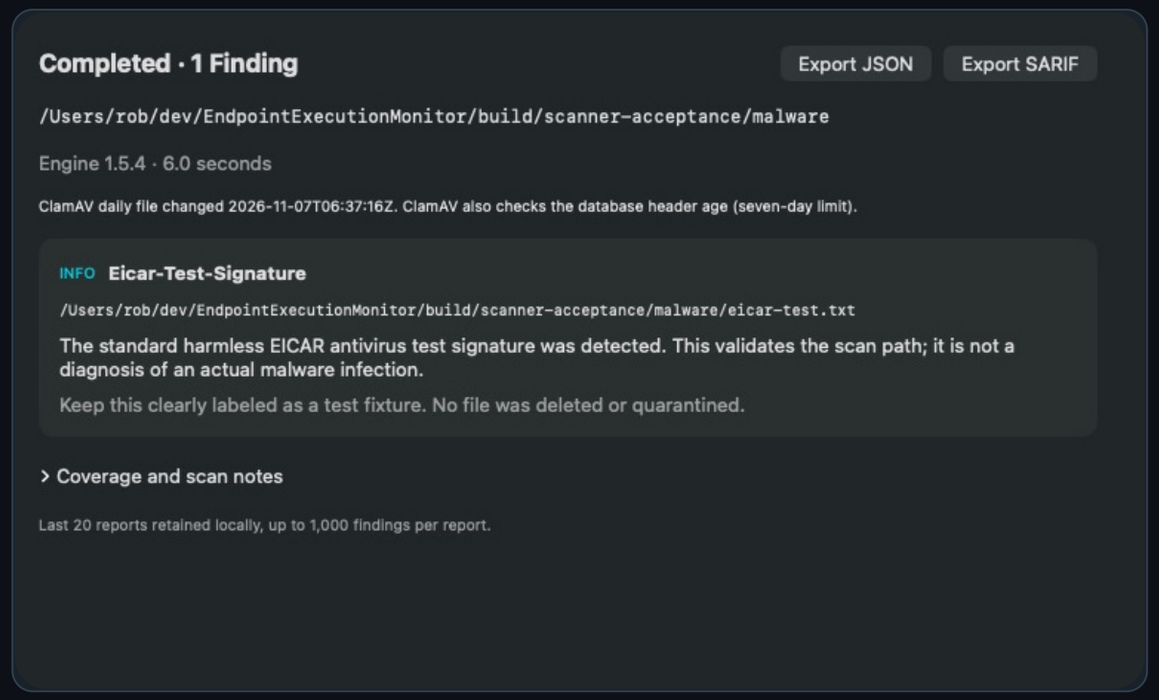
Older engines, missing tools and databases older than seven days prevent a valid scan. Read the exact status before retrying.

Run a malware scan

ClamAV reports signature matches in a selected scope. It does not delete or quarantine files.



Run scan detail: choose Malware, select a file or folder, then Run selected scan.

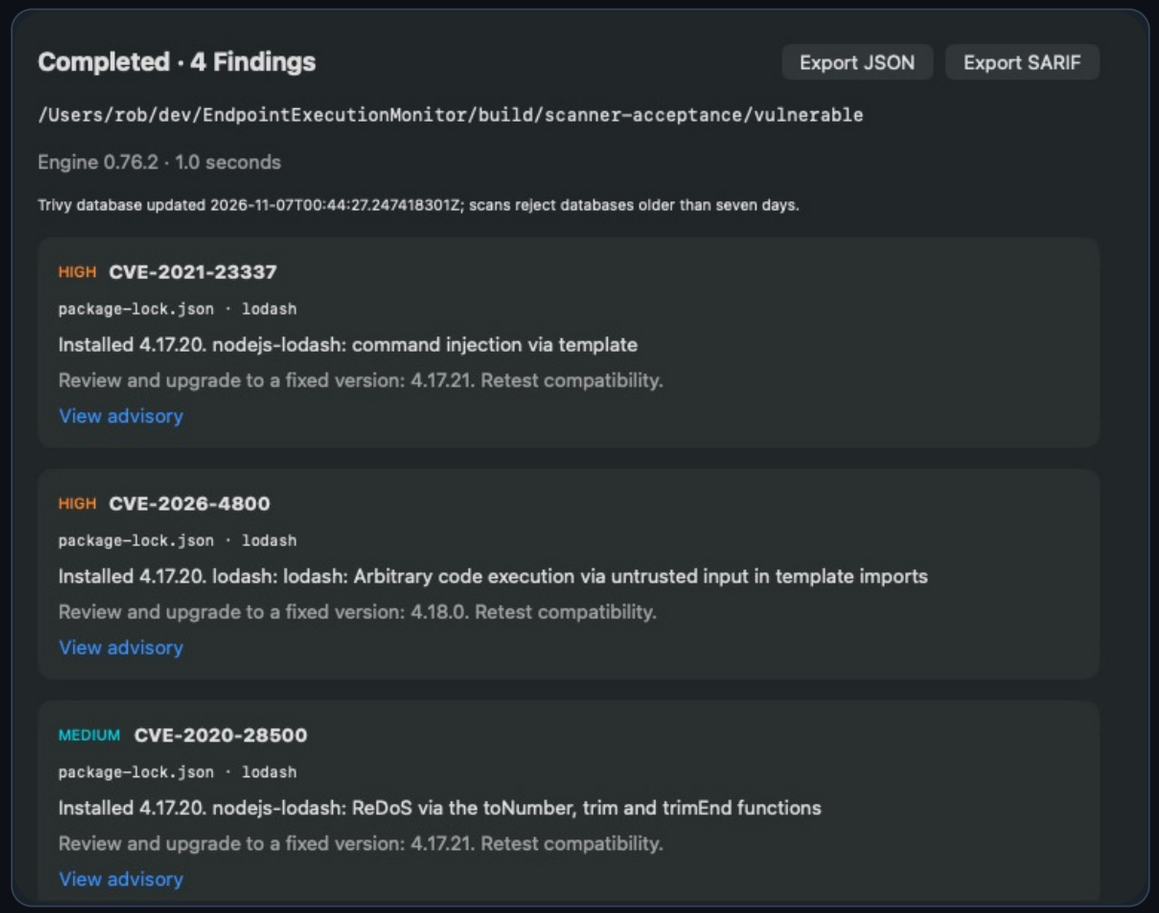


Report detail: EICAR is a harmless antivirus test signature, clearly labeled as test data.

1. **Choose Malware - ClamAV**
   Scan a new download before opening it, or choose a limited folder for a periodic review.
2. **Run, then read Reports**
   Review the scope, engine, outcome, severity, signature name and recommendation. Export if you need a record.
3. **Act on evidence**
   A real signature match deserves investigation. Verify provenance and use your organization's response process; the app performs no automatic cleanup.
4. **Respect coverage limits**
   The 64 MB file limit, archive limits, encryption, permissions or time limits can prevent inspection. Incomplete is not clean.

Review dependency vulnerabilities

Trivy analyzes supported project manifests and lockfiles using the downloaded advisory database.



Report detail from a metadata-only lodash test project. The vulnerable package was not installed or executed.

A point-in-time report

Advisory databases and fixed-version guidance change. Retain the scan date and database detail with the report. An empty result is not proof that the Mac is fully patched.

1. **Scan the project folder**

   Choose Vulnerabilities - Trivy in Run scan. Use it after dependency changes and before a software release.

2. **Read each advisory**

   Compare package name, installed version, CVE, severity and suggested fixed versions. Open View advisory for primary investigation details.

3. **Update and retest**

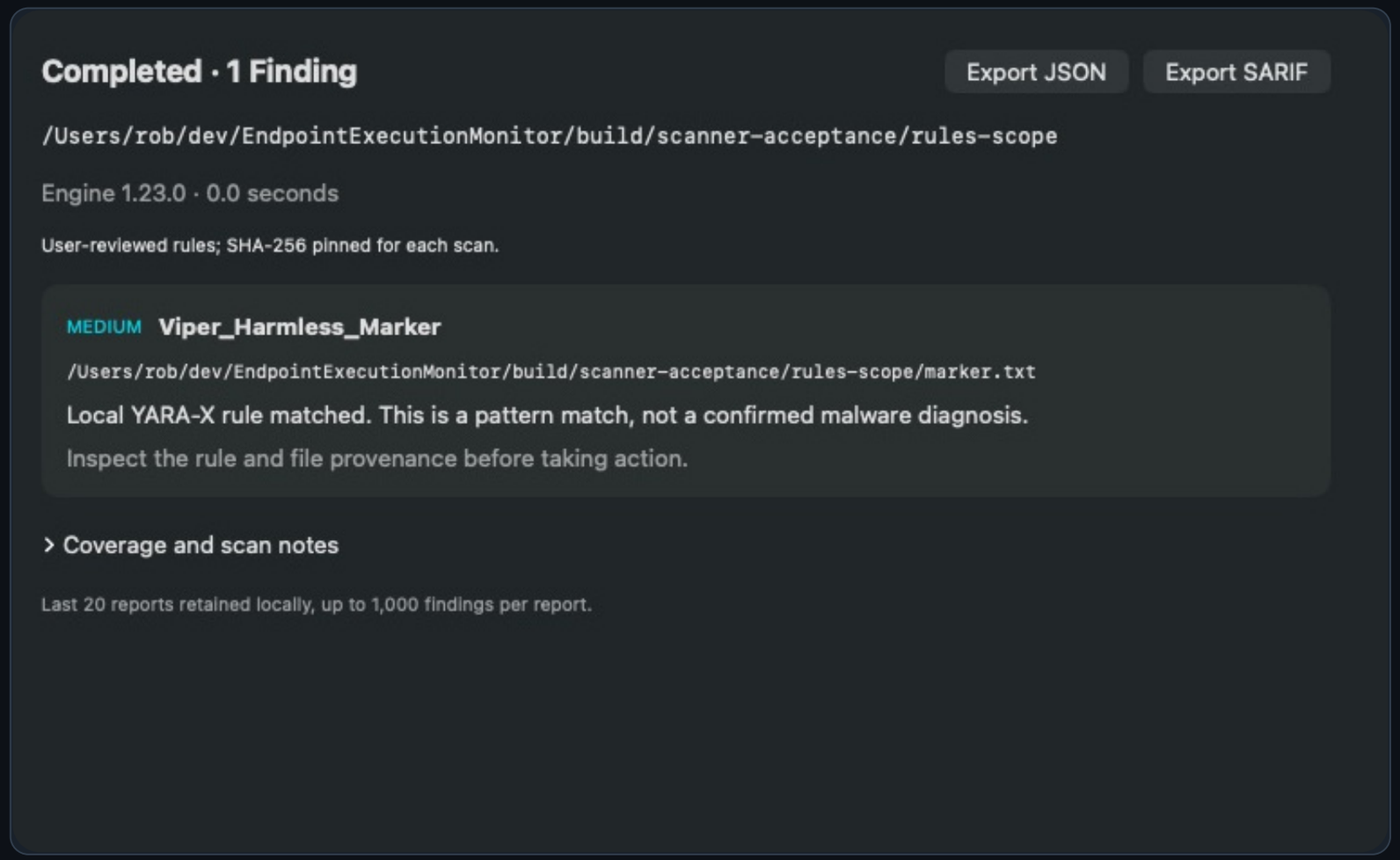
   Review the proposed fix, update the dependency intentionally, test your app and run the same scoped scan again.

4. **Keep the limits in view**

   This is not an inventory of every Mac application or every macOS CVE. No supported targets or missing indexes can limit the result.

Use reviewed YARA-X rules

YARA-X answers whether local file content matches patterns in the rule source you supply.



Report detail. The test rule Viper_Harmless_Marker matched a deliberately created text fixture.

1. **Choose Threat rules - YARA-X**
   Select the artifact or folder you intend to inspect.

2. **Select reviewed source**
   Use Choose reviewed YARA rules. Supply one trusted UTF-8 source file up to 1 MB. Includes are rejected; rules are not downloaded automatically.

3. **Keep the pinned identity**
   Viper records the rule SHA-256 and checks it again before scanning. A changed file must be selected and reviewed again.

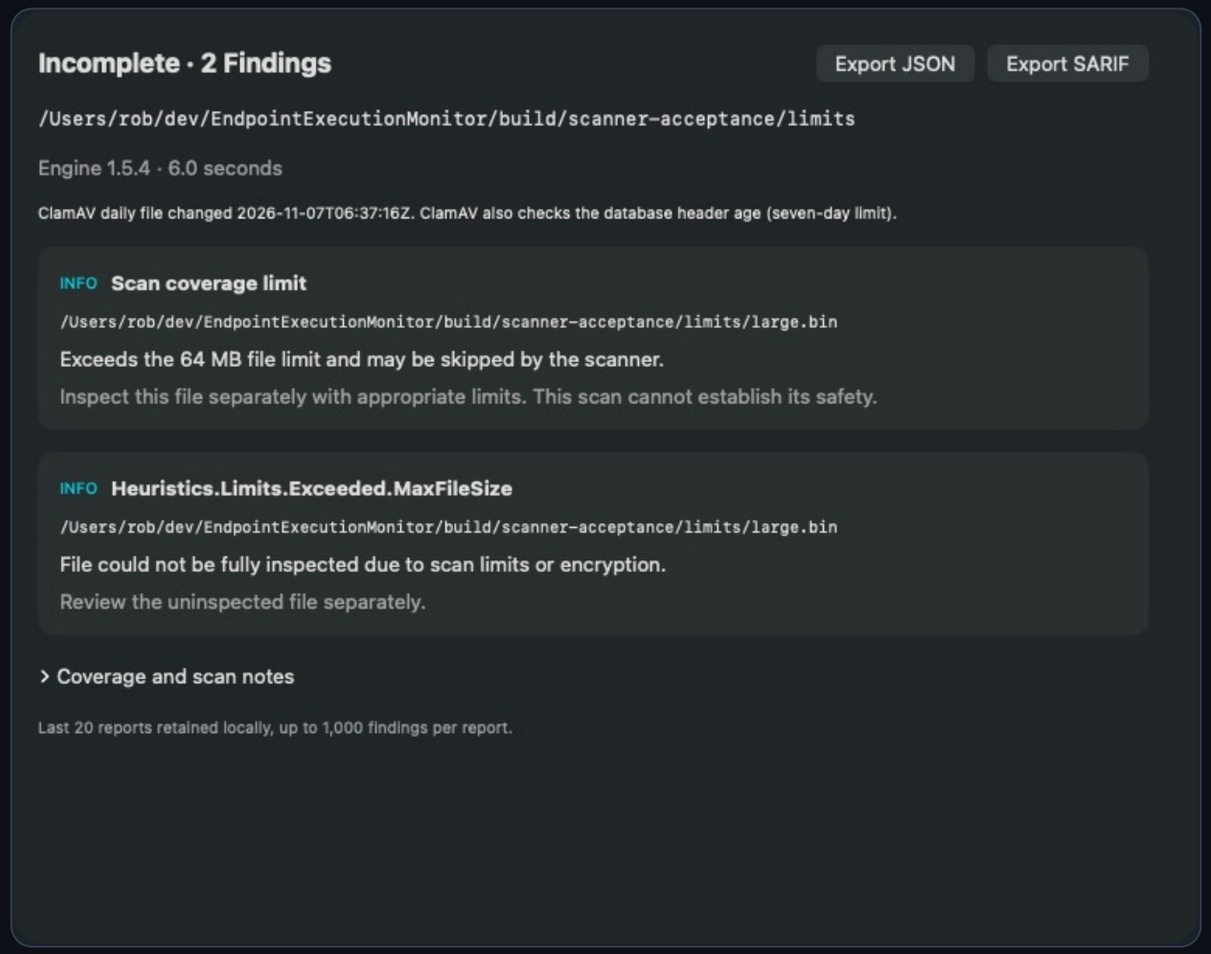
4. **Interpret a match**
   Review the rule logic and file provenance. A match is an investigation lead, not a confirmed infection. The capture uses a harmless custom marker.

Bounded work

YARA-X uses limited threads, a rule-scan timeout, depth limits and a 64 MB file limit. Skipped or inaccessible content is outside the evidence of a match-free result.

An outcome is not a verdict of safety

Read the outcome and coverage notes before interpreting the finding count.



Real limit test: a deliberately oversized file produced Incomplete, with the uninspected file named.

Completed

The tool finished the configured scan. No findings means none reported within that scope and its limits.

Incomplete

Some content could not be inspected, or the tool reported an inspection problem. Investigate the coverage gap.

Failed

The operation could not produce a usable result: for example, missing tools, malformed output or changed rules.

Cancelled / timed out

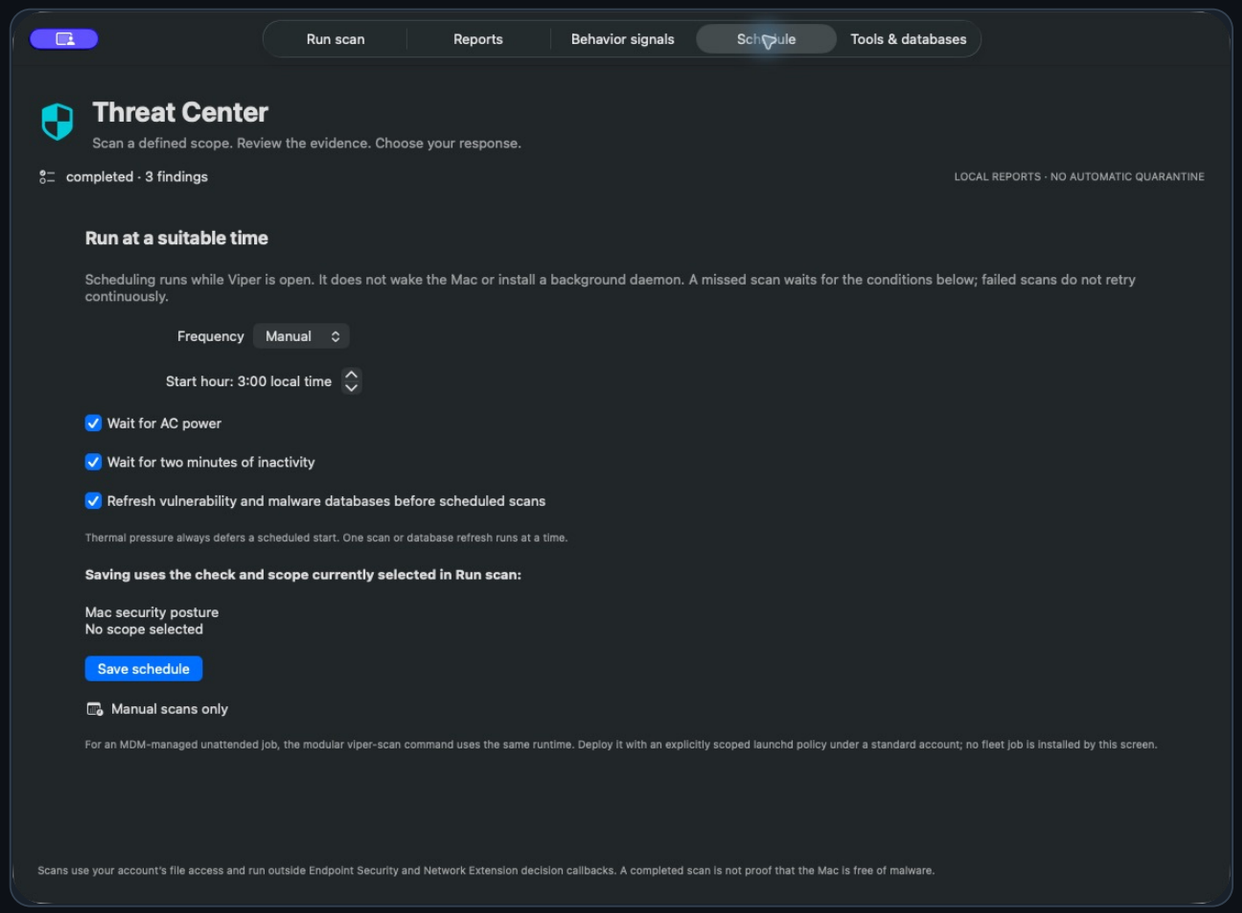
The job stopped before completion. A zero finding count does not establish a clean result.

Export a useful record

Select a report, then Export JSON or Export SARIF. JSON preserves detailed evidence; SARIF fits supported code-scanning workflows. Incomplete and unsuccessful SARIF runs are marked unsuccessful. Review local paths before sharing.

Schedule a check at a suitable time

Saved schedules use the per-user scan agent after Viper closes. The Mac must be awake and your account logged in.



Historical 0.7 timing controls. Version 0.8.1 adds agent registration, heartbeat and last-scan status; Manual remains the default.

Separate lifecycles

Saved schedules and Run Saved Scan Now use the agent. Manual scans and behavior analysis remain host-owned. Saving Manual cancels future runs, not a job already running.

1. **Choose the job first**

   In Run scan, select the scan kind and a specific scope. For YARA-X, select the reviewed rule source too.

2. **Set timing in Schedule**

   Choose Manual, Daily or Weekly and the local start hour. Review AC power, two-minute inactivity and database-refresh options.

3. **Save the actual job**

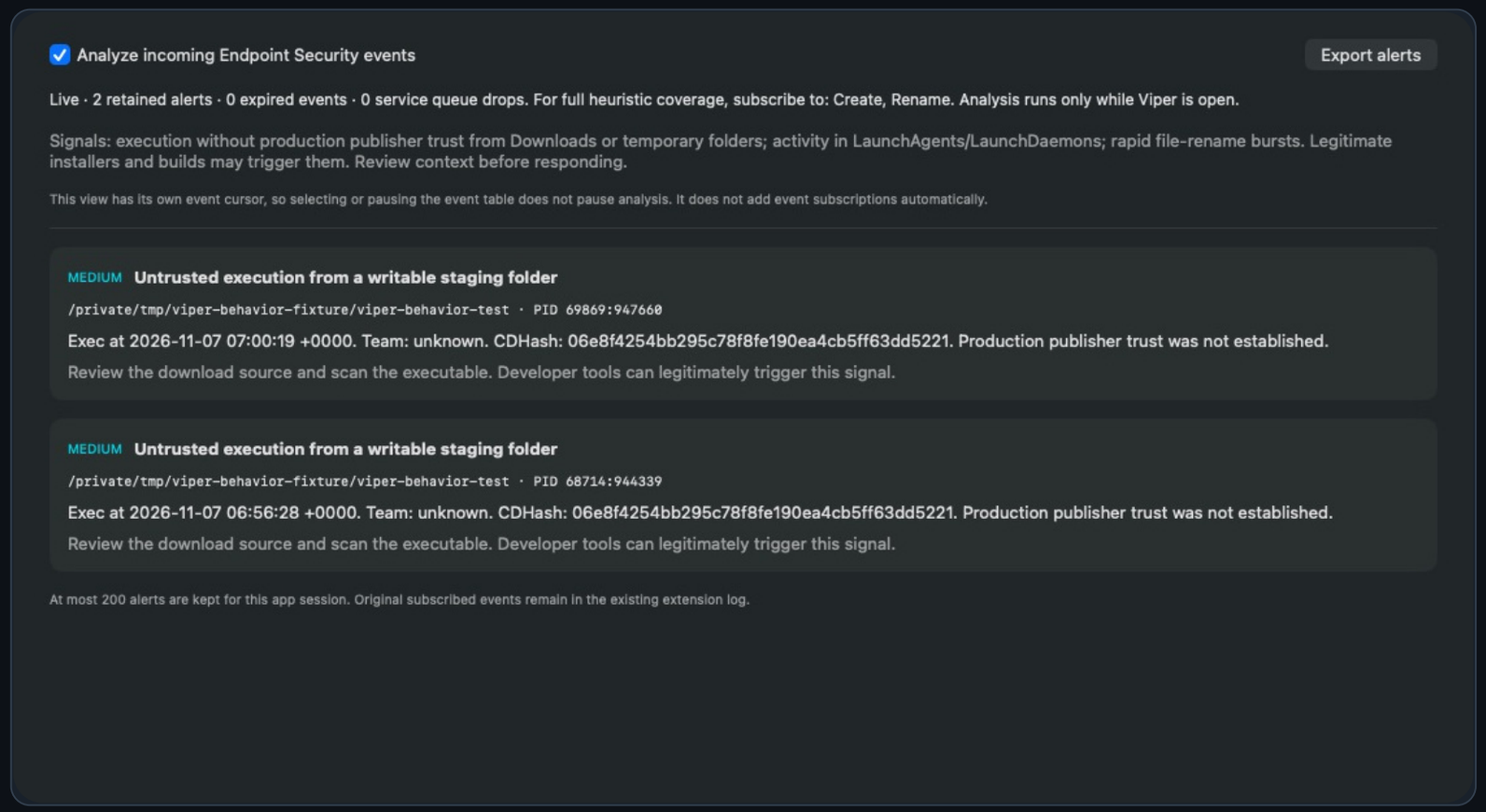
   Save captures the selected scope. Check agent registration, approval and heartbeat. **Enable background components** retries registration; macOS Login Items approval may be needed.

4. **Understand deferral**

   A missed start waits for the required conditions. Thermal pressure always defers a start, and only one scan or update runs at a time.

Investigate behavior signals

These are explainable leads. They do not automatically block a process, quarantine a file or change policy.



Two harmless test executions generated these alerts. Create and Rename were not enabled; the UI accurately reports that coverage gap.

Independent inspection

Behavior analysis uses its own event cursor. Holding the main table does not pause analysis. Closing Viper does stop host-side behavior analysis.

1. **Check the event coverage**

   The panel lists missing subscriptions and service/event losses. For full heuristic coverage, use Exec, Create and Rename in the main monitor and Apply.

2. **Read the evidence**

   Signals cover execution without production-publisher trust from staging folders, persistence-location activity, and rapid rename bursts.

3. **Consider benign causes**

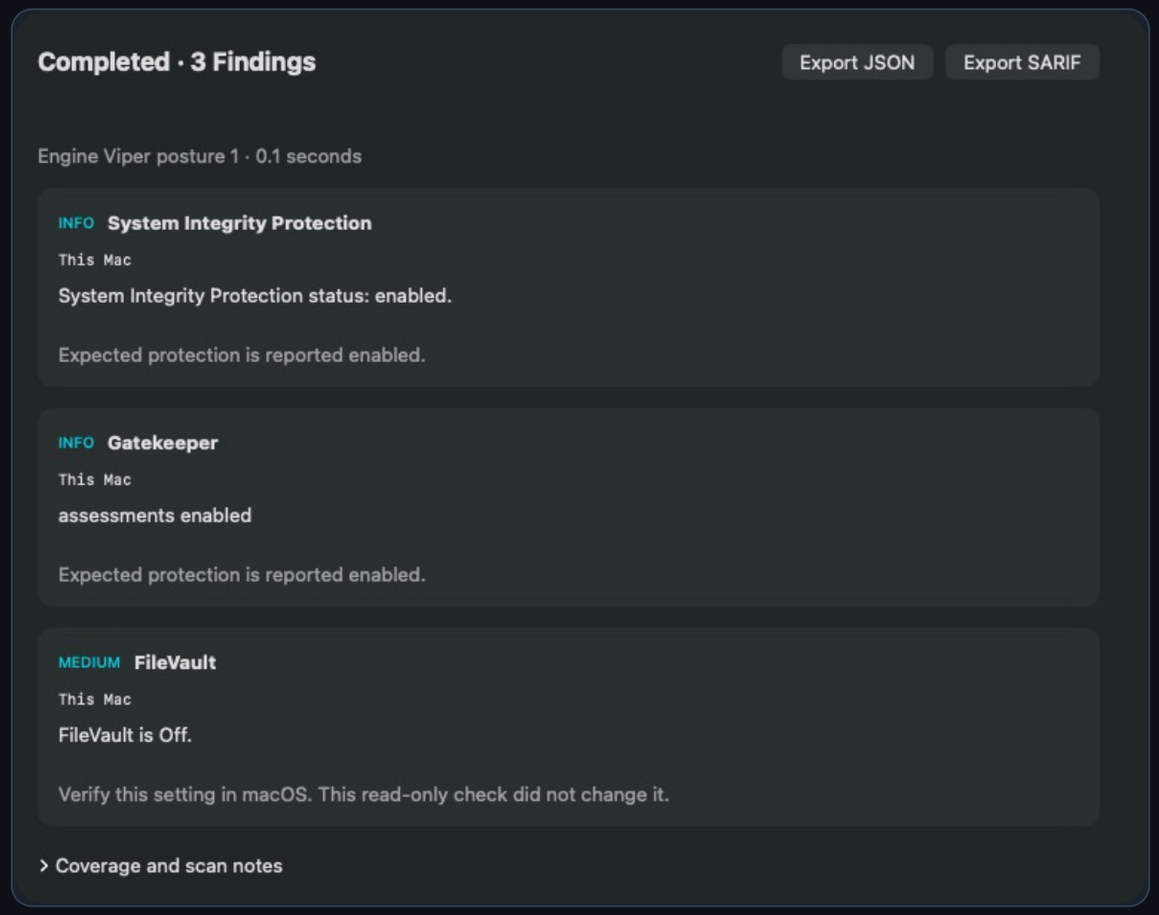
   Build tools, installers and sync software can trigger these patterns. A rename burst does not prove encryption; a close event alone does not prove a write.

4. **Preserve and investigate**

   Export alerts if needed, inspect the original event and scan the scoped artifact. Up to 200 alerts are retained for the app session.

Read the Mac security posture

Yes: this is a read-only hardening check. It inspects selected protection settings, not whether every file is safe.



Actual report on the evaluation Mac: SIP enabled, Gatekeeper assessments enabled, FileVault off. This is an example state, not a required configuration.

Three checks, not a full audit

This first posture check does not assess every update, backup, account, firewall, MDM or recovery setting. A completed report is not a security certification.

1. **Run the check**

   In Threat Center > Run scan, choose Mac security posture, then Run selected scan. No file or folder is required.

2. **Review all three controls**

   SIP protects important system locations and restricts privileged modification. Gatekeeper checks software trust at launch. FileVault protects stored data with user-bound disk unlocking.

3. **Interpret the severity**

   INFO can record an expected enabled state. MEDIUM flags a protection to review. Unknown or unavailable needs manual verification.

4. **Change settings separately**

   The check ran csrutil status, spctl --status and fdesetup status. It did not enable, disable or repair any protection.

What does "FileVault off" mean?

It is a configuration finding. It does not mean malware was found, and on Apple silicon it does not mean the internal drive has no encryption.

APPLE SILICON / T2

FileVault OFF

On Apple silicon and T2 Macs, internal storage is already hardware-encrypted. With FileVault off, the volume-key protection is based on the hardware UID alone.

ADDITIONAL CREDENTIAL PROTECTION

FileVault ON

FileVault binds unlocking to the user's credentials as well as the hardware. This strengthens protection for stored data when someone has physical access to the Mac.

1. ### Verify it in macOS

   Open **System Settings > Privacy & Security > FileVault**. This check concerns the startup volume; external drives and backups have separate encryption settings.

2. ### Plan recovery before enabling

   Keep a current backup. Choose the recovery method offered by macOS and keep the recovery information safely accessible away from the startup disk. For a managed Mac, use the organization's recovery-key process.

3. ### Understand the operating impact

   Enabling FileVault changes startup unlocking. Plan for reboot and unattended-access needs before enabling it on an always-on machine.

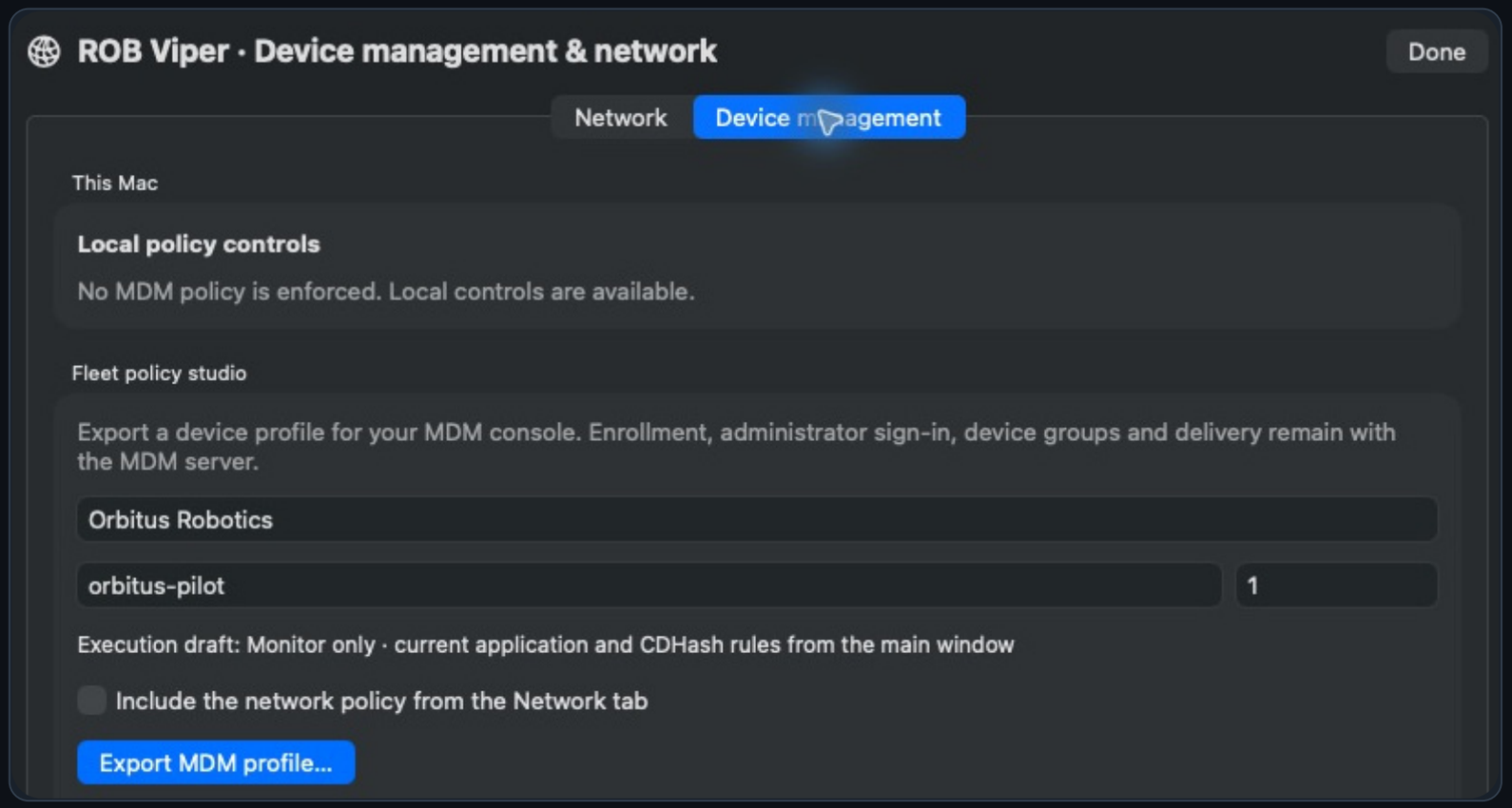
4. ### Recheck after your change

   Run Mac security posture again after macOS reports FileVault enabled. Viper should then report that protection as enabled; the screenshot in this guide remains a historical example.

FileVault protects data at rest. It does not stop malware already running in an unlocked user session. Apple's FileVault overview and setup guide explain the hardware distinction and recovery options; see sources [A1] and [A2] on page 35.

Manage a pilot fleet

Viper is the local security agent. Enrollment, administrator identity, device groups and profile delivery belong to your MDM server.



Device-management detail: this evaluation Mac has no forced MDM policy. Profile export is not evidence of enrollment or delivery.

Version compatibility

Application-control profiles require outer schema 4 and 0.6+ agents; use matching 0.8.1 components for this manual. The older 0.4/schema-2 hint in the legacy pane does not cover application-control policies.

1. **Read ownership first**

   Device management shows whether local controls are available or a forced policy controls the Mac.

2. **Prepare and export**

   Review the execution draft, organization, policy ID and revision in Fleet policy studio. Include network policy only when intended. Export MDM profile creates a file; it does not enroll this Mac.

3. **Deliver to a small pilot**

   Use your approved MDM workflow for extension approvals, Full Disk Access and the reviewed policy. Check the actual installed version and active service on each pilot device.

4. **Roll forward and roll back explicitly**

   Start in Audit. Increase revisions for reviewed changes, including a rollback to an allowing policy. Forced management can lock local Apply and Stop.

Start with the visible state

Resolve the narrow failing state before changing policy. Recheck the active mode and revision after any change.

Approved, but no network filtering

Setup & privacy may correctly say OFF. Approval permits the extension; Save & enable starts a reviewed configuration. Verify the running provider and fresh test connections.

Event list will not move

A selected event or Pause display can hold the view. Resume live updates clears selection. Check the text filter before assuming collection stopped.

Connected, but no events

Check Collecting, Active subscriptions and Full Disk Access. Trigger a harmless observed event. A service connection alone does not prove an active ES client.

A changed rule has no effect

Editing a draft does not apply it. Confirm Apply completed, the active mode/revision changed, and the event identity matches the rule. Existing processes or flows may continue.

A scan has zero findings but failed

Treat Failed, Incomplete, Cancelled and Timed out as unfinished evidence. Read Coverage and scan notes; resolve tools, databases, access or size limits.

A scheduled scan did not run

Check agent registration, Login Items approval, heartbeat, saved scope and idle/AC conditions. The Mac must be awake and logged in. Another scan/update can hold the job slot.

Saved policy cannot be read

Use Start monitoring to retry the saved policy read without resetting rules. Use Update extension if a newer bundled service is offered. Recheck the active state; an access failure is not a Full Disk Access verdict.

No hostname or map point

macOS may provide only an IP. PTR names do not grant trust. Map locations require an imported database and matching loaded history.

Keep the evidence useful and private

Viper exposes separate layers: live policy, event evidence, scanner results and system-setting checks.

NAVIGATE

Command-Shift-T

Open Threat Center.

Command-Shift-N

Open Network Control.

Command-Shift-V

Open Vulnerabilities & CVEs.

Quick tour

Open beside Execution policy for Capture, Audit, App rules, Enforce and Recover.

Resume live updates / Resume live

Clear a held process-event or connection selection.

Restore saved

Discard unsaved network-policy edits and reload the saved draft.

> **A repeatable daily routine**
>
> Review readiness and losses; inspect new behavior leads; scan new downloads before opening them; review dependency changes before release; export the records that support a decision.

DATA AND RETENTION

Execution events

Local rotating service logs: five files of 4 MiB. Bounded queues can drop telemetry; loss counters matter.

Network history

Opt-in local metadata with time, row and storage limits. No packet payloads or full URLs. Exports contain the loaded metadata.

Scan reports

Private per-user storage; last 20 reports, up to 1,000 findings each. Reports include paths, package/rule identifiers, versions, dates and coverage.

Behavior signals

Up to 200 alerts for the app session. Export useful alerts before closing the app.

External requests

Tool checks, CVE feeds and selected public source files contact their hosts. PTR queries use your resolver. Local inventory, files and reports are not automatically uploaded.

Before sharing: review exports and screenshots for local paths, identities, hostnames, IPs and information unrelated to the incident.

Browse reports across platforms

Open Vulnerabilities & CVEs with Command-Shift-V. This is a report library; the native Viper monitor still runs on macOS.



November 10 detail: platform, modified-date window, distribution/release and reported-version filters.

1. **Choose the platform and window**

   Select macOS, Linux or Windows and Last 7, 30 or 90 days. Updated means recently modified records, including older CVEs revised within that window.

2. **Narrow the reported release**

   Choose Distribution / release or Browse versions. The version field searches source text. Searching 6.1 does not evaluate whether 6.1 falls inside a stated range.

3. **Read the source and date**

   Select a CVE, inspect its report, original JSON, platform qualifications and advisory links. A full CVE ID can be looked up outside the rolling feed.

4. **Check coverage before conclusions**

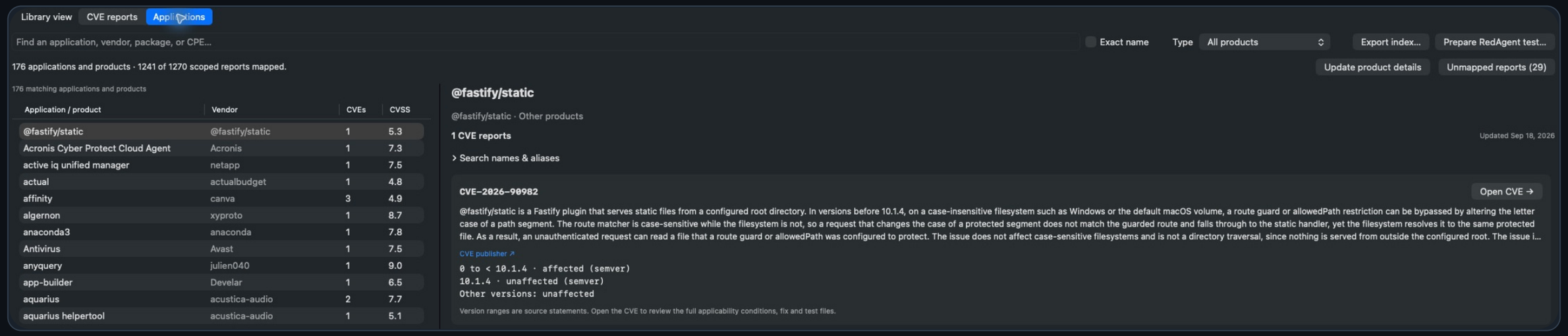
   Keyword feeds can miss records or include incidental mentions. Unknown releases remain visible. Failed refreshes retain the last complete cache and its date.

Platform grouping is evidence, not a host verdict

A CVE can belong to several groups; their counts are not additive. An application version is not an OS release. Review the vendor advisory and original conditions before deciding that a particular machine is affected.

Search the affected-product index

Applications lists products mentioned as affected in the selected library. It does not list software installed on this Mac.



November 10 detail crop: product search, mapping coverage, export, RedAgent handoff and reported ranges.

1 Search by identity

Enter product, vendor, package or CPE text. Exact name uses normalized aliases. Type separates applications, operating systems, hardware and other products.

2 Review every association

Select a product to inspect its CVEs, reported ranges and platform context. Open CVE returns to the report and source-file browser.

3 Inspect mapping gaps

Unmapped reports explains missing product metadata. Update product details continues or retries official CVE mappings; unmapped does not mean unaffected.

4 Export the actual scope

Export index retains the chosen platform, date window and release grouping. Prepare RedAgent test adds a pinned plan for a separate local inventory run.

What the hash table means

Keys normalize vendor/product names and aliases for fast lookups. They are not binary hashes. Only named affected products in loaded records are indexed; this is not every application or the entire historical CVE database.

Read tests and patches as evidence

Open a CVE, then Tests & patches. Public files are displayed as text; browsing does not execute tests or apply patches.

1 / ASSOCIATION

Follow the actual reference

Supported links include public GitHub commit patches, blob text and raw text files linked by the CVE. Viper does not guess a repository or invent a fix association.

2 / CONTENT

Distinguish diff from full file

Browse loads a selected file. A commit patch expands into changed-file diffs. Read full test/source file loads text at the referenced commit when available.

1. **Record provenance**

   Keep the source URL, retrieval date, byte count and SHA-256. Save copy exports the selected text; Export JSON preserves the original NVD record.

2. **Interpret TEST carefully**

   TEST labels come from filenames such as tests/ or *_test.go. They do not prove that a test reproduces this CVE, was executed or passed.

3. **Respect unavailable artifacts**

   Some advisories have no public patch or regression test. A deleted file or unsupported host remains a reference or explicit error.

4. **Plan verification separately**

   Read the vendor fix guidance, review the changed code and identify the applicable release. Any later test execution or patch deployment needs its own controlled workflow.

Read-only source browser

No shell, test runner, package installer or patch application is invoked by this window. A saved file is evidence to review; its hash identifies bytes and does not establish publisher trust.

Prepare a portable inventory test

Viper prepares the question and pins the lookup data. RedAgent collects installed-software metadata on the selected host.

VIPER

1. Select the catalog

In Vulnerabilities > Applications, choose the target platform, time window and optional release group. Review mapped and unmapped coverage first.

PORTABLE FILES

2. Save the test folder

Choose Prepare RedAgent test and a new folder. Keep plan.json, application-index.json and README.txt together when copying it to the target.

REDAGENT

3. Review on the host

Open a RedAgent build supporting application-inventory-v1 on the target computer. An IP in Network tests is unrelated to this local inventory.

1. **Keep the pinned bytes**

   The plan records platform and index SHA-256. Editing the index invalidates the plan. Regenerate a test when you need a newer catalog.

2. **Use the exact folder**

   Choose test uses the directory containing plan.json. A typo in a date suffix or selecting its parent causes No such file or directory.

3. **Confirm the host**

   In Installed apps, verify Expected hostname for the machine where RedAgent is running. Do not substitute a remote IP.

4. **Preview first**

   Choose Preview inventory test. Review platform, scope, pinned index and limits before choosing Run inventory test.

Preparation is not execution

Exporting the folder and previewing it do not inventory a host. The collector uses ordinary local access; it does not probe the network, launch inventoried apps, execute downloaded tests or apply patches.

Run the test and keep the evidence

Use Installed apps in RedAgent. A completed collection and an applicable vulnerability are separate conclusions.

1. ### Choose the evidence destination

   Set a private evidence folder. Existing cases are preserved; each run creates a unique case with plan and index snapshots.

2. ### Run the reviewed test

   Choose Run inventory test after the preview and host check. The collector reads macOS bundle metadata and OS details, Linux dpkg/rpm databases, or Windows uninstall registry entries.

3. ### Read outcome and coverage

   Review collector failures, missing metadata, cancellation and limits. Completed with gaps means the gaps still matter; it is not a clean verdict.

4. ### Open the full list

   All applications is the default view. It includes every collected entry, not only candidate matches. A small finding count is different from the inventory count.

5. ### Inspect a row

   Review installed version, publisher, source/path, candidate CVEs and reported ranges. Confirm product identity and installed-version applicability against the advisory.

6. ### Reopen the saved case

   After restarting, use Open saved result and select report.json inside the case folder. Keep inventory.json and the receipt beside it for integrity checks.

Saved evidence

The case contains inventory.json, report.json, plan/index snapshots, applicable raw collector output and a SHA-256 receipt. Hash verification detects changed bytes; it is not a publisher signature or proof of complete coverage.

The collector is bounded to 120 seconds, 50,000 inventory entries and 100,000 candidate/report associations. Cancellation retains partial evidence. It does not upload the inventory.

Filter the full application list

Resize the RedAgent window to expand the list and details. Search and filters operate on the loaded saved case.

SHOW

All applications

Every collected entry in that case. Catalog aliases are grouped into one installed-entry row; separate installation paths remain separate entries.

REVIEW

Potential matches

Names matched the pinned catalog. Installed versions are retained, but neither ranges nor vendor identity are automatically adjudicated.

COVERAGE

No index match / Not assessed

No index match means no name candidate in this catalog. Not assessed keeps unassessed entries visible. Neither status establishes safety.

1. **Combine useful filters**

   Search names, aliases, versions, publishers, paths or CVE IDs. Use Show and Source together to narrow the inventory.

2. **Sort and reset**

   Click a column heading to sort. Clear filters restores the full loaded list. Compare Showing with the total entry count.

3. **Explain a small result count**

   A case may have many installed entries and few candidate products. Check All applications before assuming the collector saw only the matches.

4. **Investigate missing software**

   Check collection roots, package-manager support, permissions, metadata and report limits. Portable apps or unsupported locations may not be covered.

Use counts as a navigation aid

For example, the earlier local validation collected 115 entries and two candidate products. That is historical engineering evidence, not a promised result or a list of confirmed vulnerabilities. Private host inventories are not included in this publication.

Sources, scope and support

This guide documents the installed 0.8.1 (18) development preview. It is not a claim of production antivirus certification or complete EDR coverage.

VERIFIED FOR THIS EDITION

The November 10 source revision passed 284 Swift tests, with four optional tests skipped, plus a separate live Homebrew source check. Earlier scanner and network figures are labeled historical controls. Current captures show the installed tools and CVE library; counts and versions are snapshots, not permanent guarantees.

This revision documents 0.8.1 (18) and the RedAgent 0.2.0 inventory workflow. Documentation work did not enable enforcement, change network policy, run downloaded tests, apply patches or enroll devices. Earlier policy screenshots retain their original historical captions.

Some images are labeled detail crops to keep controls readable. No fabricated alerts, rules, approvals or scan results are presented as live outcomes. The small purple screen-capture badge visible in some images belongs to the capture environment.

Reproduction and maintenance:
The private project contains SCANNING.txt, EXECUTION-CONTROL.txt, SECURITY-TESTS.txt, validation receipts, this manual generator and a screenshot hash manifest. Capture dates and example data are historical; recheck the app before a consequential action.

PRIMARY READING

[A1] Apple - Intro to FileVault
support.apple.com/en-gb/guide/deployment/dep82064ec40/web

[A2] Apple - Protect data with FileVault
support.apple.com/guide/mac-help/mh11785/mac

[A3] Apple - System Integrity Protection
support.apple.com/en-us/102149

[A4] Apple - Gatekeeper and runtime protection
support.apple.com/guide/security/sec5599b66df/web

ClamAV - Scanning and signature updates
docs.clamav.net/manual/Usage/Scanning.html

Trivy - Filesystem scanning
trivy.dev/latest/docs/target/filesystem/

YARA-X - CLI commands
virustotal.github.io/yara-x/docs/cli/commands/

ROB Viper product and release status
www.orbitusrobotics.com/products/rob-viper/

Support: orbitus@orbitusrobotics.com
Describe the issue first. Do not send passwords, recovery keys or unrelated private logs.