Next-generation cyber defense

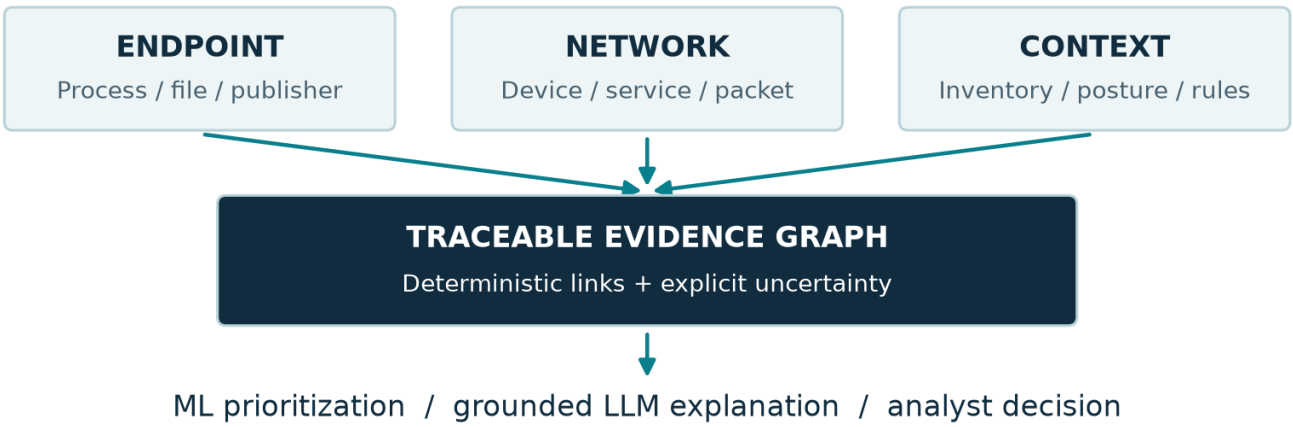
LLMs, machine learning, and Apple Endpoint Security: an architecture and competitive assessment of SignalAtlas and BlackViper

OR-WP-2026-005 | Version 1.0 | November 7, 2026

Orbitus Robotics · Project lead: Rodolfo Aramayo

Abstract

The next generation of cyber defense should connect observations before asking a model to explain them. This paper assesses SignalAtlas and BlackViper against network scanners, protocol analyzers, endpoint controls and fleet platforms, then proposes an evidence-preserving architecture around Apple's Endpoint Security framework. Deterministic correlation links local process identity, connection metadata, packet records and time-bounded device observations. Machine learning ranks behavioral leads; retrieval-grounded large language models (LLMs) explain evidence and uncertainty. Both operate outside authorization callbacks. Recent provenance-learning and security-log research informs the design, but does not establish detection performance on this suite. A frozen source baseline, explicit deployment limits and measurable acceptance gates distinguish functioning components from planned integrations.



Conceptual target architecture. Correlation and learned analysis are proposals in this edition; the source baseline already supplies several underlying sensors and import paths.

Technical white paper - Not peer reviewed. Sources checked 2026-11-07.

Keywords: Endpoint Security, SignalAtlas, BlackViper, network forensics, LLM, machine learning, provenance graphs

1. Executive direction and evidence boundary

> The suite is a developing investigation workbench. Its strongest opportunity is a traceable path from a discovered device or suspicious flow to the local process, publisher and evidence that justify an analyst decision.

SignalAtlas combines wireless observations, bounded native LAN discovery, imported network evidence and a graphical TShark workflow. BlackViper contributes Endpoint Security events, execution-policy controls, a separate Network Extension and integrations with established file and dependency scanners. These components are useful together, but their presence does not establish full Nmap, Wireshark, Jamf or enterprise endpoint detection and response (EDR) equivalence. [1]

This edition freezes the inspected implementation at SignalAtlas commit 30a867890f6e and BlackViper commit 4314a0d82219. It draws on their checked-in documentation and validation records. Implementation means code exists at that baseline; recorded validation means a specific test is documented; live acceptance means the actual permissioned workflow has been exercised. Publication of this paper is separate from product distribution. The public supplement identifies source paths relative to each project, complete commit IDs and evidence hashes. [1]

Decision	Architectural consequence
Correlate before generating explanations	Use deterministic joins, source IDs and uncertainty labels. An LLM must not invent missing relationships.
Preserve captured evidence	Retain raw and decoded values with categories, explicit collection limits and restricted access. A feature vector is a derived view.
Separate decision paths	Keep OS authorization deterministic and bounded. Run inference, enrichment and investigation asynchronously.
Measure utility before complexity	Establish simple baselines, held-out evaluation and analyst workload metrics before deploying graph or foundation models.

The research review is selective, current to the edition date, and focused on provenance learning, network representations and LLM log reasoning. It is not a claim that one model or vendor leads every security task. Performance on an external benchmark is a research result, not a measured capability of this application suite. [22,23,24,25,26]

2. What the inspected suite delivers

Area	Implemented or recorded	Boundary at this baseline
Wireless survey	Wi-Fi and BLE observations, history and receiver-location mapping.	Radar angles are illustrative. Receiver location does not identify transmitter position.
LAN discovery	IPv4 TCP connect scan, selected ports, PTR names, available ARP-cache MACs, CSV and saved baselines.	Explicit /22 to /32 scopes; no complete IPv6, ICMP or UDP discovery engine. Silence is not absence.
Network evidence	Nmap XML and Suricata EVE imports; TShark saved-capture analysis with filters, protocol records and categorized values.	Imports do not mean the suite executed Nmap or operated a Suricata sensor.
Live capture controls	Interface selection, BPF filter, start/stop and capture limits; PCAPNG retained for later analysis.	Permissioned live capture and native Installer acceptance were pending. Capture and decoding are sequential.
Endpoint and policy	Process/file events; Monitor, Audit and explicit Enforce modes; publisher, app and build identity rules.	Recorded live execution test exercised Audit, not actual denial. Publisher trust/Lockdown require macOS 26+.
Scanning and lifecycle	ClamAV, YARA-X, Trivy and posture workflows; per-user scheduled scan agent.	The Mac must be awake and logged in. Scheduled file scans are distinct from continuous behavioral EDR.
Cross-app workflow	Persistent evidence reports and compatible import/read paths.	Transfer is manual; automatic correlation and learned ranking are proposed in this edition.

The live-capture milestone records 56 SignalAtlas tests and three cross-repository checks passing. The preceding full-retention milestone records 220 BlackViper tests. Synthetic fixtures and loopback exercised retention, cancellation and report exchange. A real loopback capture instead reached BPF permission denial. These records demonstrate bounded software behavior, not production capture acceptance or detection accuracy. [1]

The Workbench bundle is an ad-hoc-signed development artifact. Its external capture worker is excluded from the App Store target. BlackViper documentation also states that production distribution remains pending. Installing an extension, compiling a target or listing network interfaces does not by itself establish a working capture or enforcement session. [1]

3. Competitive map: discovery and packet analysis

The following comparison uses official feature descriptions, not a benchmark of installed competing products. Partial overlap describes the suite at the frozen baseline. The next deliverables should close specific workflow gaps rather than claim a complete replacement. [7,8,12,13,14,21]

Tool / purpose	Documented capability	Suite overlap and remaining work
NetAdmin Pro LAN administration [13]	Device discovery, identity signals, notes/tags, scheduled monitoring, loss/jitter, SNMP/MIB and network diagnostics.	Native TCP inventory and baseline comparison overlap. Add IPv6, Bonjour/UPnP, durable identities, monitored history and SNMP before broad replacement.
Nmap Active assessment [12]	Host discovery, multiple port-scan methods, service/version and OS probing, NSE and traceroute.	XML import retains richer reported results. The native scanner is narrower. A future launcher needs explicit profiles, job provenance and integration/distribution review. [28]
Wireshark / TShark / dumpcap Packet investigation [7,8,14]	Capture and protocol dissection, capture/display filters, multi-file capture options and stream-oriented investigation.	The suite orchestrates dumpcap and TShark; these remain the engines. Add progressive decoding, stream views, rotation, multi-interface handling and decryption configuration.
NetSpot Wireless planning [21]	Site surveys, coverage heatmaps and wireless planning.	SignalAtlas provides observations and receiver tracks. Floorplan calibration and validated survey/planning workflows remain separate deliverables.

A usable device record should distinguish observed address, time range, interface, hostname source, MAC/OUI source, reported vendor/model, probe method, port state, service evidence and operator notes. Imported OS guesses stay labeled as guesses. A port-number label is not a product fingerprint, and an open TCP connection does not establish a vulnerability. [1,12]

> The practical near-term goal is a graphical investigation workspace that controls and explains established engines while preserving their original outputs. Tool integration is valuable without implying feature or performance parity.

4. Competitive map: endpoint and enterprise operations

Tool / purpose	Documented capability	Position of the Orbitus suite
Jamf Pro [10]	Apple device enrollment, inventory, configuration, application lifecycle and fleet management.	Managed preferences and profile export provide deployment building blocks. No Jamf API connector, enrollment service or fleet console is claimed.
Jamf Protect [11]	Endpoint/network protection, custom detections and policies, maintained threat detection and response.	Endpoint and network telemetry plus scanner integrations are foundations. Detection maintenance, fleet operation and response qualification remain major gaps.
Santa [15]	macOS binary/file access authorization, policy modes and execution event records.	Viper has publisher/app/build rules and Audit. It lacks full Santa policy/synchronization parity; Monitor mode semantics also differ.
Little Snitch [16]	Application connection alerts, allow/deny decisions and network visibility.	Per-app network rules and connection history overlap. The proposed differentiator is correlation with endpoint and packet evidence.
Suricata / Security Onion [17,18]	IDS/IPS and network security monitoring; centralized network/host logs, hunting and case workflows.	EVE import is useful interoperability. It is not an operating IDS/IPS sensor, distributed collector or centralized SOC platform.
Wazuh / Nessus [19,20]	Wazuh provides agent/log, inventory, vulnerability and response workflows; Nessus provides network vulnerability assessment.	Selected dependency scanning and local posture checks cover a narrower problem. Broad host/app CVE inventory and validated network findings need additional engines and data.

Jamf Pro and Jamf Protect solve different jobs and should remain distinct in requirements. A proposed Jamf connector should first enrich assets with authenticated inventory, enrollment and policy state, preserving tenant, device ID, observation time and source revision. Response actions should be separately scoped and reviewed. The Endpoint Security entitlement alone does not provide device management. [5,10,11]

The product opportunity is an accessible macOS investigation surface: users can move from a network observation to its evidence, local application identity and candidate policy. Sustained competition in enterprise protection additionally requires fleet deployment, update/rollback operations, incident workflows, detection upkeep and measured reliability. These are roadmap commitments to evaluate, not delivered capabilities.

5. Apple Endpoint Security: two execution paths

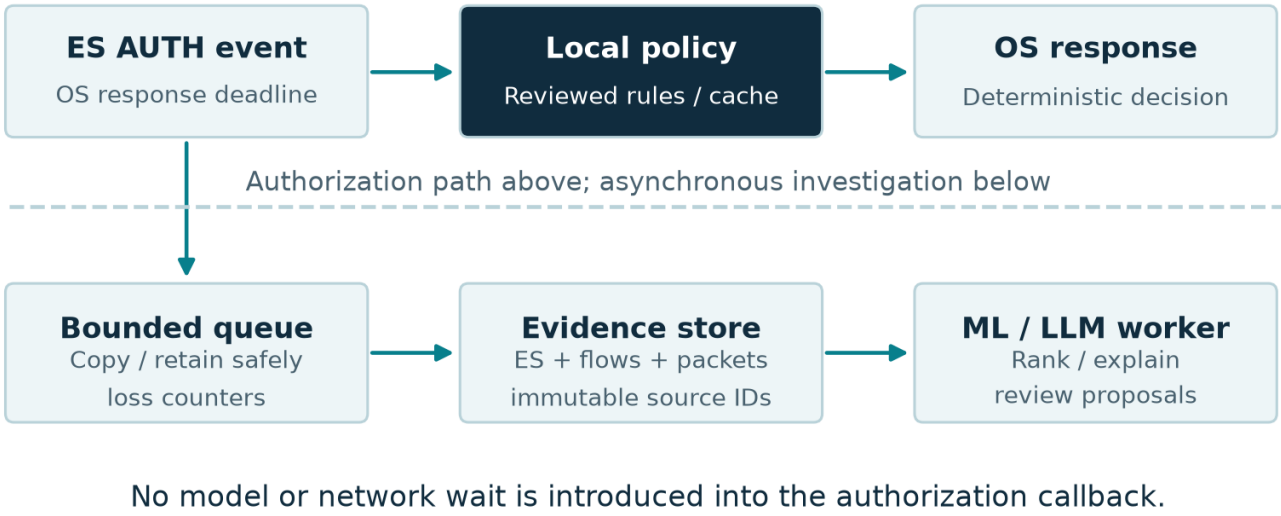


Figure 1. Proposed separation of time-critical authorization and asynchronous analysis. Existing framework integrations supply inputs; the complete analysis plane is future work.

Apple Endpoint Security is a C API for observing system events and authorizing supported pending operations. A client receives notifications and authorization requests; it can be packaged in a system extension installed through the host app. This is the framework intended by the phrase low-level security endpoint framework. It is distinct from Linux eBPF and from macOS BPF packet devices. [2]

An authorization message carries a deadline. Missing it can terminate or restart a client, and repeated misses can prevent new client connections. The proposed callback uses reviewed local rules and prepared state, with a documented fallback. It never waits for an LLM, cloud service, DNS lookup or analyst. Cache use and invalidation must match each event API and policy revision. [3]

Copy or retain messages using supported APIs before processing beyond their documented lifetime; keep queues bounded and release retained resources. Record sequence gaps, queue losses, subscription changes and restarts as coverage evidence. Absence of a suspicious event during a gap is not evidence of safety. Apple exposes per-type sequence numbers for detecting kernel drops. [4,5]

Client entitlement and TCC/Full Disk Access are separate deployment requirements. Network Extension supplies a separate flow/filter path; dumpcap uses packet-capture access. ChmodBPF is Wireshark's supported macOS capture-permission mechanism, with administrative installation. The app should not collect an administrator password itself. Granting packet access does not grant Endpoint Security authorization rights. [5,6,9]

6. Automatic correlation with explicit uncertainty

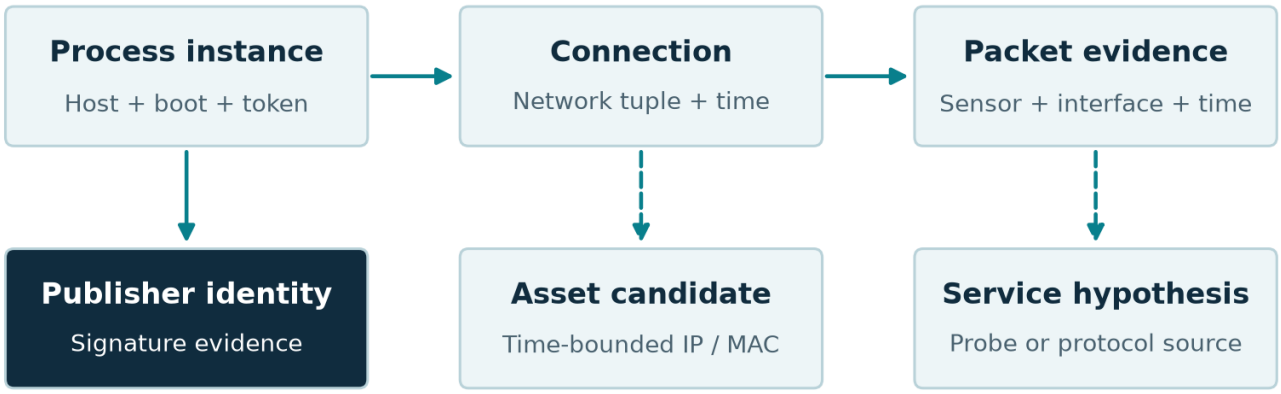


Figure 2. Proposed relationship types. A shared address is an observation, not sufficient proof of device identity or causality.

Use an append-only observation model: case and sensor IDs, source kind/version, capture or import settings, record locator, source hash, event and ingest times, clock basis and quality flags. Store derived links separately with join method, candidate count, time tolerance and the exact supporting observations. A hash detects changed bytes; it does not authenticate the original source by itself.

For a local flow-to-process join, require a compatible observation domain and protocol, both endpoint addresses and ports, temporal overlap within a justified clock tolerance, and a process-instance identity. Viper already records connection/application fields useful for this work. A PID alone is reusable; pair it with a boot/session and audit-token or process-start identity where available. Preserve actual process and responsible application as separate identities. [1]

Maintain competing candidates when DHCP reuse, NAT, VPN translation, proxies, reused ports or uncertain clocks make a join ambiguous. Do not invent a confidence percentage: use direct, inferred, ambiguous and unknown states until calibrated probabilities exist. Remote packets do not identify the remote process without remote endpoint telemetry. MAC randomization and BLE identifiers further limit device fusion.

> Example, not an observed incident: a persistence write, a new executable instance and an outbound flow share a defensible local identity/time chain. The interface can show all three records and packet references. If VPN translation breaks the tuple match, it should show an unresolved link rather than attribute the flow to that process.

A correlation service must read the retained records, not only the UI's packet/flow summary. It must also carry the acquisition scope into every case so filters, truncation, missing packets and time bounds remain visible. At the assessed baseline, shared report provenance does not yet carry every live-acquisition setting. [1]

7. Full evidence retention and useful data categories

The baseline preserves available decoded values and source bytes within explicit limits. Authentication values, cookies, query data and payload fields are categorized rather than omitted. Network access remains enabled for the analysis worker; optional name resolution is a recorded setting. A model can consume selected features without altering the underlying evidence. This is a distinction between storage and analysis inputs, not a redaction policy. [1]

Evidence layer	Retain and categorize	Interpretation boundary
Acquisition	Raw PCAPNG, interface, filter, snapshot length, duration/size/packet limits, exit status and reported drops.	Only traffic visible to that interface and within capture settings exists in the artifact.
Decoded protocol	Frame locators, timestamps, protocol trees, repeated fields, addresses, ports and raw byte fields.	Decoder versions, encryption, truncation and malformed traffic constrain interpretation.
Content and identity	Authentication/session values, URLs/queries, payload categories, certificates and available service evidence.	A category describes the data; it does not establish that a credential is valid or a service is malicious.
Endpoint and posture	Process lineage, file activity, code identity, flow decisions, scanner findings and configuration observations.	Retain actual source and time. A signature identifies an assessed signer/build, not benign intent.
Derived analysis	Join records, feature definitions, model/version, prompts, retrieved IDs, output and analyst disposition.	Keep observations, hypotheses and human decisions distinguishable and reproducible.

Current private log directories and files use restrictive filesystem modes. The proposed evidence service adds explicit retention/capacity policy, protected indexing, access audit and export provenance. The existing capture worker is bounded: decoded output is capped, report/record counts are limited, and the displayed summary is smaller than the retained record set. Exceeding a limit must create an incomplete or failed status, never a false all-clear. [1]

Capture filters select what is acquired; display filters select what is analyzed or displayed. Neither decrypts encrypted payloads. A normal host interface does not necessarily observe other devices' unicast traffic on a switched network. Enterprise network visibility requires appropriate sensor placement, which is a deployment design separate from adding an app control. [7,8]

> Complete retention means preserving what the configured sensor actually acquired, with limits and gaps recorded. It cannot mean every byte on every network or plaintext hidden by encryption.

8. What recent machine-learning research changes

Several recent results support a staged research program. Each entry below describes the authors' work and our architectural inference separately. These studies use different data and metrics; their scores cannot be ranked as one product leaderboard.

Research / status	Finding reported by the authors	Implication for this suite
Bilot et al. USENIX Security 2025 [22]	A unified study of eight provenance intrusion-detection systems identifies practical shortcomings; a simple neural model reaches state-of-the-art detection on five of seven DARPA datasets.	Start with reproducible simple baselines and operational metrics. More graph complexity is not evidence of better deployment behavior.
DUPIN USENIX Security 2026 [23]	Unsupervised pretraining on provenance graphs is followed by few-shot attack learning. The study uses 7.3 TB of audit data and evaluates 25 campaigns across four data sources.	Explore graph representations only after the event schema and labels are trustworthy. Few-shot adaptation still needs attack evidence and local validation.
Traffic-MoE July 2026 v2 preprint [26]	Sparse expert routing targets traffic representation and inference efficiency across multiple security tasks. The paper reports throughput and latency gains in its setup.	Treat it as a candidate representation approach. Verify weights, license, feature requirements and Apple hardware behavior before claiming deployability.
SecLog-Bench October 2026 preprint [25]	A 29-scenario multi-source benchmark evaluates eight LLMs. Its best reported accuracy is 44.44%; correct predictions can still contain faulty reasoning.	Score evidence support and attack-stage reasoning separately from final labels. Model fluency is not an acceptance metric.

The proposed learning ladder starts with transparent rates, counts and temporal rules, then compares tabular and small sequence models. Only afterward should self-supervised graph embeddings or traffic representations compete for promotion. Candidate features include process ancestry, publisher novelty, service changes, destination rarity, flow timing and file-activity bursts. Novelty is a lead to investigate, not a malware verdict.

Core ML is one potential on-device runtime for compatible models and can use CPU, GPU and Neural Engine resources. That does not guarantee conversion or useful performance for a particular research model. Benchmark the actual model, precision, input pipeline, memory and energy budget on target Macs. Local inference and network-enabled enrichment can coexist; this design imposes no universal offline requirement. [27]

9. LLM investigation with verifiable evidence

The proposed LLM role is evidence navigation and explanation: summarize a timeline, identify contradictory observations, explain an imported finding, suggest a bounded query, or draft a policy proposal for review. Retrieval-augmented generation (RAG) should retrieve typed observations and approved reference material with stable record IDs. It must not treat plausible text as a substitute for a missing record.

Stage	Required output or boundary
Retrieve	Select by case, time, sensor and entity; include coverage gaps and competing candidates. Keep attacker-controlled strings labeled as data.
Explain	Return structured claims with supporting record IDs, alternative explanations and missing evidence. Abstain when attribution is ambiguous.
Verify	A separate validator checks record existence, temporal compatibility and support for each claim. A second LLM opinion alone is not validation.
Review	Show the analyst the original records and source locations beside the summary. Keep the model's proposal distinct from an accepted finding.
Act	Use a separate policy/action service with explicit authorization, scoped operations, audit and rollback. Model output is never executable policy by itself.

Karanjai and colleagues demonstrate passive prompt injection through attacker-controlled network-log fields. Their LogInject evaluation shows that stored telemetry can contaminate a later analyst query. This directly applies to protocol strings, filenames, URLs and payloads entering an assistant context. Formatting data as JSON or asking the model to ignore instructions is not a complete defense. [24]

The proposed controls are architectural: separate trusted instructions from evidence, avoid unrestricted shell tools, constrain retrieval to the case, validate generated arguments, and prevent retrieved content from changing destinations or privileges. Keep all captured values in the evidence store; prepare a typed model input appropriate to the task. If a model needs raw content, preserve its provenance and test the surrounding action boundary against hostile content.

Record model identity, prompt/template version, retrieval set and output. An investigation should remain reproducible when a model is upgraded or unavailable. External enrichment and remote model services remain optional deployment choices with explicit destinations and data handling; their latency or outage must never hold up an OS authorization reply.

> Acceptance requires grounded claims, useful uncertainty and bounded actions. An eloquent explanation that assigns a remote flow to the wrong local process is a failure even when its final threat label happens to be correct.

10. Evaluation before promotion to protection

No new detection effectiveness experiment was performed for this paper. The following is a proposed acceptance program. Thresholds must be chosen before testing for a defined pilot workload; no unmeasured percentage is presented as a product result.

Evaluation layer	Measure and test	Promotion gate
Evidence integrity	Hash and record round trips; repeated fields; capture failure; partial output; schema migration.	Every displayed claim resolves to source records; limits and incomplete sessions are visible.
Correlation	Known local flows with PID/port reuse, clock skew, DHCP, VPN, NAT and multiple candidates.	Measure precision, coverage and abstention separately. No forced identity on deliberately ambiguous fixtures.
Behavior models	Temporal and host/site holdouts; benign administration, builds, updates and validated attack simulations.	Compare against rules and simple models with confidence intervals and identical inputs.
Analyst utility	False leads per endpoint/day, time to inspect, duplicate alerts, useful incident recall and review agreement.	Show an improvement in the intended workflow without hiding alert volume behind aggregate accuracy.
LLM grounding	Unsupported claims, wrong source IDs, missed contradictions, prompt injection and action-boundary attempts.	Report both outcome and evidence quality; no unauthorized action in the defined adversarial test set.
Runtime reliability	Callback deadline misses, sequence gaps, local queue loss, p95/p99 processing latency, memory and energy.	Sustain the pilot workload with documented overload behavior. Model failure leaves local policy available.
Deployment	Entitlements, signing/notarization, permissions, boot/login behavior, upgrades and rollback.	A clean-machine, permissioned live workflow succeeds; source builds alone are insufficient.

Prevent leakage by splitting before feature construction and learning vocabularies only on training data. Keep duplicate incidents, related captures and process lineages in one partition. Evaluate rare attacks with precision/recall and precision-recall curves, including false-alert burden; overall accuracy is misleading under strong class imbalance. Calibrate any probability estimate and measure drift after OS, workload or network changes.

Use reviewed labels that distinguish confirmed incident, benign activity, unknown and insufficient visibility. Analyst feedback is useful but noisy; it should enter a versioned review process rather than silently retrain production protection. New detectors should first run in shadow mode, followed by a bounded pilot and a rollback-ready release.

11. Deliverables and an integration-first roadmap

Milestone	Concrete deliverable	Evidence required to call it complete
1. Capture acceptance	Permissioned live capture, reliable stop, acquisition manifest carried into shared reports; richer drop/partial-state visibility.	Real capture and saved analysis on a clean supported Mac, matching source hashes and an observed GUI workflow.
2. Deterministic correlation	Typed evidence store, process/flow/packet joins, candidate identities, uncertainty UI and case export.	Reproducible fixture and live-flow attribution; ambiguity and loss preserved across app exchange.
3. Network administration	Bonjour/UPnP, IPv6 discovery, device notes/history, scheduled differences and selected SNMP diagnostics.	Validated device identity sources and measured monitoring results across address changes and silent hosts.
4. Engine and fleet adapters	Explicit Nmap job profiles after distribution review; Jamf inventory enrichment; sensor imports with documented schemas.	Tool/version/argument provenance, authenticated scoped connector behavior and repeatable failure handling.
5. Learned prioritization	Versioned feature extraction and simple baseline models, then selected graph/traffic candidates.	Held-out improvement over the baseline with calibration, drift and resource measurements.
6. Grounded assistant	Evidence-cited summaries, query suggestions and reviewable policy proposals.	Claim verification, injection evaluation, analyst utility and auditable action separation.

Milestones describe deliverables and gates, not promised dates. Several can progress independently, but learned models depend on reliable evidence and labels, and trustworthy explanations depend on reliable correlation. Fleet deployment and enterprise operations remain separate product work even if an individual Mac workflow succeeds.

Retain dedicated tools where they are strongest: Nmap for active assessment, Wireshark/TShark for protocol interpretation, Suricata for network detections, and fleet platforms for management and centralized operation. The suite should preserve their attribution and original findings while making cross-tool investigation easier. Integrating an engine does not make its findings a proprietary detector.

> The defensible next generation is an evidence system with useful machine intelligence: fast local policy, durable observations, explainable relationships, measured learned ranking and analyst-controlled decisions. The immediate engineering priority is the correlation foundation that makes every later model answer checkable.

References and supporting records

[1] Orbitus Robotics. Public evidence supplement and frozen implementation record for this edition.

https://www.orbitusrobotics.com/whitepapers/evidence-first-cyber-defense.evidence.json

[2] Apple. Endpoint Security framework overview.

https://developer.apple.com/documentation/endpointsecurity

[3] Apple. es_message_t.deadline: authorization response deadline.

https://developer.apple.com/documentation/endpointsecurity/es_message_t/deadline

[4] Apple. es_message_t.seq_num: event-drop accounting.

https://developer.apple.com/documentation/endpointsecurity/es_message_t/seq_num

[5] Apple. es_new_client: message handler, entitlement and Full Disk Access.

https://developer.apple.com/documentation/endpointsecurity/es_new_client(_:_:)

[6] Apple. Network Extension content filter providers.

https://developer.apple.com/documentation/networkextension/content-filter-providers

[7] Wireshark Foundation. TShark manual.

https://www.wireshark.org/docs/man-pages/tshark.html

[8] Wireshark Foundation. dumpcap manual.

https://www.wireshark.org/docs/man-pages/dumpcap.html

[9] Wireshark Foundation. Installing Wireshark under macOS; ChmodBPF.

https://www.wireshark.org/docs/wsug_html_chunked/ChBuildInstallOSXInstall.html

[10] Jamf. Jamf Pro: Apple device management.

https://www.jamf.com/products/jamf-pro/

[11] Jamf. About Jamf Protect: endpoint and network security.

https://learn.jamf.com/r/en-US/jamf-protect-documentation/About_Jamf_Protect

[12] Nmap Project. Nmap Reference Guide.

https://nmap.org/book/man.html

[13] Aribada Inc. NetAdmin Pro - Network Scanner. Vendor App Store listing.

https://apps.apple.com/us/app/netadmin-pro-network-scanner/id675649574?mt=12

[14] Wireshark Foundation. Following protocol streams.

https://www.wireshark.org/docs/wsug_html_chunked/ChAdvFollowStreamSection.html

[15] North Pole Security. Santa: binary and file access authorization for macOS.

https://github.com/northpolesec/santa

[16] Objective Development. Little Snitch: application firewall and network monitor.

https://www.obdev.at/products/littlesnitch/index.html

[17] OISF. What is Suricata. Development documentation; cited for engine role.

https://docs.suricata.io/en/latest/what-is-suricata.html

[18] Security Onion. Introduction, documentation 2.4.

https://docs.securityonion.net/en/2.4/introduction.html

[19] Wazuh. Platform overview and use cases.

https://wazuh.com/platform/overview/

[20] Tenable. Nessus vulnerability scanner.

https://www.tenable.com/products/nessus

[21] NetSpot. Wi-Fi planning, surveys and coverage features.

https://www.netspotapp.com/features.html

[22] Bilot, T., et al. Sometimes Simpler is Better: A Comprehensive Analysis of State-of-the-Art Provenance-Based Intrusion Detection Systems. USENIX Security 2025.

https://www.usenix.org/conference/usenixsecurity25/presentation/bilot

[23] Bae, C., Ding, H., Ma, S., and Zhang, X. DUPIN: Attack Learning Is Still Needed! Demonstrating Few-Shot after Unsupervised Pretraining Is A Nimble Forensics Learner. USENIX Security 2026.

https://www.usenix.org/conference/usenixsecurity26/presentation/bae

[24] Karanjai, R., et al. Context Contamination in LLM Analysis of Network Security Logs: Poison with Passive Prompt Injection and Mitigation Evaluation. USENIX Security 2026; author manuscript arXiv:2607.14493.

https://arxiv.org/abs/2607.14493

[25] Wang, Y., et al. SecLog-Bench: A Scalable Benchmark for Cybersecurity Log Reasoning with Scaffolded Evidence. arXiv:2610.31000v1, October 22, 2026. Preprint.

https://arxiv.org/abs/2610.31000v1

[26] Zhou, J., et al. Traffic-MoE: A Sparse Foundation Model for Network Traffic Security Analysis. arXiv:2601.00357v2, July 8, 2026. Preprint.

https://arxiv.org/abs/2601.00357v2

[27] Apple. Core ML overview and on-device model integration.

https://developer.apple.com/documentation/coreml/

[28] Nmap Project. Nmap Public Source License. Integration/distribution terms require review.

https://nmap.org/npsl/

Web edition and supplements: https://www.orbitusrobotics.com/whitepapers/evidence-first-cyber-defense/

First edition. Assessed source baseline: SignalAtlas 30a867890f6e and BlackViper 4314a0d82219. Research checked November 7, 2026. Later development is outside this edition.

Prepared with AI-assisted research, source inspection, technical drafting and diagram generation for Orbitus Robotics. This is a self-published technical assessment, not an independently peer-reviewed evaluation. Vendor capabilities are taken from their primary documentation; no head-to-head product trial or new malware-detection benchmark was performed. Repository test results are identified as recorded results, not rerun or production qualification. No private packet payloads, credentials or operational device records are published.

Original conceptual diagrams: Orbitus Robotics. Correspondence: orbitus@orbitusrobotics.com