Overall Project Checklist vs Proposal (Revision)

Reference: `Security Detection & Threat Intelligence Enhancement Proposal-revise.md`
Updated: March 4, 2026

Legend:

- `[x]` Completed
- `[~]` Partially completed / in progress
- `[ ]` Not started

1) Architecture & Core Platform

- ☑ Detection layer (Wazuh) deployed and integrated
  - Evidence: `wazuh-docker/single-node/docker-compose.yml`, `compose-overrides/wazuh.shared-network.yml`
- ☑ Automation layer (Shuffle) integrated
  - Evidence: `Shuffle/docker-compose.yml`, `scripts/update-shuffle-workflow-from-template.sh`
- ☑ Case management integrated (IRIS-web used in implementation)
  - Evidence: `soc-integrator/app/adapters/iris.py`, `soc-integrator/app/main.py` (`/iris/tickets`)
- ☑ Escalation path (PagerDuty stub in MVP) integrated
  - Evidence: `compose-overrides/pagerduty.stub.yml`, `soc-integrator/app/adapters/pagerduty.py`
- ☑ Orchestration/API layer (`soc-integrator`) operational
  - Evidence: `soc-integrator/app/main.py`, `soc-integrator/app/routes/mvp.py`

Note: Proposal mentions DFIRTrack in architecture section; current implementation uses IRIS-web.

2) Scope of Work (Section 3)

2.1 Create & Tune New Detection Rules / Use Cases

- ☑ Baseline rules/decoders for proposal use cases added

  - Evidence:
    - `wazuh-docker/single-node/config/wazuh_cluster/local_decoder.xml`
    - `wazuh-docker/single-node/config/wazuh_cluster/local_rules.xml`
    - `wazuh-docker/single-node/config/wazuh_cluster/rules/soc-a1-ioc-rules.xml`
    - `wazuh-docker/single-node/config/wazuh_cluster/rules/soc-a2-fortigate-fw-rules.xml`
    - `wazuh-docker/single-node/config/wazuh_cluster/rules/soc-a3-fortigate-vpn-rules.xml`
    - `wazuh-docker/single-node/config/wazuh_cluster/rules/soc-a4-windows-ad-rules.xml`

- [~] Tuning against real production traffic
  - Status: simulator/UAT-oriented tuning done; production false-positive tuning remains

2.2 IOC Detection (DNS / Firewall / IDS-IPS)

☑ IOC enrichment/evaluation APIs implemented

  - Evidence: `soc-integrator/app/main.py` (`/ioc/enrich` , `/ioc/evaluate` , `/ioc/history`)

☑ VirusTotal and AbuseIPDB integrations implemented

  - Evidence: `soc-integrator/app/adapters/virustotal.py` , `soc-integrator/app/adapters/abuseipdb.py`

☑ IOC trace persistence implemented

  - Evidence: `soc-integrator/app/repositories/mvp_repo.py` (`ioc_trace` methods)

- [~] Scheduled IOC feed lifecycle hardening for production
  - Status: core IOC workflow exists; production feed governance/SLAs still to finalize

2.3 VPN Authentication Success from Outside Thailand

☑ MVP VPN evaluate flow implemented

  - Evidence: `soc-integrator/app/routes/mvp.py` (`/mvp/vpn/evaluate`), `soc-integrator/app/services/mvp_service.py`

☑ GeoIP enrichment capability implemented

  - Evidence: `soc-integrator/app/adapters/geoip.py`, `soc-integrator/app/main.py` (`/geoip/{ip}`)

- [~] Production exception list and policy hardening
  - Status: policy framework exists; enterprise exception governance pending

3) End-to-End Workflow & Integration Deliverables (Section 4 / 4.1)

- ☑ Wazuh -> soc-integrator pipeline implemented
  - Evidence: `soc-integrator/app/main.py` (`/wazuh/sync-to-mvp`, `/mvp/incidents/ingest`)
- ☑ soc-integrator -> IRIS ticket/case creation implemented
  - Evidence: `soc-integrator/app/main.py` (`/iris/tickets`)
- ☑ soc-integrator -> Shuffle workflow trigger implemented
  - Evidence: `soc-integrator/app/main.py` (`/shuffle/workflows/{workflow_id}/execute`)
- ☑ soc-integrator -> PagerDuty stub escalation path implemented
  - Evidence: `soc-integrator/app/adapters/pagerduty.py`, `/action/create-incident`

4) Appendix A (Initial Scope Use Cases)

- ☑ A1 DNS/Firewall IOC coverage artifacts in place
- ☑ A2 FortiGate IPS/Firewall coverage artifacts in place
- ☑ A3 FortiGate VPN coverage artifacts in place
- ☑ A4 Windows/AD coverage artifacts in place
  - Evidence: `wazuh-docker/single-node/config/wazuh_cluster/rules/soc-a*-*.xml`

5) Appendix B (Optional Add-On)

- ☑ B1 VMware rule artifact present
  - Evidence: `wazuh-docker/single-node/config/wazuh_cluster/rules/soc-b1-vmware-rules.xml`
- ☑ B2 Log loss monitoring implemented in soc-integrator
  - Evidence: `soc-integrator/app/main.py` (`/monitor/log-loss/check`)
- ☑ B3 Sysmon rule artifact present
  - Evidence: `wazuh-docker/single-node/config/wazuh_cluster/rules/soc-b3-sysmon-rules.xml`

- ☑ Appendix B simulation script added
  - Evidence: `scripts/send-wazuh-proposal-appendix-b-events.sh`

6) Appendix C (Future Enhancements)

- ☑ C1 implemented (impossible travel detection logic)
  - Evidence: `soc-integrator/app/services/c_detection_service.py` (`C1-01`)
- ☑ C2 implemented (C2-01..C2-04)
  - Evidence: `soc-integrator/app/services/c_detection_service.py`
- ☑ C3 implemented (C3-01..C3-04)
  - Evidence: `soc-integrator/app/services/c_detection_service.py`
- ☑ Appendix C simulator added
  - Evidence: `scripts/send-wazuh-proposal-appendix-c-events.sh`
- □ C4 ransomware early warning use cases
- □ C5 endpoint/server anomaly use cases
- □ C6 cloud/SaaS monitoring use cases
- □ C7 SOC maturity monitoring use cases

7) Dashboards, UI, and Operations

- ☑ Wazuh dashboard import automation added
  - Evidence: `scripts/import-wazuh-dashboard.sh`, `scripts/events/*.ndjson`
- ☑ SOC Integrator UI implemented with monitoring and controls
  - Evidence: `soc-integrator/app/ui/index.html`, `soc-integrator/app/ui/assets/app.js`
- ☑ Sim run control in UI (start/stop/logs)
  - Evidence: `/sim/logs/start`, `/sim/logs/stop/{run_id}`, `/sim/logs/stop-running`, `/sim/logs/output/{run_id}`
- ☑ Wazuh Live Correlation view in Systems tab
  - Evidence: `/sim/logs/wazuh-latest/{run_id}` + Systems UI section
- ☑ GeoIP lookup API and UI tab
  - Evidence: `soc-integrator/app/main.py` (`/geoip/{ip}`), GeoIP tab in `/ui`

8) Remaining Work for Production-Ready Acceptance

- [~] Replace/augment lab-only assumptions (PagerDuty stub -> production PagerDuty)
- [~] Production-grade tuning on real logs (A/B/C false positives and thresholds)

- [~] Finalize runbooks, SLA/ownership, and exception governance
- [~] Harden frontend dependency reliability (current UI still references external CDN scripts)

8.1) Latest Incremental Updates (March 4, 2026)

☑ Added production-profile simulator mode for proposal scripts

  - Evidence:
    - `scripts/send-wazuh-proposal-required-events.sh` (`--profile=production`)
    - `scripts/send-wazuh-proposal-appendix-b-events.sh` (`--profile=production`)

☑ Expanded normalization test support in SOC Integrator

  - Evidence:
    - `soc-integrator/app/main.py` (`GET /ingest/wazuh-alert/samples`)
    - `soc-integrator/app/main.py` (`POST /ingest/wazuh-alert` now includes `normalized_event`)

☑ C1 normalization aligned to production log characteristics

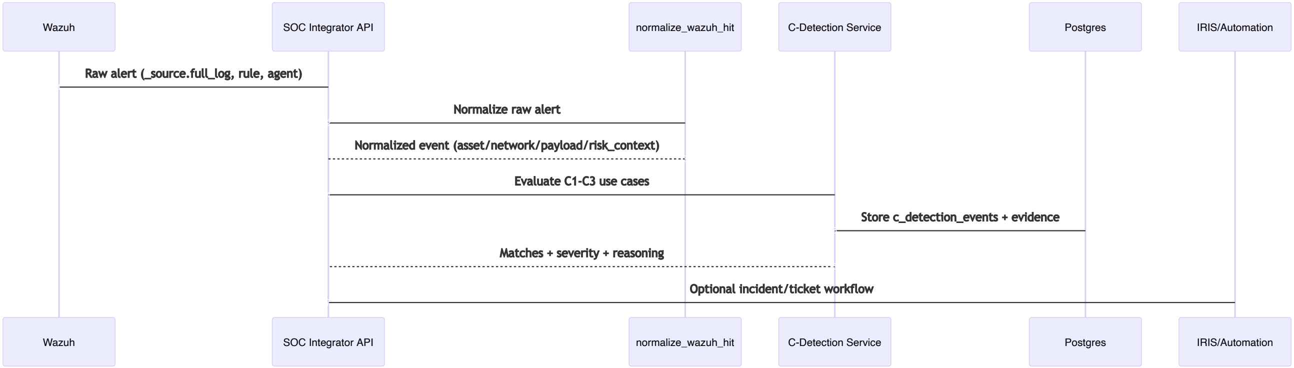
  - Evidence:
    - `soc-integrator/app/services/mvp_service.py` (production-first C1 event typing)

- [~] Production rule validation in Wazuh (`110xxx`) currently constrained by manager runtime instability during lab restarts
  - Status: ingestion works; deterministic decoder/rule verification requires stable manager window.

9) Quick Status Summary

- Completed foundation and integration: **Yes**
- Appendix A initial-scope implementation artifacts: **Present**
- Appendix B optional add-ons: **Largely implemented in lab**
- Appendix C future enhancements: **C1-C3 implemented; C4-C7 pending**
- Primary gap: **production hardening, governance, and operations finalization**

10) Mermaid: SOC Integrator C1-C3 Normalization


Wazuh
SOC Integrator API
normalize_wazuh_hit
C-Detection Service
Postgres
IRIS/Automation
Raw alert (_source.full_log, rule, agent)
Normalize raw alert
Normalized event (asset/network/payload/risk_context)
Evaluate C1-C3 use cases
Store c_detection_events + evidence
Matches + severity + reasoning
Optional incident/ticket workflow
Wazuh
SOC Integrator API
normalize_wazuh_hit
C-Detection Service
Postgres
IRIS/Automation