OFFLOAD SECURITY

Security Scanning & Reporting

Module Whitepaper

Best-of-breed scanners orchestrated into one normalized result — and board-ready reports

Version 1.0 · July 2026 · Classification: Customer-Shareable

1. Overview

The scanning engine runs proven, best-of-breed security tools against every target type — web, API, network, TLS, code, IaC, containers, Kubernetes, and cloud — then normalizes their output into one canonical finding model and generates professional, multi-format reports for executives, engineers, and auditors.

2. Key Capabilities

- Multi-tool orchestration per target: Nuclei and OWASP ZAP for web; Nmap-family and TLS handshake analysis for network/certificates; an OWASP API Top-10 scanner for APIs; OpenGrep, Bandit, and osv-scanner for code; Checkov for infrastructure-as-code; Trivy, Grype, and Syft for containers and SBOMs; Prowler for cloud; and kube-bench, Kubescape, Polaris, and kube-hunter for Kubernetes.
- Isolated execution: each scanner runs in a single-use, resource-capped container with dropped capabilities and no-new-privileges — with a native-binary fallback — so tools are never installed into the platform runtime.
- Cross-tool normalization: findings from every tool are mapped into one canonical shape (severity, evidence, CWE/CVE, OWASP category, remediation, confidence) with severity normalized across tools that grade differently.
- Standards coverage mapping: web and application findings are mapped to OWASP Top 10, OWASP API Top 10, OWASP ASVS, and NIST SSDF, producing a coverage matrix and a security rating.
- Exploit and threat enrichment: findings are enriched with CISA KEV, EPSS, and MITRE ATT&CK context to prioritize what is actually exploitable.
- Professional reporting in multiple formats — PDF, HTML, DOCX, CSV, and JSON — with executive, technical, and compliance/audit variants.
- Reports include an executive summary, per-finding detail with evidence and remediation, charts, framework/compliance mapping, a scan-coverage matrix, and the exact tool versions used — an audit-defensible record.
- Scheduled, recurring scans via a persistent cron/interval scheduler, and on-demand report generation.
- Optional AI-assisted analysis and narrative enrichment, using the customer's own model-provider key when enabled.

3. How It Works

A scan request resolves the target type to its tool set and dispatches each scanner as an isolated container job on worker infrastructure. For a single web or application target, tools run through a phased pipeline (reconnaissance → discovery → testing → post-processing); heavy scanners are paced by a configurable rate profile that defaults to running them one at a time for stability rather than overwhelming the target.

As tools complete, a normalizer collapses their varied output formats into one finding model, normalizes severities, maps to standards, and enriches with KEV/EPSS/MITRE. A report engine then renders the result through a shared professional template — cover page, document-control block, findings, charts, and appendices — into the format the audience needs.

4. Data Handling & Security

Aspect	How it is handled
Credential protection	All connection credentials are encrypted at rest with Fernet authenticated encryption (AES-128-CBC + HMAC), decrypted in memory only at the moment of use, and never returned by APIs.
Tenant isolation	Every record is scoped to the owning team; all reads and writes filter on the tenant identifier, enforced by database indexes.
Retention & deletion	Results are subject to configurable retention windows; data can be hard-deleted per scan and is removed by the tenant-offboarding cascade.
Deployment	Available as SaaS or fully self-hosted / on-premises — in the on-premises model no security data leaves your environment.

5. One Platform, One Risk View

Scan findings flow into the platform's unified vulnerability queue and risk register, feed compliance evidence, and drive the executive dashboards — one scanning and reporting layer shared across cloud, code, container, and Kubernetes security.