VI CONGRESO INTERNACIONAL DE CIBERSEGURIDAD, PREVENCIÓN DE FRAUDES Y SEGURIDAD FÍSICA

Fortaleciendo la postura de AppSec a escala

La IA como herramienta que cambia el *workflow*

Vladimir Villa | CEO - Fluid Attacks

¿Cuáles son los cambios que estamos experimentando?

Cambios en el ***workflow***:

- **Desarrollo**
- **Inyección**
- **Detección**
- **Remediación**

De vulnerabilidades

Cambios en las **arquitecturas** de las aplicaciones, introduciendo **nuevos componentes** y transformando las **UI**.

Aplicaciones basadas en IA

Objetivo de evaluación

Nuevos ambientes de desarrollo y ejecución en CLI, IDEs, LLMs y MCP.

▶ Cadena de suministro - SCA

Cambio de arquitectura

Chat / Interfaz de voz

▼ Campo de input ilimitado

▼ Menos debilidades, más vectores de ataque (lógica de negocio, sobre todo)

Backends MCP/API

▼ **Pruebas tradicionales:**

- SAST
- SCA
- DAST API
- PTaaS API
- SCR

Desarrollo de *software* por agentes

1

Código inseguro
Más código, más vulnerabilidades, mayor riesgo

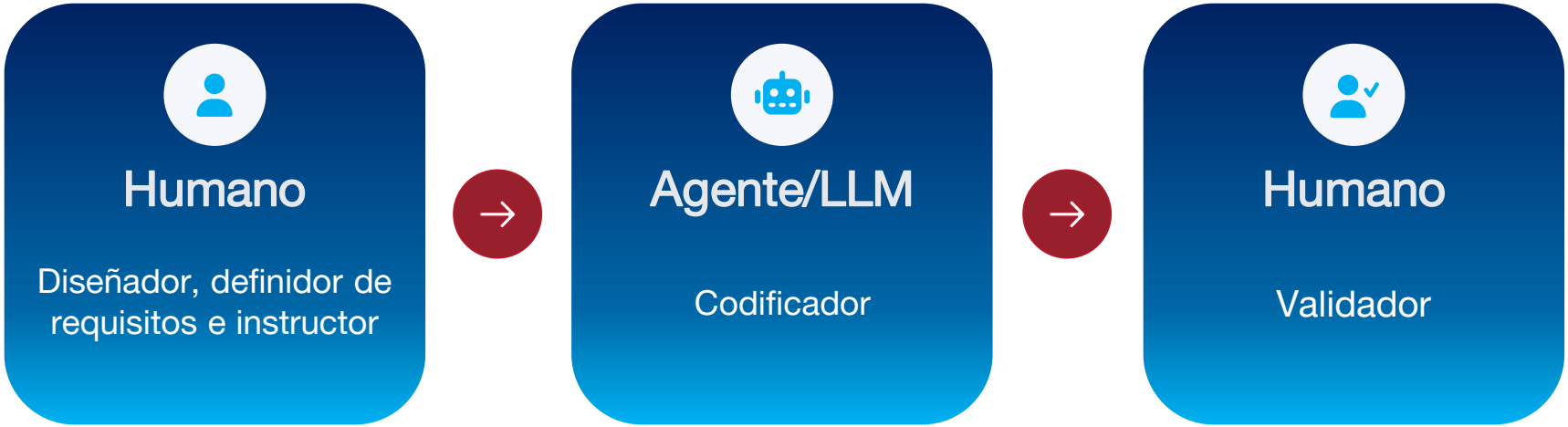
Desarrollador (validador)	Cadena de suministro - SCA	Pruebas tradicionales:
• *Instrucciones de desarrollo seguro* • *¿Qué revisar? peer review*		• SAST • SCA • DAST API • PTaaS API • SCR

2

Código seguro
Más código, más superficie de ataque, más vulnerabilidades
Vulnerabilidades de lógica de negocio

PTaaS y SCR	Cadena de suministro - SCA

Los agentes de IA construyen.
Los humanos lideran el desarrollo.



Resumiendo hasta aquí

Aumenta la velocidad de desarrollo, la **cantidad de código** y el **tamaño de la superficie de ataque**, por lo que aparecen más vulnerabilidades por remediar.

Surgen **nuevos objetivos de evaluación, entornos, interfaces y superficies de ataque,** generando nuevos riesgos, casos de abuso y técnicas de prueba.

La cadena de suministro de *software* es crítica y **será aún más crítica.**

Los desarrolladores pasarán a trabajar más como **validadores** que como escritores de *software*.

La **IA** cambia **cómo** se construye el *software*, no **quién** es **responsable** de asegurarlo.

Resumiendo hasta aquí

Las aplicaciones siempre se ejecutan en un ambiente. **El código puede ser seguro y el ambiente no.**

¿Es suficiente con leer el código? Un código con **sintaxis perfecta** puede ser **vulnerable**.

Hay vulnerabilidades que no se pueden detectar desde el código; es necesario **revisarlas en ejecución.**

Los **falsos negativos** no pierden su relevancia; siempre son **vectores de ataque abiertos**.

Desarrollar es barato y automatizado, y los **falsos positivos** se resuelven fácil y rápido. La priorización pierde relevancia gracias a la **economía** en la remediación.

¿Qué estamos protegiendo?

Lo que ya conocíamos

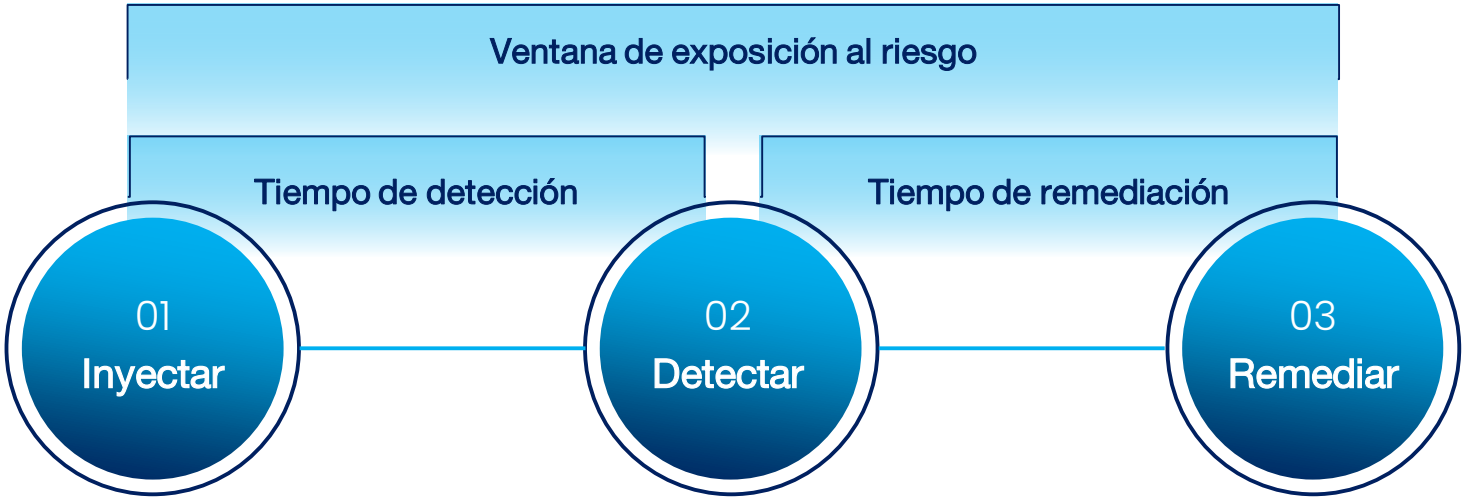
- Código propio
- Dependencias *open-source*
- Infraestructura y contenedores
- APIs

+

Lo que se suma

- LLMs como actores en la cadena de valor
- Agentes con acceso a herramientas
- MCP *servers* como eslabones de interconexión
- Interfaces de chat y voz
- Secretos compartidos entre más sistemas

Objetivo: Desarrollar y mantener *software* seguro



La ventana de exposición al riesgo debe ser siempre menor al tiempo de explotación de la vulnerabilidad

Tener presente que los atacantes también mejoran sus estrategias con las herramientas de IA

Nuestra aproximación es

Explotar lo mejor de todos los mundos

Automatización

Ventajas
- Velocidad
- Determinismo
- Economía

Desventajas
- Exactitud
- Contexto

IA

Ventajas
- Razonamiento en lugar de reglas
- Velocidad
- Escalabilidad

Desventajas
- Probabilismo
- Costos
- Exactitud

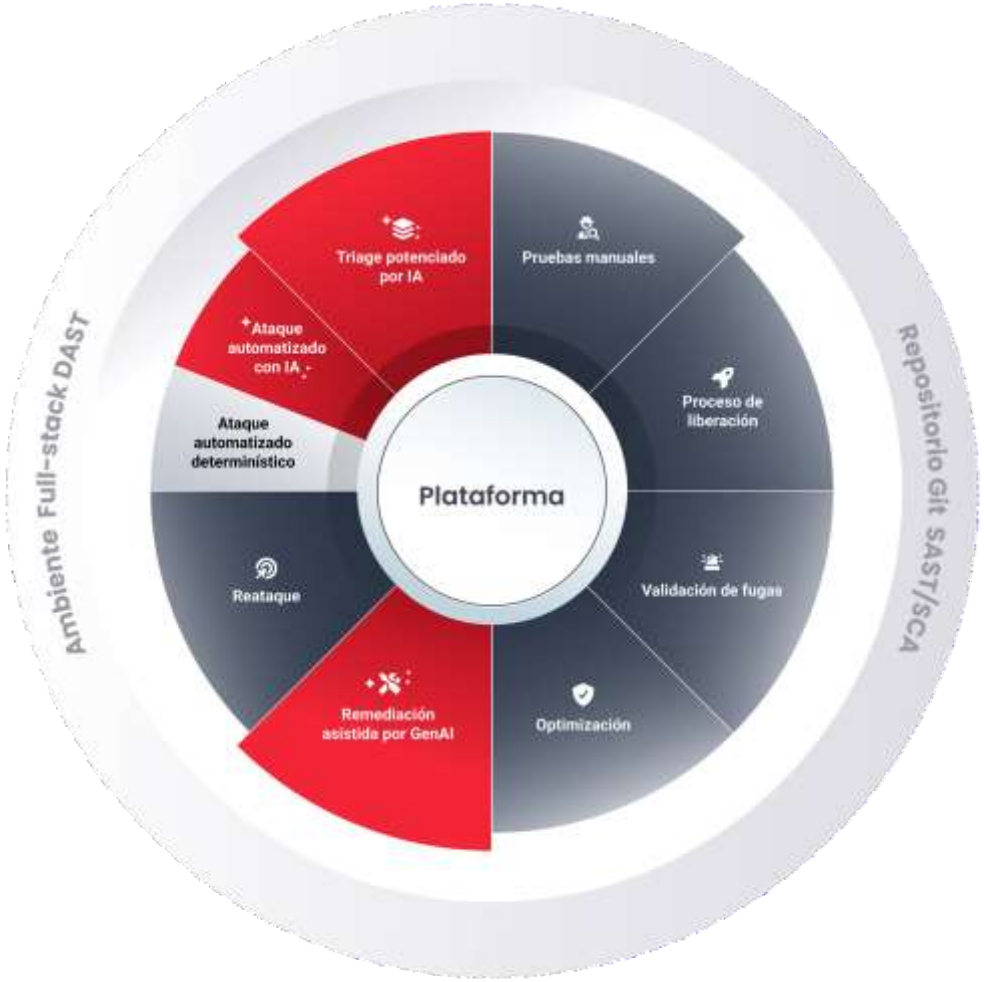
Pentester

Ventajas
- Exactitud
- Contexto de negocio

Desventajas
- Velocidad

Cómo trabajamos

Nuestro ciclo de Hacking Continuo



IA Escaneo AppSec *Pentesters*

¿Dónde ocurre el *testing*?

CLI	PR	CI	Repositorio	CI	Runtime
En tiempo de desarrollo	En revisión de código	Previo a la integración	Escaneo continuo	Antes de pasar a producción	En producción
Velocidad	**Colaboración**	**Control**	**Continuidad**	**Control**	**Realidad**
Feedback inmediato durante el desarrollo	Revisión de lo codificado por el agente	Punto de control crítico antes de la integración	Monitoreo con pruebas integrales estáticas y dinámicas	Punto de control crítico antes de producción	Pruebas dinámicas en condiciones reales de uso

IA aplicada a la realidad en AppSec

PREVENCIÓN DESDE EL CLI

Peer Reviewer Assistant

Detecta patrones de posibles vulnerabilidades para que el desarrollador las valide en la revisión por pares.

REMEDIACIÓN

AI Fixes

Asiste en la remediación de vulnerabilidades de código fuente y dependencias de terceros.

MCP

Este AI Agent permite hacer consultas en la plataforma a través de lenguaje natural.

DETECCIÓN

AI Scanner

Reporta vulnerabilidades de código fuente basándose en patrones conocidos y análisis contextual.

ML Guides

Ordena archivos en repositorios Git según las probabilidades de que contengan vulnerabilidades.

Design Map

Relaciona diseños de seguridad de las empresas con las vulnerabilidades identificadas.

Una visión precisa de los riesgos no se alcanza probando el código, la infraestructura el ambiente por separado

CONCLUSIONES

1

El *workflow* ha mutado para siempre

La IA no es un accesorio, es el nuevo motor del desarrollo. Ahora los **desarrolladores** son más **curadores**, pasando a diseñar y validar **lo que el agente produce** de forma autónoma.

CONCLUSIONES

2

Arquitecturas nuevas, riesgos nuevos

Los componentes como MCP *servers* y agentes LLM no sólo **amplían la superficie de ataque**; crean vectores cualitativamente diferentes que las herramientas tradicionales **no ven**.

CONCLUSIONES

3

Velocidad no es igual a seguridad

La **velocidad de producción** de código ha incrementado con la IA. El volumen total de vulnerabilidades está aumentando, y, aunque la seguridad está mejorando, la velocidad de entrega sólo es valiosa si la **remediación** es igual de **ágil**.

CONCLUSIONES

4

El análisis estático no se asume como un todo

Leer código no basta. El comportamiento en *runtime* y las interacciones entre capas son los verdaderos campos de batalla. Si no pruebas el **sistema en ejecución**, estás parcialmente ciego.

CONCLUSIONES

5

Falsos negativos: el riesgo silencioso

En la era de la inteligencia artificial, los **falsos positivos** son **baratos** de procesar, pero un solo **falso negativo** puede seguir siendo una puerta abierta para un **desastre catastrófico**.

CONCLUSIONES

6

La remediación resulta más económica

Falsos positivos y vulnerabilidades reales se resuelven ahora de forma más fácil y rápida con la ayuda de la IA, por lo que **la priorización de problemas pierde relevancia**.

CONCLUSIONES

7

Pruebas integrales: el camino más indicado

La seguridad no ocurre en silos. La **integración** de **IA** y **escáneres determinísticos** para la velocidad y la escala, y los **expertos humanos** para la profundidad, garantiza una cobertura real en ecosistemas distribuidos.

VI CONGRESO INTERNACIONAL DE CIBERSEGURIDAD, PREVENCIÓN DE FRAUDES Y SEGURIDAD FÍSICA

La IA construye.
El ser humano lidera.
Nosotros aseguramos.

VI CONGRESO INTERNACIONAL DE CIBERSEGURIDAD, PREVENCIÓN DE FRAUDES Y SEGURIDAD FÍSICA

fluidattacks

https://go.fluidattacks.tech/congreso

Web
https://fluidattacks.com/

Email
info@fluidattacks.com

Ubicación
95 3rd Street, 2nd Floor
San Francisco, CA 94103

Teléfono
+1 415 404 2154
+57 314 2597110

Cláusula legal

CREST MEMBER

AICPA SOC

SOC 2 Type II
SOC 3