Project Progress Update

Date: February 13, 2026 Project: FoodProject SOC Platform (Wazuh + Shuffle + IRIS-web + SOC Integrator)

1) Executive Summary

The MVP platform is operational and running end-to-end in the lab environment. Core integrations are in place:

- Detection: Wazuh
- Automation: Shuffle
- Case management: IRIS-web (replacing DFIRTrack)
- Escalation (MVP): PagerDuty Stub
- Orchestration/API layer: soc-integrator

All major containers are currently up, and key health checks are passing.

2) Completed Work

Platform orchestration and operations

- Combined stack runner created and improved (`run-combined-stack.sh`)
- Added command support for:
  - `up` , `down` , `logs` , `status` , `help`
  - per-target control (`wazuh` , `iris` , `shuffle` , `pagerduty` , `integrator`)
- Added consolidated health/status script (`soc-status.sh`)

Integration architecture

- Connected Wazuh, Shuffle, IRIS-web, PagerDuty Stub, and soc-integrator on shared network
- Resolved startup conflicts and runtime issues (port, compose, routing compatibility)

SOC Integrator (MVP)

- Added/validated integration APIs for:
  - Wazuh

    - Shuffle
    - IRIS-web
    - PagerDuty Stub
- Implemented MVP orchestration endpoints:
    - `POST /mvp/incidents/ingest`
    - `POST /mvp/ioc/evaluate`
    - `POST /mvp/vpn/evaluate`
    - `GET /mvp/config/policies`
    - `PUT /mvp/config/policies`
    - `GET /mvp/health/dependencies`
- Added internal API-key protection for mutation endpoints

Persistence layer

- Added PostgreSQL service for soc-integrator (`soc-integrator-db`)
- Added incident/policy/audit schema and startup initialization
- Enabled deduplication and audit tracking for incident processing

Testing utilities and documentation

- Added Wazuh test-event injection script:
    - `scripts/send-wazuh-test-events.sh`
- Added root project docs:
    - `README.md`
- Added root ignore rules:
    - `.gitignore`

3) Current Live Status (Lab)

Current stack status: **UP**

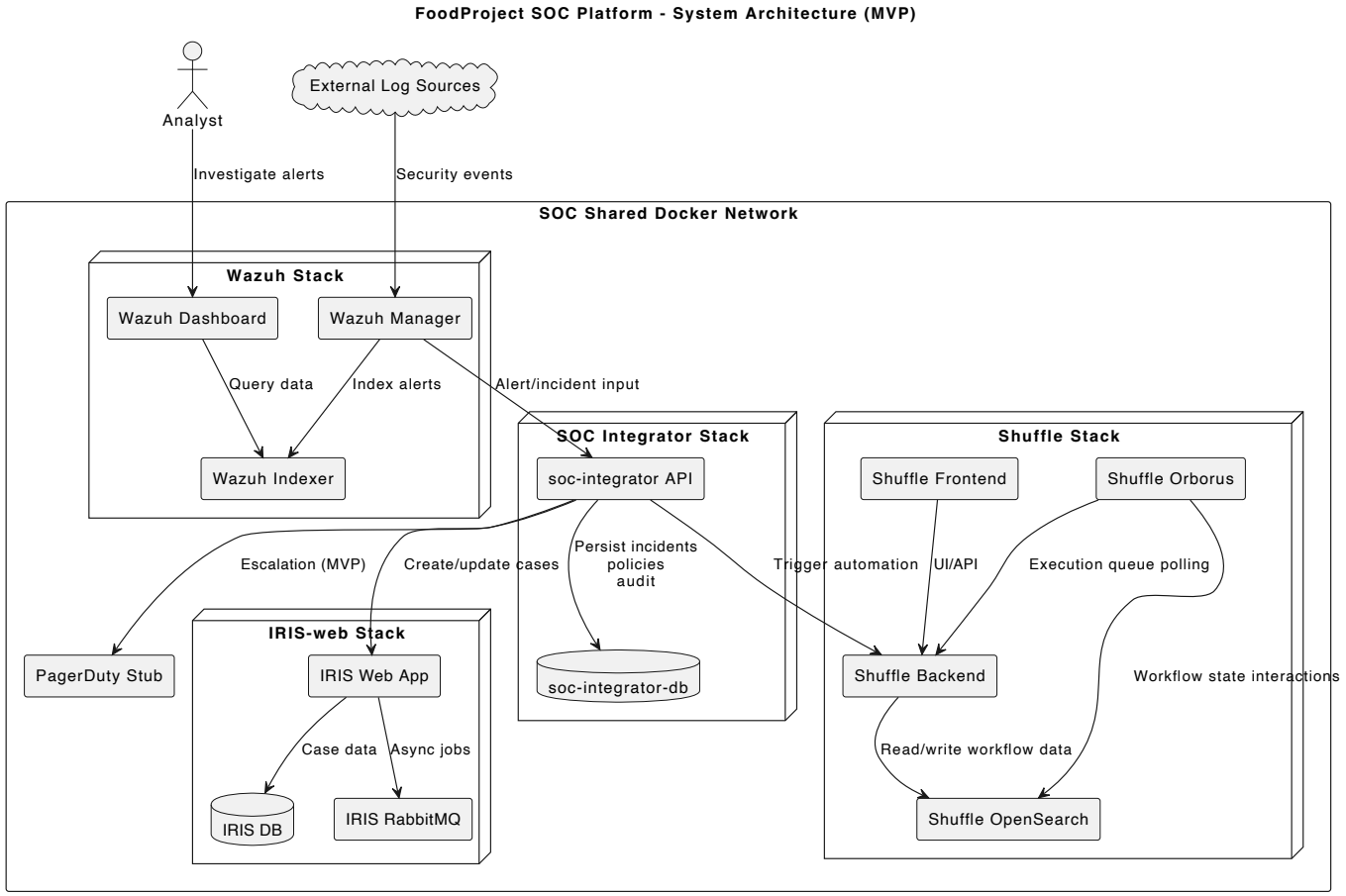
Healthy/available components:

- Wazuh manager, indexer, dashboard
- IRIS-web app/nginx/worker/db/rabbitmq
- Shuffle backend/frontend/opensearch/orborus
- PagerDuty Stub
- soc-integrator + soc-integrator-db

Endpoint checks:

- Wazuh Dashboard: OK
- Wazuh API: OK (auth-protected, expected 401 on unauthenticated root)
- IRIS Web: OK
- Shuffle Frontend: OK
- Shuffle Backend: reachable
- Shuffle OpenSearch: reachable (auth-protected)
- PagerDuty Stub: OK
- soc-integrator `/health`: OK

4) System Architecture Diagram (PlantUML)



5) In Progress / Remaining for Customer UAT

1. Detection content tuning

- Fine-tune Wazuh rules/decoders for customer log patterns and false-positive reduction

2. Use-case calibration

- Validate risk/severity mapping per approved use cases
- Tune exception list and threshold logic (especially VPN geo anomaly)

3. UAT evidence package

- Capture deterministic UAT scenarios and outputs for:
  - IOC flow
  - VPN outside-TH flow
  - IRIS case creation/update
  - PagerDuty Stub escalation path

4. Production hardening items

- Rotate default/local secrets used in lab config
- Lock down internal API keys and access boundaries

6) Risks / Notes

- Current escalation target is **PagerDuty Stub** by design for MVP. Real PagerDuty production integration is the next stage.
- Some Wazuh config certificate directories are root-owned in the local lab clone, which may affect local git add operations if not excluded/fixed.

7) Next Milestone (Proposed)

Next milestone: **MVP UAT Completion**

Target outputs:

- Approved UAT checklist execution
- Tuned policy thresholds for customer environment
- Signed-off incident lifecycle flow: Wazuh event -> soc-integrator decision -> IRIS case -> PagerDuty Stub escalation

---

Date: February 26, 2026 Project: FoodProject SOC Platform (Wazuh + Shuffle + IRIS-web + SOC Integrator)

Incremental Progress Since February 13, 2026

1) IOC Enrichment and Evaluation

- Added IOC APIs in `soc-integrator`:
  - `POST /ioc/enrich`
  - `POST /ioc/evaluate`
  - `GET /ioc/history`
  - `POST /ioc/upload-file`
  - `POST /ioc/evaluate-file`
  - `GET /ioc/analysis/{analysis_id}`
- Integrated VirusTotal adapter for domain/hash/file intelligence and analysis lookups.
- Integrated AbuseIPDB adapter for IP reputation checks.
- Added IOC trace persistence (`ioc_trace`) and repository methods for audit/history.

2) IRIS Integration Enhancements

- Added IRIS ticket APIs in `soc-integrator`:
  - `POST /iris/tickets`
  - `GET /iris/tickets`
- Updated IRIS API key in environment and verified ticket creation path via API.
- Added demo data seeding script:
  - `scripts/seed-iris-demo-data.sh`

3) Shuffle Workflow Automation

- Created and updated sample Shuffle workflow assets for webhook-driven IRIS ticket creation:
  - `shuffle-workflows/sample-webhook-soc-integrator-iris-workflow.json`
  - `shuffle-workflows/sample-webhook-soc-integrator-iris-workflow.md`
- Added workflow update helper script:
  - `scripts/update-shuffle-workflow-from-template.sh`
- Updated target workflow (`07ecad05-ff68-41cb-888d-96d1a8e8db4b`) with:
  - webhook trigger
  - HTTP action (`http 1.4.0`) to call `soc-integrator` ticket API
  - tested webhook execution path to successful completion

4) Networking and Runtime Fixes

- Resolved Shuffle action DNS failure to `soc-integrator` by attaching `soc-integrator` service to Shuffle execution network(s) in:
  - `compose-overrides/soc-integrator.yml`
- Verified connectivity from Shuffle execution context to:
  - `http://soc-integrator:8080/health`

5) Security and Repository Hygiene

- Added `.env` and `.env.*` to root `.gitignore` (kept `.env.example` tracked).
- Removed tracked env files from git cache to prevent secret leakage.
- Updated operational API keys in `soc-integrator/.env` for Shuffle, IRIS, VirusTotal, and AbuseIPDB.

6) Current Status (Lab)

- `soc-integrator` health endpoint: reachable.
- IOC enrich/evaluate flows: operational for domain/hash and file submission paths.
- Shuffle webhook-to-IRIS automation: operational after network fix.
- Core stack components remain available for continued UAT and tuning.

7) Simulation Logs Workstream

Completed

- Added FortiGate simulation coverage for multiple models:
  - 40F
  - 60F
  - 80F
  - 501E
- Added endpoint agent simulation coverage for:
  - Windows clients
  - macOS clients
  - Linux clients
- Added continuous run mode (`--forever`) to simulation scripts for long-running lab traffic generation.
- Extended script set to support realistic event streams for Wazuh ingestion and rule validation.

Operational scripts

- `scripts/send-wazuh-test-events.sh`
- `scripts/send-wazuh-endpoint-agent-test-events.sh`
- additional simulation scripts under `scripts/` for firewall and endpoint scenarios with continuous mode enabled

Detection alignment status

- Simulation work has been aligned to the detection objectives documented in:
  - `Security Detection & Threat Intelligence Enhancement Proposal-2.md`
- Proposal use-case mapping explicitly covered in simulation:
  - **A1. DNS / Firewall (IOC)**:
    - DNS network communication to malicious domain
    - DNS/Firewall malicious domain IOC detection events
  - **A2. FortiGate IPS/IDS & Firewall**:
    - allowed RDP from public IP
    - admin password change
    - create/add admin account
    - disable email notification
    - config download
    - multiple critical/high IDS alerts
    - port scanning (public/private source variants)
    - IOC detection and communication to malicious IP
  - **A3. FortiGate VPN**:
    - authentication success from guest account
    - authentication success from multiple countries
    - brute-force success pattern
    - multiple fail patterns (many accounts from one source)
    - authentication success from outside Thailand
  - **A4. Windows / Active Directory**:
    - privileged/service account authentication failures
    - password spray and multi-source fail patterns
    - success from public IP / guest account
    - pass-the-hash style success indicators
    - account/group privilege change and account lifecycle events (create/re-enable)
    - AD enumeration behavior indicators

- Endpoint client simulations were added to complement proposal scope for heterogeneous environments:
  - Windows agent events
  - macOS agent events
  - Linux agent events
- Current use is suitable for pipeline and workflow validation (ingest -> detect -> automate -> case creation).
- Remaining work is focused on fine-grained scenario calibration:
  - event frequency tuning
  - field/value realism per source
  - expected alert volume by use case for cleaner UAT evidence

8) API Request/Response Samples

IOC Enrich

Request:

```
curl -sS -X POST http://localhost:8088/ioc/enrich \
  -H 'Content-Type: application/json' \
  -d '{
    "ioc_type": "domain",
    "ioc_value": "google.com",
    "sources": ["virustotal"]
  }'
```

Sample response:

```
{
  "success": true,
  "ioc_type": "domain",
  "ioc_value": "google.com",
  "enrichment": {
    "virustotal": {
      "reputation": 120,
      "last_analysis_stats": {
        "malicious": 0,
        "suspicious": 0,
        "harmless": 90
      }
    }
  }
}
```


IOC Evaluate

Request:

```
curl -sS -X POST http://localhost:8088/ioc/evaluate \
  -H 'Content-Type: application/json' \
  -d '{
    "ioc_type": "hash",
    "ioc_value": "44d88612fea8a8f36de82e1278abb02f",
    "sources": ["virustotal"]
  }'
```

Sample response:


```
{
  "success": true,
  "matched": true,
  "severity": "high",
  "reason": "VirusTotal marked IOC as malicious",
  "ioc_type": "hash",
  "ioc_value": "44d88612fea8a8f36de82e1278abb02f"
}
```


Create IRIS Ticket (via soc-integrator)

Request:

```
curl -sS -X POST http://localhost:8088/iris/tickets \
  -H 'Content-Type: application/json' \
  -d '{
    "title": "Suspicious domain detected",
    "description": "Automated ticket from IOC evaluation pipeline",
    "severity": "medium",
    "source_ref": "shuffle-webhook-demo"
  }'
```

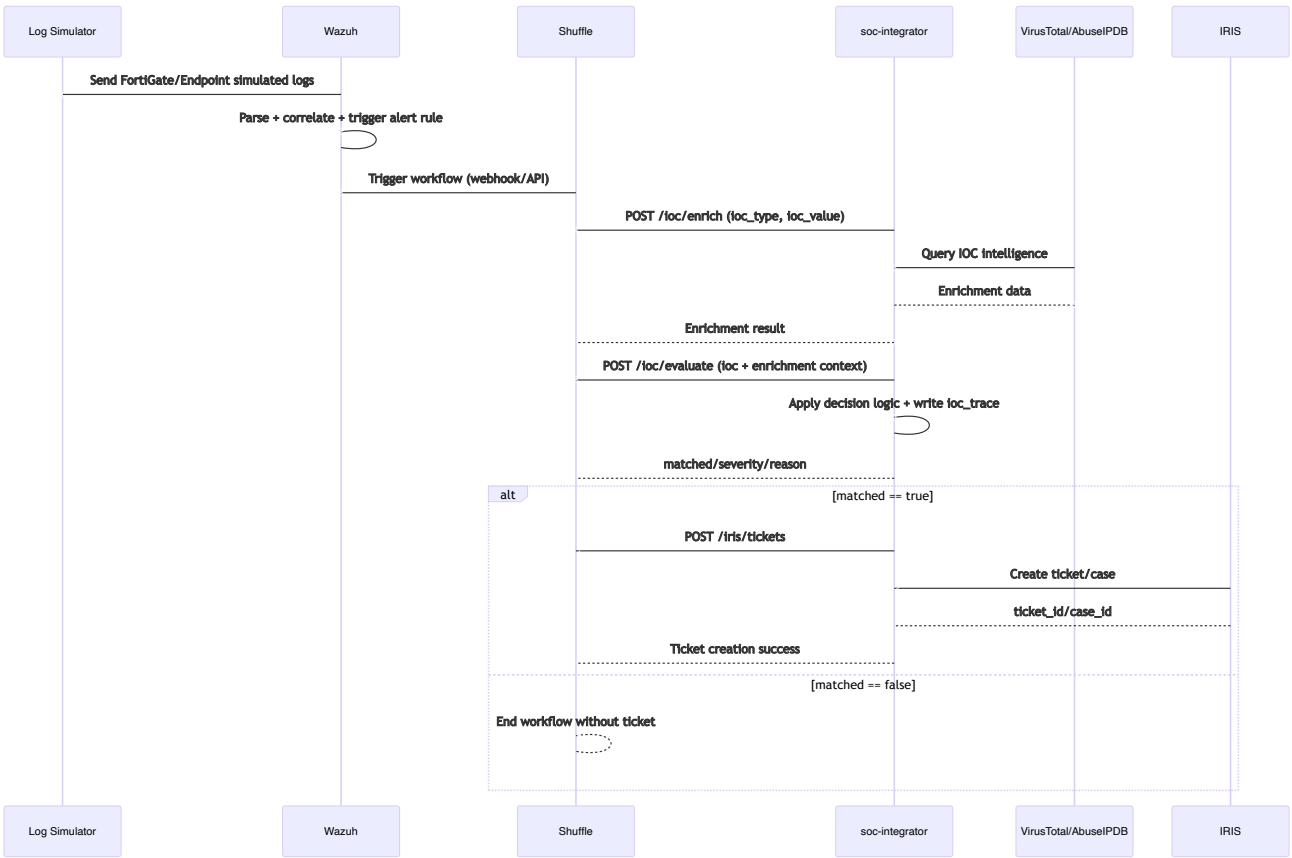
Sample response:

```
{
  "success": true,
  "ticket_id": 53,
  "case_id": 53,
  "status": "open"
}
```

9) Why IOC Was Added to SOC Integrator

- To centralize threat-intelligence logic in one API layer instead of duplicating enrichment/evaluation rules across Shuffle workflows and other services.
- To provide a consistent decision contract (`enrich` for context, `evaluate` for action/verdict) that downstream automation can trust.
- To improve traceability by storing IOC checks and decisions in `soc-integrator` history for audit, tuning, and UAT evidence.
- To simplify integrations with multiple intelligence providers (VirusTotal, AbuseIPDB, and future sources) behind one internal interface.
- To reduce workflow complexity in Shuffle so playbooks focus on orchestration (branching, ticketing, notifications) while IOC decisioning stays in backend logic.

10) Sequence Diagram (MermaidJS)


Log Simulator
Wazuh
Shuffle
soc-integrator
VirusTotal/AbuseIPDB
IRIS
Send FortiGate/Endpoint simulated logs
Parse + correlate + trigger alert rule
Trigger workflow (webhook/API)
POST /ioc/enrich (ioc_type, ioc_value)
Query IOC intelligence
Enrichment data
Enrichment result
POST /ioc/evaluate (ioc + enrichment context)
Apply decision logic + write ioc_trace
matched/severity/reason
alt
[matched == true]
POST /iris/tickets
Create ticket/case
ticket_id/case_id
Ticket creation success
[matched == false]
End workflow without ticket
Log Simulator
Wazuh
Shuffle
soc-integrator
VirusTotal/AbuseIPDB
IRIS

11) SOC Integrator API Inventory

Group	Method	Endpoint	Notes
Core	GET	`/health`	Service health and target configuration
Core	POST	`/ingest/wazuh-alert`	Normalize inbound Wazuh alert payload
Core	POST	`/action/create-incident`	Create PagerDuty incident
Core	POST	`/action/trigger-shuffle`	Trigger Shuffle workflow execution
Core	POST	`/action/create-iris-case`	Create IRIS case (legacy action endpoint)
IRIS	POST	`/iris/tickets`	Create IRIS ticket/case via soc-integrator
IRIS	GET	`/iris/tickets`	List/query IRIS tickets/cases
IOC	POST	`/ioc/enrich`	IOC enrichment from configured intel sources
IOC	POST	`/ioc/evaluate`	IOC decisioning/verdict
IOC	POST	`/ioc/upload-file`	Upload file to IOC backend (VirusTotal flow)
IOC	GET	`/ioc/analysis/{analysis_id}`	Retrieve IOC analysis status/result

Group	Method	Endpoint	Notes
IOC	POST	`/ioc/evaluate-file`	Evaluate file indicator or uploaded sample
IOC	GET	`/ioc/history`	Retrieve stored IOC trace history
Shuffle	GET	`/shuffle/health`	Shuffle service reachability check
Shuffle	GET	`/shuffle/auth-test`	Validate Shuffle API key access
Shuffle	POST	`/shuffle/login`	Login against Shuffle API
Shuffle	POST	`/shuffle/generate-apikey`	Generate Shuffle API key from credentials
Shuffle	GET	`/shuffle/workflows`	List workflows
Shuffle	GET	`/shuffle/workflows/{workflow_id}`	Get workflow detail
Shuffle	POST	`/shuffle/workflows/{workflow_id}/execute`	Execute specific workflow
Shuffle	GET	`/shuffle/apps`	List installed/available Shuffle apps
Shuffle	POST	`/shuffle/proxy`	Generic proxy request to Shuffle API
Wazuh	GET	`/sync/wazuh-version`	Fetch Wazuh version information
Wazuh	GET	`/wazuh/auth-test`	Validate Wazuh API authentication

Group	Method	Endpoint	Notes
Wazuh	GET	`/wazuh/manager-info`	Manager information
Wazuh	GET	`/wazuh/agents`	List Wazuh agents
Wazuh	GET	`/wazuh/alerts`	Query recent Wazuh alerts
Wazuh	GET	`/wazuh/manager-logs`	Read manager logs
Wazuh	POST	`/wazuh/sync-to-mvp`	Sync Wazuh alerts into MVP pipeline
Wazuh	GET	`/wazuh/auto-sync/status`	Auto-sync loop status
MVP	POST	`/mvp/incidents/ingest`	Ingest incident into MVP flow
MVP	POST	`/mvp/ioc/evaluate`	Evaluate IOC under MVP policy
MVP	POST	`/mvp/vpn/evaluate`	Evaluate VPN event under MVP policy
MVP	GET	`/mvp/config/policies`	Read MVP policy configuration
MVP	PUT	`/mvp/config/policies`	Update MVP policy configuration
MVP	GET	`/mvp/health/dependencies`	Dependency health snapshot

Additional FastAPI-generated endpoints:

- `GET /docs`
- `GET /openapi.json`