Evaluación de Herramientas para la Evaluación de Seguridad en Soluciones Desarrolladas

Se evaluaron ocho herramientas: Trivy, Grype + Syft, SonarQube, OWASP ZAP, Semgrep, Checkov, Snyk y Falco. Las herramientas open source ofrecen un ecosistema robusto y gratuito para cubrir distintos aspectos del ciclo de vida del desarrollo seguro. Se recomienda combinar varias de ellas según la fase del pipeline y el tipo de solución.

- Tabla de contenido
- Introducción
- Metodología y Criterios para la Evaluación de Herramientas de Seguridad en el Desarrollo de Soluciones Digitales del Estado Dominicano
- Análisis de Herramientas
    - Trivy (by Aqua Security)

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

- Comparativa Gráfica
- Recomendaciones
- Conclusión
- Bibliografía

Tabla de contenido

1. Introducción.
2. Metodología y Criterios para la Evaluación de Herramientas de Seguridad en el Desarrollo de Soluciones Digitales del Estado Dominicano.
   - Objetivo del Documento.
   - Metodología de Evaluación.
   - Criterios de Evaluación.
3. Análisis de Herramientas.
   - Trivy.
   - Grype + Syft.
   - SonarQube.
   - OWASP ZAP.
   - Semgrep.
   - Checkov.
   - Snyk.
   - Falco.
   - Faraday
4. Comparativa Gráfica.
5. Recomendaciones.
6. Conclusión.
7. Bibliografía.

Introducción

En el marco de la transformación digital del sector público, garantizar la seguridad de las soluciones desarrolladas es fundamental para proteger la integridad, confidencialidad y disponibilidad de los servicios digitales del Estado. La Dirección de Arquitectura Digital Gubernamental tiene como responsabilidad adoptar herramientas y prácticas que integren la seguridad como un componente esencial desde las primeras etapas del ciclo de vida del software.

Este informe proporciona un análisis comparativo y técnico de herramientas de evaluación de seguridad que pueden ser integradas en un enfoque DevSecOps en el ámbito gubernamental.

Metodología y Criterios para la Evaluación de Herramientas de Seguridad en el Desarrollo de Soluciones Digitales del Estado Dominicano

Objetivo del Documento

Seleccionar y recomendar herramientas de evaluación de seguridad que permitan fortalecer el desarrollo seguro de soluciones digitales, alineadas con estándares internacionales y buenas prácticas del sector.

Metodología de Evaluación

Se identificaron y evaluaron ocho herramientas reconocidas en la industria, considerando criterios relevantes definidos por el equipo de arquitectura digital. La información fue extraída de documentación oficial, comunidades técnicas, y pruebas en entornos de laboratorio.

Criterios de Evaluación

1. ¿Open Source?: Disponibilidad del código y posibilidad de uso sin licencia comercial.
2. Alcance de evaluación de seguridad: Cuales componentes o capas analiza (código, dependencias, contenedores, IaC, ejecución, etc.).
3. Integración con procesos DevSecOps: Facilidad para integrarse en flujos CI/CD.
4. Tipo de vulnerabilidades detectadas: Si realiza análisis SAST, DAST, SCA, IaC, runtime, secretos.
5. Facilidad de configuración y mantenimiento: Curva de aprendizaje y operatividad diaria
6. Soporte activo o comunidad: Existencia de soporte empresarial o comunidad activa.

7. Reportes comprensibles para Stakeholders: Facilidad de comprensión por distintos perfiles (técnicos, gerenciales).
8. Automatización en CI/CD: Si es posible su ejecución automatizada en pipelines.
9. Costo: Si tiene versiones gratuitas o es de pago.
10. Integración con herramientas usadas: Compatibilidad con herramientas como GitHub, Teams, Plane, Jira, etc.
11. Personalización de reglas o políticas: Posibilidad de definir reglas propias.
12. Cumplimiento de estándares: Si se alinea con OWASP, NIST, ISO 27001, etc

Análisis de Herramientas

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.

Comparativa Gráfica

Herramienta	¿Es Open Source?	Alcance de evaluación	Integrable en DevSecOps	Tipo de vulnerabilidades	Facilidad de uso	Comunidad / Soporte	Reportes comprensibles	Automatizable en CI/CD	Costo	Integraciones	Personalización de reglas	Cumplimiento de estándares
Trivy	✅ Sí	Imágenes Docker, código, IaC	✅ Sí	SCA, IaC, secretos, vuln.	⭐⭐⭐ Fácil	✅ Activa (Aqua Sec)	✅ Claro (CLI/JSON)	✅ Sí	✅ Gratis (OSS)	GitHub, GitLab, Jenkins	✅ Parcial (políticas)	✅ OWASP, CIS
Grype + Syft	✅ Sí	Análisis de imágenes/SBOM	✅ Sí	SCA (dependencias)	⭐⭐ Media	✅ Sí (Anchore)	CLI/JSON/SARIF	✅ Sí	✅ Gratis (OSS)	GitHub, GitLab	✅ Avanzado (rules.yaml)	✅ Parcial (CIS, OWASP)
SonarQube CE	✅ (CE) / ❌ (EE)	Código fuente (SAST)	✅ Sí	SAST	⭐⭐⭐ Fácil	✅ Amplia	✅ Muy buenos	✅ Sí	✅ CE gratis / EE pago	GitHub, GitLab, Jenkins	✅ Sí	✅ OWASP Top 10
OWASP ZAP	✅ Sí	Web apps (DAST)	✅ Sí	DAST	⭐⭐ Intermedio	✅ OWASP	✅ GUI + JSON	✅ Sí	✅ Gratis	Jenkins, GitHub, Jira	✅ Avanzado	✅ OWASP Top 10
Semgrep	✅ Sí	Código (SAST ligero + rules)	✅ Sí	SAST	⭐⭐⭐ Fácil	✅ Activa	✅ Personalizables	✅ Sí	✅ Gratis (OSS) / pago	GitHub, GitLab, Jira	✅ Muy flexible	✅ OWASP, PCI, etc.
Checkov	✅ Sí	Infraestructura como código	✅ Sí	IaC (Terraform, etc.)	⭐⭐⭐ Fácil	✅ Activa (Bridgecrew)	✅ CLI/JSON	✅ Sí	✅ Gratis (OSS)	GitHub, GitLab, Terraform	✅ Sí	✅ CIS, NIST

Herramienta	¿Es Open Source?	Alcance de evaluación	Integrable en DevSecOps	Tipo de vulnerabilidades	Facilidad de uso	Comunidad / Soporte	Reportes comprensibles	Automatizable en CI/CD	Costo	Integraciones	Personalización de reglas	Cumplimiento de estándares
Snyk	✅ (pero tiene CLI OSS)	Código, IaC, dependencias	✅ Sí	SAST, SCA, IaC	⭐⭐⭐ Fácil	✅ Comercial y activa	✅ Excelente GUI/CLI	✅ Sí	✅ Pago (freemium)	GitHub, GitLab, Jira	✅ Sí (limitado OSS)	✅ OWASP, NIST
Falco	✅ Sí	Tiempo de ejecución (runtime)	✅ Sí	Runtime anomalies	⭐⭐ Medio	✅ CNCF, Sysdig	Logs + alertas	✅ Sí	✅ Gratis (OSS)	SIEM, Prometheus, etc.	✅ Sí (reglas YAML)	✅ CIS Benchmarks

Recomendaciones

Herramientas recomendadas para incluir en el stack tecnológico (open source y gratuitas)

Herramienta	Licencia	Motivo para incluir
Trivy	Apache 2.0 (OSS)	Escaneo de imágenes, código fuente, repositorios, configuración IaC, secretos. Rápido y versátil.
Grype + Syft	Apache 2.0 (OSS)	SBOM + análisis de vulnerabilidades. Ideal para cumplimiento con NIST y gestión de inventario.
OWASP ZAP	Apache 2.0 (OSS)	Escaneo de aplicaciones web. GUI, CLI y automatizable. 100% alineado con OWASP.
Semgrep	LGPL (Free tier)	Excelente para escaneo SAST. Reglas YAML personalizadas, muy útil para desarrollo seguro.
Checkov	Apache 2.0 (OSS)	IaC security. Compatible con Terraform, Kubernetes, CloudFormation, etc.
Falco	Apache 2.0 (OSS)	Detección de amenazas en tiempo real en contenedores y Kubernetes. Ideal para producción.

Estas herramientas:

- Son **open source reales** o con licencia suficientemente abierta para entornos empresariales.
- Funcionan bien en **máquinas virtuales** y en local.
- Se integran fácilmente en pipelines CI/CD.
- Cumplen con estándares como OWASP, NIST, ISO 27001, CIS Benchmarks.

Herramientas útiles pero con limitaciones en la versión gratuita.

Herramienta	Licencia	Uso recomendado / limitación
SonarQube	Community Edition OSS	Muy útil para SAST. La versión gratuita no incluye todos los lenguajes ni reglas avanzadas. Instalable en VM.
Snyk	Gratis con limitaciones	Muy potente, pero requiere cuenta. En la versión free se limita el número de escaneos y proyectos privados.

Puedes usarse si:

- Si se esta dispuesto a autohospedarlas (en el caso de SonarQube).
- Aceptas limitaciones funcionales para equipos pequeños (en el caso de Snyk).

Herramientas que podrías descartar (para uso exclusivo OSS / sin costo)

Herramienta	Motivo para descartar o evitar
Snyk (versión cloud)	Aunque potente, sus planes gratuitos son limitados para uso en producción o entornos empresariales. Necesita upgrade para características clave (políticas, tickets en Jira, control RBAC, etc.).
SonarQube (versión Enterprise)	Solo necesaria si necesitas análisis profundo de C++, COBOL, Salesforce, o reglas personalizadas complejas.

Recomendaciones según casos de uso:

- **Para Imágenes Docker y Seguridad general (SCA, IaC, secretos):**
  ➤ *Trivy* (rápido, simple, bien mantenido)
  ➤ *Grype + Syft* si necesitas generar o analizar SBOM
- **Para análisis de código fuente (SAST):**
  ➤ *SonarQube* si buscas reportes detallados por roles
  ➤ *Semgrep* si necesitas reglas personalizadas y flexibilidad
- **Para vulnerabilidades en infraestructura como código (IaC):**
  ➤ *Checkov* (fuerte soporte para Terraform, CloudFormation)
- **Para pruebas activas de aplicaciones web (DAST):**
  ➤ *OWASP ZAP* (ideal para pruebas automatizadas de apps web)
- **Para detección en tiempo de ejecución:**
  ➤ *Falco* (detección de anomalías en contenedores/Kubernetes)

Stack tecnológico DevSecOps recomendado (100% open source / gratuito)

Fase DevSecOps	Herramienta	Funcionalidad
SAST	Semgrep, SonarQube CE	Escaneo de código fuente y detección de bugs
SCA	Trivy, Grype + Syft	Escaneo de dependencias y vulnerabilidades (CVEs)
IaC Security	Checkov, Trivy	Revisión de infraestructura como código
DAST	OWASP ZAP	Escaneo dinámico de aplicaciones web
SBOM	Syft	Generación de lista de materiales de software
Runtime Security	Falco	Monitoreo de comportamiento en contenedores/K8s

Conclusión

Para entornos gubernamentales, se recomienda adoptar un stack basado en herramientas open source con amplio soporte comunitario. Luego del análisis técnico de un conjunto de herramientas de seguridad open source, se concluye que el Estado dominicano puede adoptar un **stack tecnológico de ciberseguridad moderno, eficaz y sin costos de licenciamiento**, alineado con estándares internacionales, con plena capacidad de ser desplegado y operado desde infraestructuras gubernamentales (máquinas virtuales, servidores propios, nubes estatales o ambientes híbridos).

Las herramientas evaluadas —**Trivy, Grype + Syft, SonarQube (Community Edition), Semgrep, OWASP ZAP, Checkov y Falco**— permiten cubrir de manera integral las distintas capas de seguridad en el ciclo de vida del desarrollo y despliegue de sistemas digitales, incluyendo:

- Análisis estático y dinámico de aplicaciones.
- Escaneo de vulnerabilidades en código, dependencias e infraestructura como código.
- Detección de amenazas en tiempo real en entornos de contenedores o Kubernetes.
- Generación y análisis de listas de materiales de software (SBOM).

Estas herramientas están alineadas con marcos y normas reconocidas como:

- **OWASP Top 10** (para la seguridad de aplicaciones web),
- **NIST 800-53** y **NIST SSDF** (para la gestión de vulnerabilidades y desarrollo seguro),
- **ISO/IEC 27001** (para sistemas de gestión de seguridad de la información),
- **CIS Benchmarks** (para configuraciones seguras de plataformas y contenedores).

Además, su carácter open source ofrece las siguientes ventajas para el Estado:

- **Reducción de costos** al eliminar dependencias de licencias comerciales.
- **Soberanía tecnológica**, al permitir su despliegue interno sin enviar datos a terceros.
- **Escalabilidad y flexibilidad**, al integrarse fácilmente con plataformas ya adoptadas como GitHub, Jira, o entornos CI/CD.
- **Transparencia y audibilidad**, en línea con los principios de gobierno abierto y fortalecimiento institucional.

En conclusión, se recomienda la incorporación progresiva de este stack de herramientas dentro de una **estrategia nacional de DevSecOps y gestión de riesgos digitales**, para fortalecer la arquitectura de seguridad de los sistemas gubernamentales, mejorar la capacidad de respuesta ante amenazas cibernéticas y avanzar hacia un ecosistema digital estatal resiliente, sostenible y conforme a las mejores prácticas internacionales. Integrar estas herramientas en pipelines de CI/CD mejora la visibilidad, cumplimiento y mitigación temprana de riesgos.

Bibliografía

Trivy

Aqua Security. (s.f.). Trivy - Escáner de vulnerabilidades. Aqua Security.

Aqua Security. (s.f.). Trivy GitHub Action. GitHub.

Aqua Security. (s.f.). Extensión para Visual Studio Code. GitHub.

DevOps Tales. (2023). Uso de Trivy Operator para validación de imágenes.

Locustbaby. (s.f.). Interfaz gráfica para Trivy. GitHub.

Aqua Security. (2024). Postee v2.9.0.

Aqua Security. (2024). Blueprints para Trivy Operator.

Grype + Syft

Anchore. (s.f.). Grype - Análisis de vulnerabilidades de contenedores.

Anchore. (s.f.). Herramientas de escaneo de seguridad.

SecureCodeBox. (2023). Consumo de SBOM con Grype.

Syft Analytics. (s.f.). Syft - Generador de SBOM.

Anchore. (s.f.). Acción de GitHub para escaneo con Grype.

SonarQube

SonarSource. (s.f.). SonarQube.

SonarSource. (s.f.). Acción GitHub para SonarQube.

SonarSource. (s.f.). Integración con GitHub.

SonarSource. (s.f.). Extensión VSCode.

Atlassian Marketplace. (s.f.). Conector SonarQube para Jira.

Toilatester. (s.f.). Notificador SonarQube para Microsoft Teams.

Semgrep

Semgrep. (s.f.). Documentación general.

Semgrep. (s.f.). Acción GitHub para Semgrep.

Semgrep. (s.f.). Extensión de VSCode.

Semgrep. (s.f.). Integración con Slack.

Semgrep. (s.f.). Integración con Jira.

Semgrep. (s.f.). Referencia CLI y formatos SARIF/JUnit.

OWASP ZAP

OWASP. (s.f.). OWASP ZAP.

OWASP. (s.f.). Escaneo base con Docker.

Zaproxy. (s.f.). GitHub Action oficial.

Zakrush. (s.f.). Scripts de API para ZAP.

Balasooriya, K. (2021). Exportar alertas de ZAP a Jira.

Checkov

Bridgecrew. (s.f.). Checkov.

Bridgecrew. (s.f.). Acción GitHub de Checkov.

Bridgecrew. (s.f.). Extensión de VSCode.

Bridgecrew. (s.f.). Integración con DefectDojo.

Atlassian. (s.f.). REST API Jira.

Checkov. (s.f.). Soporte para SARIF y JUnit.

Checkov. (s.f.). Referencia de CLI.

Checkov. (s.f.). Escaneo de Terraform Plan.

Snyk

Snyk. (s.f.). Página principal.

Snyk. (s.f.). Acción GitHub.

Snyk. (s.f.). Integración con Jira.

Snyk. (s.f.). Integración con Slack.

Snyk. (s.f.). VSCode Extension.

Snyk. (s.f.). CLI y Web UI.

Snyk. (s.f.). Blog sobre buenas prácticas.

Kimpel, H. (2023). Webhooks y suscripciones.

Falco

Falco. (s.f.). Documentación oficial.

Falco. (s.f.). Conceptos: Outputs y Canales.

Falco. (s.f.). GitHub Exporter.

FreeCodeCamp. (2023). Integración con Prometheus, Grafana y Docker.

Elastic. (s.f.). Integración Falco y Elastic Security.

Port. (s.f.). Integraciones webhook de Falco.

Herramientas Complementarias

CrowdStrike Marketplace. (s.f.). Soar actions para Teams.

SendSonar. (s.f.). Integración con Slack.

Digicert. (s.f.). Integración Sonar y Slack.

O’Reilly. (2019). Practical Security Automation.

YouTube. (2023). Integración de DefectDojo.

Grafana. (s.f.). Sitio oficial.

Prometheus. (s.f.). Prometheus + Kubernetes.

Elastic. (s.f.). Elasticsearch.