DAEMONCORE // EVIDENCE-LED OPERATOR DEVELOPMENT

ACADEMY

MISSION OS OPERATOR GUIDE

Diagnose the signal. Build the route. Prove the work.
A local-first operating guide for practical cyber training.

PRODUCT	RELEASE	EDITION
DaemonCore Academy	6.0 Beta 3	Latest release

PROGRESS IS RECORDED. CAPABILITY IS PROVEN BY THE WORK.

31 AUGUST 2026 // DAEMONCORE APPS

Document control

Field	Value
Document	DaemonCore Academy Mission OS Operator Guide
Product release	DaemonCore Academy 6.0.0-beta.3
Edition	Latest release
Platforms	Windows 64-bit; x64 Ubuntu and Debian-family Linux
Publisher	DaemonCore Apps
Classification	Customer training documentation
Last revised	31 August 2026

This guide covers the functionality shipped in `6.0.0-beta.3`, the current Latest release. Product behavior is authoritative when it differs from the guide. Preserve a progress export before upgrading a workstation used for active study.

Intended audience

- New operators building a defensible security method
- Working professionals expanding into a new security domain
- Team leads assigning structured practical development
- Instructors reviewing local-first, evidence-led training

Conventions

- **Mission OS** means the entry diagnostic, selected professional route, adaptive plan, and recorded progress.
- **Range** means a disposable, synthetic Docker environment controlled by the desktop app.
- **Evidence** means an observation, command result, artifact, or comparison tied to a stated objective.
- **Recorded** means completion stored in the current device's local operator record.
- **FieldOps** means the separately licensed authorization-bound assessment workspace.

Contents

1. Academy at a glance

Academy is organized around proof of work. Instruction introduces a method; an interactive workbench checks the decision; a range or case makes the operator gather evidence; a review explains what the result proves and what remains uncertain. Catalog counts describe separate surfaces that may overlap conceptually and must not be added into an invented lesson total.

Core workflow

1. Install the official build and confirm the visible version and platform.
2. Export any existing local progress before a beta upgrade.
3. Use the first-run guide to understand the Learn, Practice, Launch, and Prove stages.
4. Start in Academy or complete the 12-scenario Mission OS diagnostic to build a route.
5. Select one of six professional routes.
6. Follow the persistent next-action guidance rather than collecting random completions.
7. Preserve commands, raw signals, negative controls, and conclusions during practical work.
8. Complete the after-action review and assigned remediation.
9. Revisit the route dashboard and continue from recorded evidence.

Know when a command is live

Lesson pages teach method and contain in-app workbenches. Their command blocks are labeled **Reference command // not run here** and are worked examples; copying one does not execute it. Do not paste lesson examples into the Windows terminal merely because they look executable.

A command is live only after all three conditions are true:

1. You explicitly selected **Launch sealed range** from a mission briefing.
2. The app displayed **Containment verified** after the Docker preflight.
3. The terminal prompt begins with `root@dc-`.

The first mission defaults to Guided mode and exposes the exact runbook. After a lesson is mastered, its completion screen can open the matching Lab Range, Web Forge, or Enterprise Forge surface. The workflow dock remains available throughout the main app and its **How this works** control reopens the guide at any time.

Product map

Workspace	Purpose	Access
Mission OS	Baseline six domains and organize the next best work	Free Academy
Academy	Complete 28 core and 8 advanced practical lessons	Free Academy
Lab Range	Run seven sealed Docker missions	Free Academy
Web Forge	Complete 27 web/API modules against 22 sealed conditions	Free Academy
Enterprise Forge	Work 72 lessons across 48 professional cases	Free Academy
Drills	Rehearse eight focused decision sets	Free Academy
Mastery	Complete three principal capstones	Free Academy
FieldOps	Operate authorized diagnostics, campaigns, evidence, findings, and reports	FieldOps Pro license

2. Install and verify the release

Windows

1. Download `DaemonCore-Academy-Setup.exe` from the official Latest GitHub release.
2. Verify the installer against `SHA256SUMS-windows.txt`.
3. Close any running DaemonCore Academy window.
4. Run the installer. It uses the stable application identity and should replace the prior installed version.
5. Launch the app and confirm the footer shows `6.0.0-beta.3` and `WINDOWS`.

The current installer is not Authenticode-signed. Windows may show **Unknown Publisher**. That warning is expected for this release but is not a reason to skip checksum verification. Do not distribute it through a managed production fleet as a trusted signed package.

Linux

Use the AppImage for a portable file or the Debian package for an installed desktop entry.

1. Download `DaemonCore-Academy-6.0.0-beta.3.AppImage` or `DaemonCore-Academy-6.0.0-beta.3.deb`.
2. Verify the selected package against `SHA256SUMS-linux.txt`.
3. For AppImage: run `chmod +x DaemonCore-Academy-6.0.0-beta.3.AppImage`, then launch it as the desktop user.
4. For Debian: run `sudo apt install ./DaemonCore-Academy-6.0.0-beta.3.deb`.
5. Confirm the footer shows `6.0.0-beta.3` and `LINUX`.

The supported beta matrix covers x64 Ubuntu and Debian-family desktops. Compatible derivatives may run, but they are outside the first beta support matrix.

Upgrade and rollback preparation

- Export the current operator record from Settings before upgrading.
- Let any live range stop before closing the old build.
- Keep the prior installer or AppImage until the new build opens the record successfully.
- Do not delete the per-user application-data directory during a normal upgrade.
- A newer Debian package upgrades in place; AppImage users replace the old file manually.

The installer replaces program files but retains the version-independent per-user record. Windows stores the durable record beneath `%APPDATA%\daemoncore-academy`; Linux uses the `daemoncore-academy` directory beneath the desktop user's configuration root. Do not remove that directory during an upgrade.

Release 6.0.0-beta.3 also keeps a recovery mirror. If an affected earlier package recorded progress through browser fallback storage and no durable profile exists, the app migrates that recovery record into the native desktop store on first launch. An existing durable profile always remains authoritative.

3. Prepare Docker and protected storage

Docker is required only for live disposable ranges. Curriculum review and simulation fallback remain available when Docker is unavailable.

Verify Docker

On Windows, start Docker Desktop and wait for the engine to report ready. On Linux, run both commands as the same unprivileged desktop user that launches DaemonCore:

1. Run `docker version`.
2. Run `docker compose version`.
3. Open a Lab Range mission and choose the diagnostics control before launch.
4. Resolve every failed preflight check; do not launch DaemonCore as administrator or root to bypass socket access.

Linux membership in the `docker` group is effectively root-level host access. Use a dedicated training workstation or VM when that trust level is inappropriate.

Range safety contract

Live mission definitions must pass the app's range policy before Docker receives a launch request:

- Internal-only range network
- No host filesystem mounts
- No privileged containers
- Egress denial
- Resource limits
- Declared services and deterministic objectives
- Controlled teardown on exit

Protected credential storage

Academy progress does not require a keyring. FieldOps activation and operator identity enrollment do. Windows uses the operating-system credential context. Linux requires an unlocked GNOME Keyring or KWallet backend. DaemonCore blocks protected writes when Electron reports `basic_text`.

Never launch with `--password-store=basic` or `--no-sandbox`.

4. Understand the curriculum map

DaemonCore separates instruction, rehearsal, live practice, professional casework, and final proof so that a completion can carry a clear meaning.

Shipped breadth

Surface	Shipped work	Primary evidence
Core Academy	28 practical lessons	Workshop decisions, operator artifact, knowledge validation
Advanced Academy	8 advanced lessons	Higher-complexity analysis and validation
Lab Range	7 sealed Docker missions	Commands, raw signals, objective submissions, sealed result
Web Forge	27 evidence-led modules	Baseline/comparison pair, boundary statement, control recommendation
Web conditions	22 sealed lab conditions	Deterministic synthetic vulnerability or trust failure
Enterprise Forge	72 pathway lessons	Case analysis across 48 enterprise cases
Drills	8 focused sets	Repeatable tactical decisions
Mastery	3 principal capstones	Integrated, evidence-backed operator decision

Six operating domains

- Scope & Safety
- Network Analysis
- Web & API
- Identity
- Cloud & Supply Chain
- Evidence & Detection

Progress is stored locally. Mission OS does not compare the operator with an invented cohort, assign a fake industry rank, or promise employment readiness.

5. Establish a Mission OS baseline

The entry diagnostic presents 12 scenario decisions across the six domains. It tests judgment, not trivia, and explains the signal after each response.

Take the diagnostic

1. Open **Mission OS** from the main navigation.
2. Read the scenario and the operational constraint before choosing.
3. Select the safest defensible action that still answers the stated objective.
4. Read the rationale even when the answer was correct.
5. Continue until all 12 decisions are recorded.
6. Review the six domain scores and recommended route.

Interpret the result

- Treat a low domain score as a routing signal, not an identity or ceiling.
- Check whether an incorrect response confused observation with inference, skipped a negative control, expanded scope, or ignored evidence custody.
- Retake only after completing meaningful remediation; repeating immediately measures memory of the prompt.
- A route recommendation is derived from diagnostic scores and route weights. The operator may select a different route.

OPERATOR NOTE // The diagnostic is not a certification, psychometric instrument, percentile, or employer assessment.

6. Select and operate a professional route

Available routes

Route	Professional emphasis
Penetration Tester	Scope, enumeration, validation, evidence, and defensible reporting
Web & API Specialist	Browser trust, server interpreters, identity, object authorization, and protocol behavior
Identity Security	Authentication, authorization, directory and token trust, and privilege pathways
Cloud Security	Cloud control planes, workload identity, supply chain, and evidence-driven configuration review
Detection & Response	Telemetry, timelines, hypotheses, validation, containment decisions, and review
Security Engineer	Controls, architecture, verification, automation boundaries, and resilient operations

Operate the plan

1. Select a route after reviewing both the recommendation and the role description.
2. Open the first stage marked **Next best action**.
3. Complete its required recorded work.
4. Return to Mission OS and confirm route progress changed.
5. Review completed stages when preparing for a capstone or team discussion.
6. Change routes deliberately; recorded work remains but the active sequence changes.

Route progress is calculated from the current device's completed Academy, Lab Range, Web Forge, Enterprise Forge, and Mastery work. It is not based on time spent with a page open.

7. Complete tactical lessons and workbenches

Each tactical lesson is designed to move from explanation to an operator artifact.

Lesson sequence

1. Read the operator outcome, objectives, prerequisites, environment, and boundary.
2. Work each guided section in order.
3. Study each reference command as a worked example; the lesson page does not execute it.
4. Compare the worked example with the stated expected signal.
5. Explain what the signal would prove and what it would not prove.
6. Complete the in-app operator exercise and its success criteria.
7. Pass the interactive workbench.
8. Complete the knowledge validation to record mastery.
9. Choose whether to return to the pathway or apply the method in the matching live range.

Evidence standard

For every practical, preserve four things:

- **Command or action:** the exact reproducible step
- **Raw signal:** the relevant unedited response or artifact
- **Interpretation:** the smallest conclusion directly supported by the signal
- **Boundary:** what was not tested, observed, or proven

Copying a command is not the lesson. The work is selecting the right control, recognizing the signal, and producing a reviewable conclusion.

8. Run disposable Docker missions

The seven sealed missions provide live synthetic systems without granting access to external targets. Three flagship missions receive seeded professional case emphasis in Mission OS.

Flagship cases

Mission	Focus	Required operator move
The Ghost Port	Network and service analysis	Compare the declared inventory with the observed listener and document the exposure
Broken Trust	Web/API object authorization	Hold identity constant, compare one approved cross-tenant object, and prove the missing decision
Night Shift	Detection and forensic reasoning	Validate the evidence manifest, build a timeline, and submit an evidence-backed hypothesis

Launch procedure

1. Read the brief, objectives, boundary, expected services, and stop conditions.
2. Run mission diagnostics and resolve Docker or policy failures.
3. Launch the range and wait for **Containment verified** and a `root@dc-` prompt before entering commands.
4. Work from the mission objective; use help as reference, not as the completion path.
5. Preserve commands and raw outputs before submitting each objective.
6. Stop when the objective is proven. Do not explore unrelated container behavior.
7. Complete the mission, export the integrity receipt if needed, and destroy the range.

Recovery

If a range stalls, return to diagnostics, inspect Docker readiness, then stop and relaunch the mission. Do not manually alter its Compose definition or attach additional networks to force it to run. Treat any unexpected host mount, public egress, or privileged container as a stop condition.

9. Work through Web Forge

Web Forge contains 27 evidence-led modules backed by 22 sealed conditions. It emphasizes trust decisions and reproducible comparisons rather than payload collection.

Standard method

1. Read the condition, objective, boundary, signal, and control guidance.
2. Capture a positive baseline using the designated synthetic identity or object.
3. Change one approved variable for the comparison.
4. Capture the response without enumerating unrelated identifiers or data.
5. State the trust decision that failed or held.
6. Recommend a control at the point where the server should make that decision.
7. Record the artifact requested by the module.

Review questions

- Did the comparison change exactly one decision variable?
- Is the identity, tenant, object, state, and action visible in the evidence?
- Does the conclusion distinguish authentication from authorization?
- Is the recommended control enforced server-side at every relevant path?
- Would a reviewer reproduce the result without guessing hidden prerequisites?

Web Forge fixtures and inert markers are for the sealed training environment. They are not a license to reproduce the same request against a public service.

10. Work through Enterprise Forge

Enterprise Forge contains 72 pathway lessons organized around 48 professional cases. The case is the unit of judgment: each lesson should contribute to a defensible picture of ownership, trust, consequence, and control.

Case workflow

1. Identify the system owner, business purpose, trust boundary, and available evidence.
2. Separate confirmed facts, credible hypotheses, and unknowns.
3. Select the next action that reduces the most important uncertainty without exceeding the case boundary.
4. Record both the expected signal and the result that would disprove the hypothesis.
5. Link the technical condition to a realistic business or control consequence.
6. Produce a recommendation with an owner and a measurable verification step.

Team review

For group use, assign one operator to present the case, one to challenge evidence sufficiency, and one to challenge remediation. Rotate those roles. A case is complete only when the team can name the weakest inference and the next piece of evidence that would change the decision.

11. Prove mastery with capstones

The three principal capstones integrate multiple domains. They are not additional tutorials; they test whether the operator can choose and defend a method under incomplete information.

Before starting

- Complete the relevant route stages.
- Review the latest after-action remediation.
- Prepare a clean evidence notebook or approved local case template.
- Budget time for evidence review and the final decision, not only technical execution.

Capstone standard

A defensible submission contains:

- Scope and objective in one paragraph
- Evidence inventory with source and time
- Reproducible sequence of important actions
- Positive and negative controls where applicable
- Findings separated from hypotheses
- Business or operational consequence grounded in the case
- Remediation with a measurable retest condition
- Explicit residual uncertainty

Completion records capability demonstrated in the shipped exercise. It does not create a third-party certification or guarantee equivalent performance on an unknown production system.

12. Read the after-action review

Completed adaptive ranges produce a sealed after-action review across four dimensions.

Dimension	What it measures	Improvement move
Evidence coverage	Whether required objectives are supported by accepted evidence	Revisit missing objective signals and preserve the smallest sufficient proof
Operator independence	How much guidance was required	Repeat without hints while keeping the same evidence threshold
Method discipline	Whether the sequence respected boundary and validation controls	Write the planned baseline, comparison, stop condition, and conclusion first
Time discipline	Whether the operator moved deliberately within the exercise window	Remove dead-end actions; do not trade away evidence quality for speed

Use the debrief

1. Read the overall readiness signal and every dimension, not only the operation score.
2. Review the recorded case emphasis or seed.
3. Perform the concrete remediation before attempting a higher-autonomy variant.
4. Follow the assigned next action in the active Mission OS route.
5. Export the digest-sealed receipt when the result will be reviewed outside the workstation.

The result preserves the case variation, debrief data, and recorded attempt. It must not be represented as a live-target penetration-test result.

13. Preserve progress and integrity receipts

DaemonCore is local-first. It does not automatically synchronize progress to a DaemonCore cloud account.

Before an upgrade or device change

1. Open Settings and export the current operator record.
2. Store it in an approved encrypted location.
3. Record the application version and platform used for the export.
4. Export important mission receipts separately when another reviewer needs their digests.
5. Install or upgrade the new build, then confirm the record opens and route progress is intact.

Existing records migrate to schema version 6 without intentionally discarding earlier completion. An export is still required before beta upgrades because it is the supported recovery point.

Receipt interpretation

A digest can show that the exported receipt content has not changed since it was sealed. It does not prove the operator's real-world identity, validate an employer claim, or establish that an external system was tested.

Do not hand-edit local application data to change scores, completions, or receipts. If integrity or migration behavior is unexpected, preserve the original export and stop before overwriting it with repeated imports.

14. Troubleshooting

Docker is installed but the range will not launch

1. Confirm `docker version` and `docker compose version` succeed for the same desktop user.
2. On Windows, wait for Docker Desktop to report ready.
3. On Linux, resolve socket or group access without launching DaemonCore as root.
4. Run mission diagnostics and read the exact failed policy or readiness check.
5. Stop stale range containers through the app before retrying.

Linux AppImage does not launch

Confirm execute permission. Use the Debian package when the distribution disables AppImage execution. Do not bypass Electron protections with `--no-sandbox`.

Protected storage is unavailable

Academy progress should remain usable. FieldOps activation and operator identity require secure storage. On Linux, unlock GNOME Keyring or KWallet and reopen the app. Do not use the `basic_text` backend.

Progress or route appears stale

Confirm the footer version, finish the current exercise through its completion control, return to Mission OS, and review the selected route. Imported or migrated data should be checked against the pre-upgrade export.

The app asks for an operator name after restart

1. Confirm the footer shows `6.0.0-beta.3` or newer.
2. Close the app and reopen the installed Start menu or desktop shortcut, not the setup executable in Downloads.
3. Confirm the per-user application-data directory still exists and was not removed by a cleanup tool.
4. If an earlier fallback record exists and the durable record is empty, allow the first beta.3 launch to complete its automatic migration.
5. If the profile still does not appear, preserve the application-data directory and any prior Settings export before reinstalling or importing.

A mission completed after only copying text

Use the latest `6.0.0-beta.3` build, restart the mission, and complete its objective submissions and after-action review. Help text is reference material. Report any route that records completion without accepted objective evidence, including the mission name, platform, version, and exact sequence.

Support package

Provide the visible app version, platform/distribution, package type, Docker versions, mission or lesson name, and exact error. Remove license keys, protected application data, customer information, and unrelated host details.

15. Operating plans for individuals and teams

Individual weekly cycle

1. **Baseline:** open Mission OS and select one next best action.
2. **Instruction:** complete one practical lesson and its operator artifact.
3. **Live work:** complete one range, Web Forge module, or Enterprise case.
4. **Review:** write the weakest inference and compare it with the after-action review.
5. **Remediate:** repeat one task with less guidance or a stronger negative control.
6. **Record:** export progress at a meaningful milestone.

Team cohort cycle

- Assign the same case but require independent evidence notes.
- Compare decisions before revealing the expected signal.
- Review scope, evidence sufficiency, conclusion strength, and retest design separately.
- Track completed route stages, not time in the application.
- Do not publish a DaemonCore-derived “certification” unless DaemonCore Apps explicitly creates and governs one.

Instructor review checklist

- The operator can reproduce the result without hidden steps.
- Every conclusion traces to a raw signal.
- Negative controls or disconfirming evidence are addressed.
- The stated boundary matches the actual work.
- Remediation names an owner and measurable retest.
- Guidance use and residual uncertainty are discussed honestly.

16. Product boundaries and quick reference

Honest boundaries

DaemonCore Academy teaches and records work in synthetic, local-first environments. It does not provide access to external targets, replace written authorization, verify professional identity, guarantee employment, or create an industry certification. Mission OS signals are derived only from recorded work on the current device.

FieldOps is the place for authorization-bound assessment operations. Its license unlocks the workspace; a signed engagement authorizes the target. Chaos Engine is bounded resilience sampling, not a DDoS tool.

Quick reference

Control	Shipped value
Mission OS diagnostic	12 scenarios across 6 domains
Professional routes	6
Core practical lessons	28
Advanced lessons	8
Sealed Docker missions	7
Web Forge modules	27
Sealed web conditions	22
Enterprise pathway lessons	72 across 48 cases
Drill sets	8
Principal capstones	3
After-action dimensions	4
Progress storage	Local operator record, schema version 6
Live range requirement	Docker Engine with Compose v2
FieldOps access	Separate FieldOps Pro license

Release links

- Release page: `https://github.com/DaemoncoreApps/DaemonCore-Academy/releases/tag/v6.0.0-beta.3`
- Windows installer: `https://github.com/DaemoncoreApps/DaemonCore-Academy/releases/latest/download/DaemonCore-Academy-Setup.exe`
- Linux AppImage: `DaemonCore-Academy-6.0.0-beta.3.AppImage`
- Linux Debian package: `DaemonCore-Academy-6.0.0-beta.3.deb`
- Product site: `https://academy.daemoncore.app`