Blandskron

Laboratorio de Seguridad para Arquitectura de Microservicios

1. Objetivo General

Diseñar e implementar un entorno de laboratorio en Docker que simule una arquitectura de microservicios moderna, segura y monitoreada, con capacidad para detectar y analizar amenazas, registrar eventos, y aplicar políticas de seguridad. El entorno debe permitir observación en tiempo real, escaneo de vulnerabilidades y validación de flujos de tráfico entre servicios.

2. Componentes del Laboratorio

2.1. Microservicios

- `auth-service`: Servicio de autenticación (JWT)
- `user-service`: Gestión de usuarios
- `product-service`: Servicio de productos

Todos expuestos vía REST con FastAPI o Node.js.

2.2. Gateway/API Management

- `Kong Gateway`: Enrutamiento de tráfico, autenticación, control de acceso y rate limiting

2.3. Monitoreo y Observabilidad

- `Elastic Stack (ELK)`:
    - Elasticsearch: almacenamiento centralizado de logs
    - Logstash: ingestión y transformación de logs
    - Kibana: visualización y dashboards
- `Filebeat`: recolector de logs desde contenedores

2.4. Seguridad y detección de amenazas

- `Falco`: detección de comportamiento anómalo (Syscalls, Docker, red)
- `Trivy`: escaneo de vulnerabilidades de imágenes en tiempo real

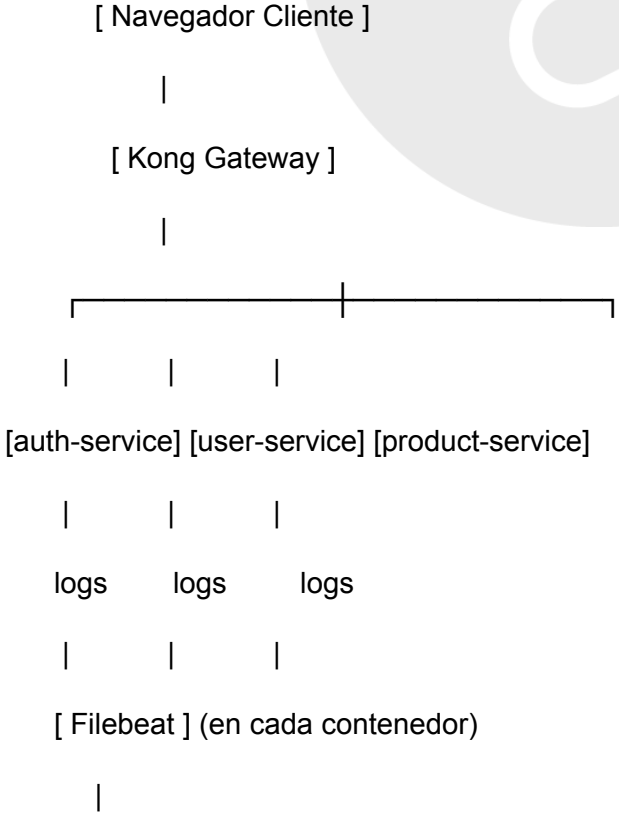
2.5. Análisis de tráfico

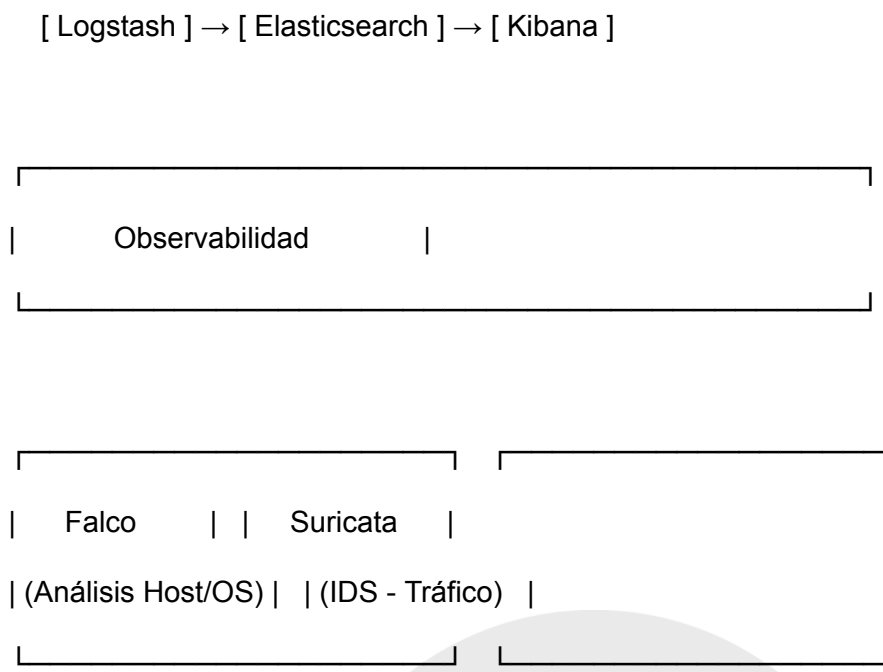
- `Suricata`: detección de amenazas en red y alertas
    - Capta tráfico entre contenedores usando red `host` o interfaz bridge

2.6. Políticas de autorización

- `Open Policy Agent (OPA)`:
    - Define y evalúa políticas RBAC por endpoint

3. Arquitectura del Entorno





4. Requisitos Previos

Software

- Docker ≥ 20.10
- Docker Compose ≥ 1.29
- SO: Linux recomendado (por permisos y compatibilidad de red)
- RAM: mínimo 6 GB libres (ideal 8–12 GB)

5. Seguridad en el Entorno

5.1. Segmentación por red

- Microservicios se despliegan en `internal_network`

- ELK en `monitoring_network`
- OPA y Kong en `gateway_network`
- Suricata usa modo `host` para inspección completa

5.2. Logs y métricas

- Todos los servicios envían logs JSON a `Filebeat`
- Logs enriquecidos con campos de origen, ID de usuario y JWT claims
- Dashboards personalizados en Kibana para:
    - Accesos fallidos
    - Actividad sospechosa
    - Eventos de Falco
    - Alertas de Suricata

5.3. Control de acceso

- Kong valida tokens JWT emitidos por `auth-service`
- OPA autoriza rutas según roles (`admin`, `user`, `auditor`)
- Plugins activos en Kong:
    - `rate-limiting`
    - `jwt`
    - `acl`
    - `cors`

6. Simulaciones permitidas

- Ataques a endpoints (`curl`, SQLi, fuerza bruta)
- Tráfico anómalo (`nmap`, `hping3`)
- Cambios en archivos del contenedor (detectados por Falco)
- Escaneo de imágenes con vulnerabilidades conocidas

7. Flujo de trabajo en el laboratorio

1. El usuario inicia sesión en `auth-service` y obtiene JWT
2. Envía peticiones a `user-service` o `product-service` vía Kong
3. Kong valida el JWT, aplica rate limiting y delega autorización a OPA
4. Los microservicios escriben logs a `stdout` (recogidos por Filebeat)
5. Suricata y Falco generan eventos si detectan comportamientos sospechosos
6. Kibana centraliza todos los eventos para su análisis

8. Extensiones opcionales

- Integrar `Wazuh` como alternativa SIEM avanzada
- Añadir `Jaeger` o `OpenTelemetry` para trazabilidad
- Incluir `Prometheus + Grafana` para métricas técnicas (CPU, RAM, red)
- Aplicar `GitHub Actions` para automatizar escaneo con Trivy

9. Archivos a desarrollar

Puedo ayudarte a construir estos archivos si lo deseas:

- `docker-compose.yml` completo
- Carpetas `/services/auth`, `/services/user`, etc.
- Configuraciones:
    - `filebeat.yml`
    - `falco_rules.yaml`
    - `suricata.yaml`
    - `opa/policies.rego`
- Dashboards para Kibana exportables
- Scripts para simular ataques

10. Conclusión

Este laboratorio replica un entorno empresarial real con enfoque en seguridad, monitoreo y microservicios. Permite al usuario:

- Comprender los flujos de ciberseguridad moderna
- Correlacionar tráfico, eventos de host, y logs de aplicación
- Desarrollar buenas prácticas de DevSecOps