Mini SOC – Wazuh on Docker Swarm

Export: 2025-08-28 20:13 UTC

Executive Summary

This submission delivers a reproducible **Mini SOC** on **Docker Swarm** with CI/CD gates (build → Trivy → tests → deploy) and a custom **Wazuh** detection rule for suspicious SSH activity.

Part One – Mini SOC
- Wazuh stack: Indexer, Manager, Dashboard with HTTPS via Traefik.
- Persistence on Swarm volumes, parameterized with Ansible.
- CI/CD: builds an example image, scans with Trivy (fail on Critical/High), runs Selenium + API probes, and deploys via Ansible on `main`.
- Secrets: GH Secrets → Swarm secrets; TLS automated with ACME.

Part Two – Threat Detection
- Decoder extracts `status`, `user`, `srcip` from sshd logs.
- Rules correlate ≥3 failures within 60s from same IP followed by a success.
- "New user" is simulated with a baseline list file (documented extension paths include CDB or FTS state).

Evidence (to include when you run it)
- CI pipeline screenshots and Trivy artifact in `evidence/`.
- Wazuh dashboard screenshot over HTTPS.
- ossec-logtest output demonstrating the rule firing.

Technical Walkthrough

1. **Swarm bootstrap**: Ansible role `swarm-init` initializes the cluster if needed.
2. **Networks & Secrets**: `networks` role creates overlay nets; `secrets` role creates Swarm secrets.
3. **Stack deploy**: `stack` role renders `wazuh-stack.yml.j2` and deploys via `docker_stack`.
4. **Tests**: Selenium verifies HTTPS + login form; API probe checks endpoint reachability.
5. **Rule Testing**: `tests/ssh_rule/generate_events.sh` pipes sample logs to `ossec-logtest`.

Assumptions
- DNS for the dashboard FQDN resolves to the Swarm ingress node(s).
- Self-hosted runner has Docker, Ansible, Trivy, Chrome/Chromedriver pre-installed.