Análisis de Herramientas

- Trivy (by Aqua Security)
- Grype + Syft (by Anchore)
- SonarQube
- OWASP ZAP
- Semgrep
- Checkov
- Snyk
- Falco
- Faraday

Trivy (by Aqua Security)

Descripción

Trivy es un escáner de seguridad todo en uno para contenedores, código, repositorios Git y más. Identifica vulnerabilidades (CVE), secretos expuestos y configuraciones erróneas.

Uso local

```
# Escanear una imagen Docker
trivy image nginx:latest

# Escanear archivos o repositorios
trivy fs .

# Escanear un repositorio remoto
trivy repo https://github.com/usuario/proyecto
```

Reportes

- Formatos: JSON, SARIF (para GitHub), tabla, plantillas personalizadas.
- Exportables a sistemas como DefectDojo.

Integración en CI/CD

- Ejecuta Trivy como acción de GitHub para escanear la imagen del contenedor Docker en busca de vulnerabilidades. Puede detener el pipeline si encuentra vulnerabilidades críticas.

```
      - name: Generate Trivy Vulnerability Report
        uses: aquasecurity/trivy-action@master
        with:
          scan-type: "fs"
          output: trivy-report.json
          format: json
          scan-ref: .
          exit-code: 0
```

Integraciones

- GitHub.
- Slack (via Webhook).
- Jira: se puede automatizar la creación de tickets con scripts + API.
- Plane, Microsoft Teams: vía Webhooks o bots.
- VSCode: extensión oficial para escaneo local.
- DevSecOps: se puede integrar con DefectDojo, Kubernetes Admission Controllers.

UI

- Tiene una **UI web experimental** en el proyecto Trivy Dashboard.
- CLI amigable.

Licencia

- **Open source (Apache 2.0)**.
- Puede instalarse localmente o en máquinas virtuales gratis.

Facilidad de uso

- Muy fácil. Documentación clara y CLI muy intuitiva.

Cumplimiento con estándares

- **OWASP:** Detecta vulnerabilidades en dependencias alineadas al OWASP Top 10, especialmente el punto A06:2021 sobre componentes vulnerables.
- **NIST:** Compatible con NIST SP 800-53 (controles RA-5, SI-2) al detectar vulnerabilidades técnicas de forma automatizada.
- **ISO/IEC 27001:** Apoya el cumplimiento del control A.12.6.1 (gestión de vulnerabilidades técnicas).
- **CIS Benchmarks:** Escanea configuraciones de contenedores y Kubernetes contra benchmarks de seguridad del CIS.

Grype + Syft (by Anchore)

Descripción

- **Syft**: genera un SBOM (Software Bill of Materials).
- **Grype**: analiza el SBOM o imagen para encontrar CVEs.

Uso local

```
# Syft - generar SBOM
syft nginx:latest -o json > sbom.json

# Grype - escanear imagen
grype nginx:latest

# Escanear SBOM generado
grype sbom:sbom.json
```

Reportes

- JSON, table, CycloneDX, SPDX, SARIF.
- Puede exportarse a sistemas como Sonatype, DefectDojo.

Integración en CI/CD

- Una acción de GitHub para invocar el escáner Grype y devolver las vulnerabilidades encontradas, y opcionalmente fallar si se encuentra una vulnerabilidad con un nivel de gravedad configurable.

```
- name: Scan current project
  uses: anchore/scan-action@v6
  with:
    path: "."
```

Integraciones

- GitHub (SARIF reports en Security tab).
- Slack, Teams, Jira (mediante integraciones manuales/API).
- VSCode: No oficial, pero puede usarse vía terminal.

UI

- No posee UI propia.
- Compatible con Anchore Enterprise UI (versión paga).

Licencia

- **Open source (Apache 2.0)**.
- Funciona perfectamente en VMs locales.

Facilidad de uso

- Fácil. CLI clara, requiere instalación de dos binarios (`syft`, `grype`).

Cumplimiento con estándares:

- **OWASP:** Ayuda a mitigar riesgos del OWASP Top 10 relacionados con componentes desactualizados o vulnerables.
- **NIST:** Compatible con NIST SP 800-218 (SSDF) por generar y analizar SBOMs; se alinea con controles RA-5, SI-2 de NIST 800-53.
- **ISO/IEC 27001:** Contribuye a controles como A.8.1.1 (inventario de activos) y A.12.6.1 (vulnerabilidades técnicas).
- **CIS Benchmarks:** Aplica indirectamente cuando se usa en conjunto con políticas de seguridad de configuración.

SonarQube

Descripción

SonarQube es una herramienta de análisis estático de código para detectar bugs, vulnerabilidades, y code smells en más de 25 lenguajes.

Uso local

```
# Análisis local con CLI
sonar-scanner -Dsonar.projectKey=myproject -Dsonar.sources=. -
Dsonar.host.url=http://localhost:9000
```

Reportes

- Interfaz web con dashboards.
- Muestra cobertura de código, vulnerabilidades, duplicaciones, etc.

Integración en CI/CD

- Plugins para GitHub Actions.
- DevSecOps: integración con calidad de código.

```
    - name: SonarQube Scan
      uses: SonarSource/sonarqube-scan-action@<action version> # Ex: v4.1.0, See the latest
version at https://github.com/marketplace/actions/official-sonarqube-scan
      env:
        SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
        SONAR_HOST_URL: ${{ vars.SONAR_HOST_URL }}
```

Integraciones

- GitHub.
- Jira: integraciones para crear tickets automáticamente.
- Slack, Microsoft Teams: plugins para notificaciones.
- VSCode: extensión oficial.

UI

- Sí, una de las mejores UI del sector.

Licencia

- Versión **Community (gratis, OSS)**.
- Versiones Enterprise y Developer con más reglas/lenguajes.
- Puede instalarse en VMs (Docker o instalación manual).

Facilidad de uso

- Moderado. Requiere configuración inicial y escáner. UI muy intuitiva.

Cumplimiento con estándares:

- **OWASP:** Incluye reglas específicas alineadas con OWASP Top 10 (inyecciones, XSS, autenticación insegura, etc.).
- **NIST:** Compatible con NIST CSF y NIST 800-53 al apoyar revisiones de código (controles SA-11, RA-5).
- **ISO/IEC 27001:** Cumple con controles como A.14.2.5 (principios de desarrollo seguro) y A.14.2.8 (pruebas técnicas).
- **PCI DSS:** Cumple con requisitos de análisis de código seguro como el 6.3.2.

OWASP ZAP

Descripción

OWASP ZAP es una herramienta de escaneo de seguridad para aplicaciones web, mantenida por OWASP. Detecta vulnerabilidades como XSS, SQLi, etc.

Uso local

```
# Interfaz gráfica
zap.sh

# Escaneo desde CLI
zap-baseline.py -t http://localhost:8080 -r reporte.html
```

Reportes

- HTML, XML, Markdown.
- Exportables a JIRA, DefectDojo, etc.

Integración en CI/CD

- Una acción de GitHub para ejecutar el ZAP Baseline scan para encontrar vulnerabilidades en su aplicación web.

```
      - name: ZAP Scan
        uses: zaproxy/action-baseline@v0.14.0
        with:
          token: ${{ secrets.GITHUB_TOKEN }}
          docker_name: 'ghcr.io/zaproxy/zaproxy:stable'
          target: 'https://www.zaproxy.org'
          rules_file_name: '.zap/rules.tsv'
          cmd_options: '-a'
```

Integraciones

- GitHub, Jira (plugin).
- Slack, Teams: notificaciones mediante scripting.
- VSCode: sin integración directa, pero puede abrirse con CLI.

UI

- GUI muy completa (modo GUI, CLI y daemon).

Licencia

- **Open source (Apache 2.0)**.
- Disponible para máquinas virtuales, también en Docker.

Facilidad de uso

- Moderado. Ideal para usuarios con conocimiento en pruebas de seguridad web.

Cumplimiento con estándares:

- **OWASP:** Está alineado directamente con el OWASP Top 10, siendo una herramienta oficial de la fundación.
- **NIST:** Se alinea con NIST 800-53 (RA-5: escaneo de vulnerabilidades, CA-7: monitoreo continuo).
- **ISO/IEC 27001:** Permite cumplir con controles como A.12.6.1 (gestión de vulnerabilidades) y A.14.2.8 (pruebas técnicas).
- **PCI DSS:** Soporta los requerimientos de pruebas de seguridad continuas (requisito 11.2).

Semgrep

Descripción

Semgrep es un escáner de análisis estático que detecta vulnerabilidades, errores de seguridad, problemas de estilo, etc., mediante reglas personalizables.

Uso local

```
# Escaneo básico
semgrep scan --config=p/ci .

# Escaneo con reglas personalizadas
semgrep scan --config=rules/mi_regla.yml .
```

Reportes

- JSON, JUnit, SARIF.
- Integración con GitHub Security tab.

Integración en CI/CD

- Semgrep Action ejecuta Semgrep en entornos de CI. También puede conectarse a la aplicación Semgrep para configurar reglas y revisar hallazgos en una interfaz web.

```
- uses: returntocorp/semgrep-action@v1
```

Integraciones

- GitHub, Jira (con integración pagada o scripts).
- Slack, Teams, Plane.
- VSCode: extensión oficial para ejecutar reglas locales.

UI

- Dashboard web gratuito (requiere cuenta).

Licencia

- Versión gratuita OSS.
- Versión empresarial con dashboard avanzado.

Facilidad de uso

- Fácil. Reglas YAML simples de entender.

Cumplimiento con estándares:

- **OWASP:** Las reglas pueden alinearse al OWASP Top 10, detectando problemas de seguridad en el código fuente.
- **NIST:** Compatible con el marco NIST CSF y NIST 800-53, específicamente en prácticas de desarrollo seguro (SA-11).
- **ISO/IEC 27001:** Apoya controles como A.14.2.5 (control del desarrollo de software) y A.14.2.8 (pruebas técnicas).
- **SOC 2 / PCI DSS:** Puede integrarse en pipelines para cumplimiento de requisitos de seguridad de código seguro.

Checkov

Descripción

Checkov es un escáner de infraestructura como código (IaC) que detecta configuraciones inseguras en Terraform, CloudFormation, Kubernetes, etc.

Uso local

```
checkov -d .    # Escanea el directorio actual
```

Reportes

- CLI, JSON, JUnit, SARIF.
- Puede integrarse con plataformas como Prisma Cloud o DefectDojo.

Integración en CI/CD

- Esta acción de GitHub ejecuta Checkov en infraestructura como código, paquetes de código abierto, imágenes de contenedores y configuraciones de CI/CD para identificar configuraciones incorrectas, vulnerabilidades y problemas de cumplimiento de licencias.

```
      - name: Run Checkov action
        id: checkov
        uses: bridgecrewio/checkov-action@master
        with:
          directory: .
          soft_fail: true
          download_external_modules: true
          github_pat: ${{ secrets.GH_PAT }}
        env:
          GITHUB_OVERRIDE_URL: true  # optional: this can be used to instruct the action to
override the global GIT config to inject the PAT to the URL
```

Integraciones

- GitHub, Jira (API), Slack.

- VSCode: extensión oficial para resaltar problemas en tiempo real.

UI

- Checkov OSS no tiene UI propia, pero Prisma Cloud (versión paga) sí.

Licencia

- **Open source (Apache 2.0)**.
- Instalación simple en máquinas virtuales.

Facilidad de uso

- Muy fácil. CLI sencilla y reglas predefinidas muy completas.

Cumplimiento con estándares:

- **OWASP:** Apoya indirectamente la mitigación de riesgos del OWASP Top 10 para aplicaciones cloud-native.
- **NIST:** Compatible con controles de NIST 800-53 (por ejemplo, CM-6: configuración, SC-12: seguridad criptográfica).
- **ISO/IEC 27001:** Apoya la seguridad en configuración de infraestructura (controles A.12.1.2 y A.14.1.3).
- **CIS Benchmarks:** Checkov valida configuraciones directamente contra benchmarks CIS para AWS, Azure, GCP, Kubernetes.

Snyk

Descripción

Snyk es una herramienta para detectar vulnerabilidades en dependencias, contenedores, código IaC y código fuente.

Uso local

```
snyk test     # Escaneo de dependencias
snyk code test  # Escaneo de código fuente
```

Reportes

- JSON, CLI, UI Web.
- Excelente presentación en su portal online.

Integración en CI/CD

- Un conjunto de acciones de GitHub para usar Snyk y buscar vulnerabilidades en tus proyectos de GitHub. Se requiere una acción diferente según el lenguaje o la herramienta de compilación que uses.

```
      - name: Run Snyk to check for vulnerabilities
        uses: snyk/actions/node@master
        env:
          SNYK_TOKEN: ${{ secrets.SNYK_TOKEN }}
        with:
          command: monitor
```

Integraciones

- GitHub (Security Tab), Jira, Slack, Microsoft Teams.
- VSCode: extensión oficial con análisis en tiempo real.

UI

- UI Web muy amigable.

Licencia

- Versión gratuita limitada (para proyectos públicos o uso individual).
- Planes comerciales.
- Instalación en máquinas virtuales posible.

Facilidad de uso

- Muy fácil. Requiere cuenta en Snyk.

Cumplimiento con estándares:

- **OWASP:** Compatible con el OWASP Top 10 a través de escaneo de código y dependencias vulnerables.
- **NIST:** Alineado con NIST SSDF (Secure Software Development Framework) y NIST 800-53 (RA-5, SI-2).
- **ISO/IEC 27001:** Aporta cumplimiento en controles de gestión de vulnerabilidades (A.12.6.1) y desarrollo seguro (A.14.2.5).
- **CIS Benchmarks:** Soporta escaneo de infraestructura como código conforme a CIS Benchmarks.
- **SOC 2 / GDPR / HIPAA / PCI DSS:** Ayuda en auditorías de cumplimiento mediante remediación continua y visibilidad.

Falco

Descripción

Falco es un sistema de detección de intrusos para Kubernetes y contenedores. Monitorea el comportamiento en tiempo real basado en reglas.

Uso local

```
falco        # Inicia la monitorización
```

Reportes

- Logs, JSON, alertas.
- Puede enviar a Syslog, Slack, Webhooks, Prometheus.

Integración en CI/CD

- No escanea código, pero puede monitorear pods en tiempo real durante tests.

Integraciones

- Prometheus, Grafana, Slack, Microsoft Teams, Elasticsearch, Jira.
- VSCode: no aplicable.

UI

- No posee UI, se integra con dashboards como Grafana.

Licencia

- **Open source (Apache 2.0)**.
- Instalación sencilla en VMs o clústeres.

Facilidad de uso

- Avanzado. Requiere conocimientos de reglas y eventos del sistema.

Cumplimiento con estándares:

- **OWASP:** No aplica directamente, pero puede apoyar seguridad operacional en ambientes donde se despliegan apps OWASP.
- **NIST:** Compatible con NIST 800-53 en controles como SI-4 (detección de incidentes), AU-6 (auditoría y monitoreo).
- **ISO/IEC 27001:** Apoya el cumplimiento de controles como A.12.4 (registro de eventos) y A.16.1 (gestión de incidentes).
- **CIS Benchmarks:** Se utiliza para detectar desviaciones en tiempo real de configuraciones definidas por CIS.

Faraday

Descripción

Faraday es una plataforma de gestión de vulnerabilidades que permite integrar resultados de herramientas de análisis de seguridad y centralizar reportes orientada a equipos de seguridad, pentesters y procesos de DevSecOps.

Permite:

- Consolidar hallazgos de múltiples herramientas de análisis.
- Gestionar y priorizar vulnerabilidades de forma centralizada.
- Colaborar en tiempo real en auditorías y pruebas de seguridad.

Funciona como un **servidor central** con una interfaz web y un cliente CLI (`faraday-cli`) que facilita la autenticación, gestión de workspaces y la carga automática de reportes a la plataforma, lo que permite integrarlo en pipelines de CI/CD. Soporta múltiples formatos de reportes y herramientas de seguridad, como Trivy, OpenVAS, Nessus, Burp Suite, etc.

Uso local

Faraday se puede ejecutar localmente mediante Docker o instalación directa en Linux.
Ejemplo con Docker:

```
docker run -it -p 5985:5985 -v faraday_data:/home/faraday/.faraday faradaysec/faraday
```

CLI básico

Iniciar sesión:

```
faraday-cli auth login --server https://<host>:5985 --username faraday --password <pass>
```

Subir un reporte:

```
faraday-cli tool report ./trivy-report.json --workspace <workspace>
```

Listar workspaces:

```
faraday-cli workspace list
```

Reportes

Faraday procesa los reportes de herramientas de seguridad y los almacena en un **workspace**.

- **Ubicación de datos**: Principalmente en la sección **Assets**, donde se relacionan vulnerabilidades con hosts, servicios o aplicaciones.
- **Dashboard**: No siempre muestra todo automáticamente, ya que prioriza métricas y gráficos generales.
- **Formatos soportados**: XML, JSON, CSV y formatos nativos de herramientas como Nessus, OpenVAS, Trivy, Burp, Nmap, etc.

Integración en CI/CD

Faraday puede integrarse en pipelines para subir reportes automáticamente después de un escaneo.

```
      - name: Generate Trivy Filesystem Report
        uses: aquasecurity/trivy-action@master
        with:
          scan-type: fs
          output: trivy-fs-report.json
          format: json
          scan-ref: .
          exit-code: 0

      - name: Upload Trivy reports
        run: |
          faraday-cli tool report -w "$WORKSPACE" trivy-fs-report.json
```

Integraciones

- **Herramientas compatibles**: Trivy, OpenVAS, Burp Suite, Nessus, Nikto, Nmap, OWASP ZAP, etc.
- **Colaboración**: Slack, Jira y otros mediante API.
- **Automatización**: API REST para crear workspaces, subir hallazgos y consultar datos.

UI

- Interfaz web accesible desde cualquier navegador.
- Funcionalidades:
  - Vista de **Assets** y vulnerabilidades asociadas.
  - **Dashboard** con métricas, severidad y gráficos.
  - Administración de **workspaces**, usuarios y roles.
  - Filtros avanzados para priorizar hallazgos.

Licencia

- Faraday **Community Edition**: Licencia GPLv3 (código abierto).
- Faraday **Professional & Corporate**: Licencia comercial con características avanzadas.

Facilidad de uso

- **Ventajas**:
  - Compatible con una gran variedad de herramientas.
  - CLI intuitiva y scripts de automatización.
  - Dashboard centralizado para equipos.
- **Desafíos**:
  - Requiere configuración inicial de SSL y credenciales.
  - Algunos formatos de reporte necesitan preprocesado, tienes que asegurarte de que esté en un formato estructurado y limpio que Faraday pueda leer.

Cumplimiento con estándares

Faraday no certifica por sí mismo estándares, pero **facilita el cumplimiento** al centralizar la gestión de vulnerabilidades y permitir auditorías trazables.

- **OWASP**: Compatible con flujos de seguridad recomendados, soporta OWASP ZAP y otras herramientas.
- **NIST 800-53 / 800-115**: Permite documentar, priorizar y remediar vulnerabilidades según guías NIST.
- **ISO 27001**: Ayuda a implementar controles de seguridad relacionados con la identificación y tratamiento de riesgos.