TABLAS DE INDICADORES Y MAPEO DE PROCESOS

Sistema HybridSecScan - Auditoria Hibrida SAST + DAST

> **Proyecto:** Sistema de Auditoria Automatizada Hibrida para APIs REST
> **Autor:** Oscar Isaac Laguna Santa Cruz
> **Universidad:** UNMSM - Facultad de Ingenieria de Sistemas e Informatica
> **Fecha:** Noviembre 2024

TABLA 1: INDICADORES CUANTITATIVOS

Tabla 1

Indicadores Cuantitativos del Sistema

Indicadores	Valor
Tasa de deteccion de vulnerabilidades OWASP API Top 10	>= 85%
Tasa de falsos positivos generados por el sistema	<= 15%
Tiempo promedio de analisis completo (SAST + DAST)	<= 20 min
Cobertura de categorias OWASP API Security Top 10	100% (10/10)

DESCRIPCION DE INDICADORES

1. Tasa de deteccion de vulnerabilidades OWASP API Top 10

Definicion: Porcentaje de vulnerabilidades reales identificadas por el sistema hibrido en relacion al total de vulnerabilidades presentes en APIs de prueba documentadas.

Formula:

```
Tasa de deteccion = (Vulnerabilidades detectadas / Vulnerabilidades totales conocidas) × 100
```

Valor objetivo: >= 85%

Metodo de medicion: Analisis experimental con APIs vulnerables documentadas (OWASP Juice Shop API, VAmPI, crAPI).

---

2. Tasa de falsos positivos generados por el sistema

Definicion: Proporcion de alertas incorrectas reportadas que no corresponden a vulnerabilidades reales, generando trabajo innecesario para desarrolladores.

Formula:

```
Tasa de FP = (Numero de falsos positivos / Total de hallazgos reportados) × 100
```

Valor objetivo: <= 15%

Metodo de medicion: Verificacion manual por experto en seguridad + Validacion mediante intentos de explotacion.

---

3. Tiempo promedio de analisis completo (SAST + DAST)

Definicion: Duracion total del proceso de auditoria hibrida desde la carga de la especificacion OpenAPI hasta la generacion del reporte final en formatos PDF y JSON.

Formula:

```
Tiempo promedio = Suma(tiempos de analisis individuales) / Numero de analisis ejecutados
```

Valor objetivo: <= 20 minutos

Metodo de medicion: Logs del sistema con timestamps de inicio/fin + Modulo de monitoreo de rendimiento.

Desglose:

- SAST (analisis estatico): 1-2 min
- Correlacion ML: 0.5-1 min
- DAST (analisis dinamico): 10-15 min
- Consolidacion y reportes: 1-2 min

---

4. Cobertura de categorias OWASP API Security Top 10

Definicion: Cantidad de categorias del estandar OWASP API Security Top 10 (2023) que el sistema es capaz de detectar mediante analisis hibrido.

Formula:

```
Cobertura = (Categorias OWASP detectadas / 10 categorias totales) × 100
```

Valor objetivo: 100% (10/10 categorias)

Metodo de medicion: Analisis de clasificacion en reportes JSON/PDF + Validacion con matriz de trazabilidad OWASP.

Categorias evaluadas:

1. API1:2023 - Broken Object Level Authorization (BOLA)
2. API2:2023 - Broken Authentication
3. API3:2023 - Broken Object Property Level Authorization
4. API4:2023 - Unrestricted Resource Consumption
5. API5:2023 - Broken Function Level Authorization
6. API6:2023 - Unrestricted Access to Sensitive Business Flows
7. API7:2023 - Server Side Request Forgery (SSRF)
8. API8:2023 - Security Misconfiguration
9. API9:2023 - Improper Inventory Management
10. API10:2023 - Unsafe Consumption of APIs

JUSTIFICACION DE VALORES OBJETIVO

Indicador	Linea Base (Herramientas Actuales)	Valor Objetivo HybridSecScan	Mejora Esperada
Tasa de deteccion	65% (OWASP ZAP solo)	>= 85%	+20 puntos porcentuales
Tasa de falsos positivos	35% (OWASP ZAP solo)	<= 15%	-20 puntos porcentuales
Tiempo de analisis	25 min (analisis manual)	<= 20 min	-5 minutos (-20%)
Cobertura OWASP	70% (7/10 categorias)	100%	+30 puntos porcentuales

INSTRUMENTOS DE MEDICION

APIs de Prueba Documentadas

- **OWASP Juice Shop API** - Vulnerabilidades intencionadas documentadas
- **VAmPI** (Vulnerable API) - API REST vulnerable para pruebas
- **crAPI** (Completely Ridiculous API) - Proyecto OWASP para testing
- **DVWA REST API** - Damn Vulnerable Web Application con endpoints REST
- **APIs custom** - Desarrolladas especificamente con vulnerabilidades OWASP API Top 10

Herramientas de Validacion

- Verificacion manual por experto en seguridad
- Intentos de explotacion controlados
- Analisis de logs del sistema
- Modulo de monitoreo de rendimiento
- Comparacion con reportes de herramientas tradicionales (OWASP ZAP, Burp Suite)

PLAN DE VALIDACION EXPERIMENTAL

Fase 1: Preparacion (1 semana)

- Seleccionar 5 APIs de prueba con vulnerabilidades documentadas
- Configurar entorno de experimentacion (Docker, base de datos, servidores de prueba)
- Preparar plantillas de recoleccion de datos

Fase 2: Ejecucion (2-3 semanas)

- Ejecutar 10 analisis por cada API (50 analisis totales)
- Registrar metricas de los 4 indicadores
- Documentar observaciones y anomalias

Fase 3: Validacion (1 semana)

- Verificar manualmente todos los hallazgos
- Calcular promedios y desviaciones estandar
- Comparar contra linea base de herramientas tradicionales

Fase 4: Analisis Estadistico (1 semana)

- Pruebas de hipotesis (t-test para IND-01 e IND-02)

- Calculo de intervalos de confianza (95%)
- Generacion de graficos comparativos

MATRIZ DE TRAZABILIDAD INDICADORES VS OBJETIVOS

Indicador	Objetivo Especifico de la Tesis	Aporte al Sistema
IND-01: Tasa de deteccion	OE5: Evaluacion experimental y comparativa	Demuestra efectividad del enfoque hibrido
IND-02: Tasa de FP	OE5: Evaluacion experimental y comparativa	Valida precision y usabilidad practica
IND-03: Tiempo de analisis	OE4: Definicion de metodologia de validacion	Confirma viabilidad para CI/CD
IND-04: Cobertura OWASP	OE1: Analisis del estado del arte	Garantiza cumplimiento de estandar

---

EJEMPLO DE RESULTADOS ESPERADOS

Caso de Prueba: OWASP Juice Shop API

Configuracion:

- Endpoints analizados: 42
- Vulnerabilidades conocidas documentadas: 20
- Tipo de analisis: Hibrido (SAST + DAST)

Resultados:

Indicador	Resultado Obtenido	Cumple Objetivo
Tasa de deteccion	18/20 = 90%	Si (>=85%)
Tasa de falsos positivos	3/21 = 14.3%	Si (<=15%)
Tiempo de analisis	17 minutos	Si (<=20 min)
Cobertura OWASP	10/10 = 100%	Si (100%)

Clasificacion de Vulnerabilidades Detectadas:

- API1:2023 (BOLA): 4 vulnerabilidades
- API2:2023 (Broken Auth): 2 vulnerabilidades
- API3:2023 (Mass Assignment): 3 vulnerabilidades
- API5:2023 (BFLA): 2 vulnerabilidades
- API8:2023 (Security Misconfiguration): 5 vulnerabilidades
- API10:2023 (Unsafe Consumption): 2 vulnerabilidades

TABLA 2: MAPEO DE PROCESOS

Tabla 2

Mapeo de Procesos del Sistema de Auditoria Automatizada

Proceso	Subproceso	Entrada	Salida	Responsable	Tiempo Estimado
1. Autenticacion y Configuracion	1.1 Login de usuario	Credenciales (usuario/contrasena)	Token JWT valido	Modulo de Autenticacion	2-5 seg
	1.2 Seleccion de tipo de analisis	Opcion seleccionada (SAST/DAST/Hibrido)	Configuracion de analisis	Frontend Web	10-20 seg
	1.3 Carga de especificacion OpenAPI	Archivo OpenAPI/Swagger (YAML/JSON)	Archivo validado y parseado	API Gateway	5-10 seg
	1.4 Configuracion de parametros	Parametros de escaneo (URL, tokens, etc.)	Configuracion guardada en BD	API Gateway	15-30 seg
2. Validacion de Entrada	2.1 Validacion de archivo OpenAPI	Archivo cargado	Esquema validado sintacticamente	Modulo de Validacion	5-10 seg
	2.2 Extraccion de endpoints	Especificacion OpenAPI	Lista de endpoints y parametros	Parser OpenAPI	10-20 seg
	2.3 Almacenamiento de metadatos	Endpoints extraidos	Metadatos en base de datos	Sistema de Persistencia	2-5 seg
3. Analisis Estatico (SAST)	3.1 Inicializacion de Semgrep	Configuracion SAST	Motor Semgrep listo	Modulo SAST	5-10 seg
	3.2 Analisis sintactico y semantico	Especificacion OpenAPI	Estructura analizada	Semgrep Engine	20-40 seg
	3.3 Deteccion de vulnerabilidades	Estructura analizada	Vulnerabilidades identificadas	Semgrep Rules	30-60 seg
	3.4 Generacion de hallazgos SAST	Vulnerabilidades identificadas	Reporte JSON con hallazgos	Modulo SAST	10-15 seg
4. Correlacion con Machine Learning	4.1 Carga de modelo Random Forest	Modelo entrenado (.pkl)	Modelo cargado en memoria	Motor de Correlacion ML	3-5 seg
	4.2 Extraccion de caracteristicas	Hallazgos SAST + metadatos	15 features por endpoint	Feature Extractor	10-20 seg
	4.3 Prediccion de correlacion	Features extraidas	Probabilidad de vulnerabilidad (0-1)	Modelo Random Forest	5-15 seg
	4.4 Priorizacion de endpoints	Probabilidades calculadas	Lista priorizada de endpoints	Ranking Algorithm	5-10 seg

Proceso	Subproceso	Entrada	Salida	Responsable	Tiempo Estimado
5. Analisis Dinamico (DAST)	5.1 Inicializacion de Docker ZAP	Comando Docker	Contenedor OWASP ZAP activo	Modulo DAST	30-60 seg
	5.2 Configuracion de contexto	URL base + autenticacion	Contexto configurado en ZAP	ZAP Context Manager	10-20 seg
	5.3 Importacion de OpenAPI a ZAP	Especificacion OpenAPI	Arbol de endpoints en ZAP	ZAP OpenAPI Plugin	15-30 seg
	5.4 Generacion de casos de prueba	Endpoints priorizados	Test cases de ataque	ZAP Test Generator	20-40 seg
	5.5 Ejecucion de ataques	Test cases generados	Alertas de vulnerabilidades	ZAP Active Scanner	10-15 min
	5.6 Obtencion de resultados	Escaneo completado	Reporte XML/JSON de ZAP	ZAP API Client	10-20 seg
6. Consolidacion de Resultados	6.1 Unificacion SAST + DAST	Reporte SAST + Reporte DAST	Reporte unificado preliminar	Motor de Consolidacion	15-30 seg
	6.2 Normalizacion de formato	Reporte preliminar	Formato estandarizado	Data Normalizer	10-20 seg
	6.3 Correlacion de duplicados	Hallazgos normalizados	Hallazgos unicos identificados	Algoritmo ML de correlacion	20-40 seg
	6.4 Eliminacion de falsos positivos	Hallazgos correlacionados	Hallazgos validados	Filter ML Model	15-30 seg
	6.5 Asignacion de severidad	Hallazgos validados	Severidad CVSS asignada	CVSS Calculator	10-20 seg
	6.6 Clasificacion OWASP	Hallazgos con severidad	Categorias OWASP API Top 10	OWASP Classifier	10-15 seg
7. Sistema de Evaluacion	7.1 Calculo de metricas	Hallazgos clasificados	Total de vulnerabilidades por tipo	Metrics Calculator	5-10 seg
	7.2 Calculo de score de seguridad	Metricas calculadas	Puntuacion 0-100	Scoring Algorithm	5-10 seg
	7.3 Determinacion de nivel de riesgo	Score calculado	Nivel (Critico/Alto/Medio/Bajo)	Risk Evaluator	2-5 seg
8. Generacion de Reportes	8.1 Generacion de estructura	Hallazgos consolidados	Estructura de reporte	Report Builder	5-10 seg

Proceso	Subproceso	Entrada	Salida	Responsable	Tiempo Estimado
	8.2 Organizacion por severidad	Estructura generada	Hallazgos ordenados	Report Organizer	5-10 seg
	8.3 Inclusion de recomendaciones	Hallazgos organizados	Recomendaciones de mitigacion	Recommendation Engine	10-20 seg
	8.4 Generacion de evidencias y graficos	Datos consolidados	Graficos y tablas	Chart Generator (Matplotlib)	15-30 seg
	8.5 Exportacion a PDF	Reporte completo	Archivo PDF ejecutivo	ReportLab Library	20-40 seg
	8.6 Exportacion a JSON	Reporte completo	Archivo JSON programatico	JSON Encoder	5-10 seg
	8.7 Generacion de Dashboard Web	Reporte completo	Dashboard HTML interactivo	Web Dashboard Module	10-20 seg
9. Notificaciones y Persistencia	9.1 Envio de notificaciones	Reporte finalizado + config	Email/Webhook enviados	Notification Service	3-7 seg
	9.2 Almacenamiento en base de datos	Reporte completo	Registro en BD con timestamp	Database Manager	2-5 seg
	9.3 Generacion de logs de auditoria	Eventos del sistema	Logs persistidos	Logging System	Continuo

DESCRIPCION DETALLADA DE PROCESOS PRINCIPALES

Proceso 1: Autenticacion y Configuracion (Total: 30-65 segundos)

Objetivo: Establecer sesion de usuario y configurar parametros iniciales del analisis.

Flujo:

1. Usuario ingresa credenciales → Sistema valida contra BD → Token JWT generado
2. Usuario selecciona tipo de analisis (SAST/DAST/Hibrido)
3. Usuario carga archivo OpenAPI/Swagger
4. Usuario configura parametros adicionales (URL base, tokens de autenticacion, etc.)

Salida: Configuracion lista para iniciar analisis

Proceso 2: Validacion de Entrada (Total: 17-35 segundos)

Objetivo: Validar sintaxis y estructura de la especificacion OpenAPI antes de analisis.

Flujo:

1. Validacion sintactica del archivo YAML/JSON

2. Extraccion de endpoints, metodos HTTP, parametros y esquemas
3. Almacenamiento de metadatos en base de datos SQLite

Salida: Lista de endpoints validados y persistidos

Proceso 3: Analisis Estatico - SAST (Total: 1-2 minutos)

Objetivo: Detectar vulnerabilidades mediante analisis de especificacion OpenAPI sin ejecutar la API.

Tecnicas aplicadas:

- Analisis de esquemas de autenticacion (JWT, OAuth2, API Keys)
- Deteccion de configuraciones inseguras (CORS, headers, metodos HTTP)
- Validacion de esquemas de entrada/salida
- Identificacion de endpoints sin autenticacion

Herramientas: Semgrep + Reglas personalizadas

Salida: Reporte JSON con vulnerabilidades SAST

Proceso 4: Correlacion con Machine Learning (Total: 23-50 segundos)

Objetivo: Priorizar endpoints criticos para analisis dinamico mediante modelo Random Forest.

Caracteristicas extraidas (15 features):

1. Numero de vulnerabilidades SAST detectadas
2. Severidad promedio SAST
3. Presencia de autenticacion
4. Metodos HTTP permitidos
5. Numero de parametros por endpoint
6. Complejidad de esquemas JSON
7. Presencia de validacion de entrada
8. Tipo de autenticacion (JWT/OAuth2/None)
9. Endpoints con operaciones sensibles (DELETE, PUT)
10. Presencia de rate limiting
11. Configuracion CORS
12. Longitud de path del endpoint
13. Numero de respuestas definidas
14. Presencia de ejemplos en OpenAPI
15. Profundidad de objetos JSON

Modelo: Random Forest Classifier entrenado con 500 APIs

Salida: Lista de endpoints ordenados por probabilidad de vulnerabilidad

---

Proceso 5: Analisis Dinamico - DAST (Total: 10-17 minutos)

Objetivo: Ejecutar ataques activos sobre la API en ejecucion para validar vulnerabilidades explotables.

Tipos de ataques ejecutados:

- Inyeccion SQL (SQL Injection)
- Cross-Site Scripting (XSS)
- Broken Object Level Authorization (BOLA)

- Fuzzing de parametros
- Manipulacion de tokens JWT
- Ataques de denegacion de servicio (DoS)
- Server-Side Request Forgery (SSRF)

Herramienta: OWASP ZAP en contenedor Docker

Salida: Reporte XML/JSON de ZAP con alertas

Proceso 6: Consolidacion de Resultados (Total: 1.5-2.5 minutos)

Objetivo: Unificar hallazgos SAST y DAST, eliminar duplicados y asignar severidad.

Algoritmo de correlacion:

```
Para cada hallazgo_DAST:
    Para cada hallazgo_SAST:
        Si (mismo_endpoint AND misma_vulnerabilidad_OWASP):
            Marcar como duplicado
            Consolidar evidencias
            Asignar severidad maxima entre ambos
```

Calculo de severidad CVSS:

- Base Score: 0-10
- Temporal Score: Ajuste por explotabilidad
- Environmental Score: Contexto de la organizacion

Salida: Reporte consolidado sin duplicados con severidad CVSS

Proceso 7: Sistema de Evaluacion (Total: 12-25 segundos)

Objetivo: Calcular score global de seguridad y nivel de riesgo.

Formula de Score (0-100):

```
Score = 100 - (Σ(vulnerabilidades × peso_severidad) / factor_normalizacion)

Pesos de severidad:
- Critica: 10 puntos
- Alta: 7 puntos
- Media: 4 puntos
- Baja: 1 punto
```

Niveles de riesgo:

- **Critico:** Score 0-40 (>=1 vulnerabilidad critica)
- **Alto:** Score 41-60 (>=3 vulnerabilidades altas)
- **Medio:** Score 61-80 (solo vulnerabilidades medias/bajas)
- **Bajo:** Score 81-100 (<=2 vulnerabilidades bajas)

Salida: Score numerico + Nivel de riesgo categorico

Proceso 8: Generacion de Reportes (Total: 1-2 minutos)

Objetivo: Producir reportes en multiples formatos para diferentes audiencias.

Formato PDF (Ejecutivo):

- Resumen ejecutivo con score global
- Graficos de distribucion de vulnerabilidades
- Top 5 vulnerabilidades criticas
- Recomendaciones priorizadas
- Evidencias con screenshots

Formato JSON (Programatico):

```
{
  "scan_id": "scan_20241123_001",
  "timestamp": "2024-11-23T10:15:00Z",
  "score": 72,
  "risk_level": "Medium",
  "vulnerabilities": [
    {
      "id": "VULN-001",
      "title": "Broken Object Level Authorization",
      "severity": "High",
      "cvss": 8.2,
      "owasp": "API1:2023",
      "endpoint": "/api/users/{id}",
      "method": "GET",
      "evidence": "...",
      "recommendation": "..."
    }
  ]
}
```

Dashboard Web:

- Graficos interactivos con Chart.js
- Filtros por severidad y categoria OWASP
- Exportacion de datos
- Historico de escaneos

Salida: 3 archivos (PDF, JSON, HTML)

Proceso 9: Notificaciones y Persistencia (Total: 5-12 segundos)

Objetivo: Notificar a stakeholders y persistir resultados para trazabilidad.

Canales de notificacion:

- Email (SMTP)
- Slack (Webhook)
- Microsoft Teams (Webhook)
- Webhooks personalizados

Base de datos:

- SQLite para desarrollo
- PostgreSQL para produccion
- Esquema: `scans`, `vulnerabilities`, `recommendations`, `logs`

Logs de auditoria:

- Timestamp de cada operacion
- Usuario que ejecuto el analisis
- Errores y excepciones
- Tiempo de ejecucion por modulo

Salida: Notificaciones enviadas + Datos persistidos

FLUJO COMPLETO DEL SISTEMA

```
Inicio → Autenticacion (30s) → Validacion (20s) →
    ↓
Decision: ¿Tipo de analisis?
    ↓
├─ Solo SAST → SAST (1.5min) → Reportes (1min) → Fin
├─ Solo DAST → DAST (15min) → Reportes (1min) → Fin
└─ Hibrido → SAST (1.5min) → Correlacion ML (40s) →
             DAST (15min) → Consolidacion (2min) →
             Evaluacion (15s) → Reportes (1min) →
             Notificaciones (10s) → Fin

Tiempo total maximo (Hibrido): ~20 minutos
```

MATRIZ DE RESPONSABILIDADES

Componente	Tecnologia	Responsabilidad
Frontend Web	React + TypeScript	Interfaz de usuario, carga de archivos
API Gateway	FastAPI (Python)	Validacion, enrutamiento, autenticacion JWT
Motor SAST	Semgrep + Python	Analisis estatico de OpenAPI
Motor ML	Scikit-learn + Random Forest	Correlacion y priorizacion
Motor DAST	OWASP ZAP + Docker	Analisis dinamico con ataques activos
Consolidacion	Python + Pandas	Unificacion y eliminacion de duplicados
Reportes	ReportLab + Matplotlib	Generacion de PDF, JSON, Dashboard
Base de Datos	SQLite / PostgreSQL	Persistencia de resultados
Notificaciones	SMTP + Webhooks	Envio de alertas

CONSIDERACIONES OPERACIONALES

Requisitos de Sistema

- **CPU:** 4 cores minimo (8 cores recomendado)
- **RAM:** 8 GB minimo (16 GB recomendado)

- **Disco:** 20 GB de espacio libre
- **Docker:** Version 20.10+
- **Python:** Version 3.9+

Escalabilidad

- **Analisis concurrentes:** Hasta 5 escaneos simultaneos
- **APIs por escaneo:** 50 endpoints maximo recomendado
- **Historico:** Retencion de 90 dias en BD

Limitaciones

- DAST requiere API accesible via HTTP/HTTPS
- Analisis de APIs con autenticacion compleja requiere configuracion manual
- Tiempo de DAST depende del numero de endpoints (1 min/endpoint aprox.)

REFERENCIAS

- OWASP Foundation. (2023). *OWASP API Security Top 10 - 2023*. https://owasp.org/API-Security/
- Corradini, D., et al. (2023). Automated black-box testing of mass assignment vulnerabilities in RESTful APIs. *Software Testing, Verification and Reliability*.
- Atlidakis, V., et al. (2019). RESTler: Stateful REST API fuzzing. *IEEE/ACM ICSE*.
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine.
- Salt Security. (2023). *State of API Security Report Q1 2023*.

Elaborado por: Oscar Isaac Laguna Santa Cruz
Institucion: Universidad Nacional Mayor de San Marcos - FISI
Escuela Profesional: Ingenieria de Software
Fecha: Noviembre 2024
Version: 1.0