# Técnicas y Herramientas para la Detección de Vulnerabilidades en Sistemas de Información.

I.C. Ana Laura Hernández Saucedo

CIMAT

# Agenda


- Introducción
- Revisión sistemática
- Ataques basados en vulnerabilidades
- Herramientas para la detección de vulnerabilidades
- Técnicas para la detección de vulnerabilidades
- Propuesta de utilización de Herramientas
- Conclusiones

# Introducción

CIMAT

- En la actualidad el **nivel de complejidad** de las Tecnologías de la Información y Comunicación **TICs ha aumentado**, agregando un mayor riesgo para los sistemas informáticos, teniendo como consecuencia el **aumento en el número de ataques** aprovechando las vulnerabilidades o fallos de seguridad [17]. Dentro de los principales ataques basados en vulnerabilidades se encuentran los **de inyección de SQL**, secuencia de comandos en sitios cruzados, falsificación de petición en sitios cruzados (CSRF) [23], entre otros.

- Una manera de **evitar los ataques** informáticos **es la prevención**. Existen varios enfoques para la detección de vulnerabilidades, algunos de ellos son Black-box y White-box [4]. Existen más enfoques como es el análisis estático y dinámico [4], de ellos existen más técnicas como passive testing[15], fuzz testing[22], penetration testing [14], entre otros.

- En este trabajo se presenta una propuesta de técnicas y herramientas para la detección de vulnerabilidades actuales en sistemas de información. Además de permitirnos conocer el estado actual en esta área.

# Revisión Sistemática

CIMAT

- Una revisión sistemática es un **método** que permite a los especialistas **obtener resultados relevantes** y cuantificados. Esto puede llevar a la identificación, selección y presentación de pruebas en relación con la investigación en un tema en particular.

# Revisión Sistemática

CIMAT


| Vulnerabilidades | Técnicas | Herramientas |
|---|---|---|
| Inyección | Black-box | QualysGuard Web Application Scanning WAS |
| Perdida de Autenticación y Gestión de sesiones | White-box | McAfee Vulnerability Manager |
| Secuencia de Comandos en Sitios Cruzados | Análisis estático de código (auditoria de código fuente) | Nessus Vulnerability Scanner |
| Referencia Directa Insegura a Objetos | Análisis dinámico de código | Web Site Security Audit - WSSA |
| Configuración de Seguridad Incorrecta | Pruebas de penetración | Retina Web Security Scanner |
| Exposición de datos sensibles | Pruebas pasivas | WEBAPP360: ENTERPRISE CLASS WEB APPLICATION SCANNING |
| Ausencia de Control de Acceso a Funciones | Pruebas activas | WhatWeb |
| Falsificación de Petición en Sitios Cruzados (CSRF) | Fuzz testing (pruebas de caja negra) | Frame-C |
| Utilización de componentes con vulnerabilidades conocidas | | Parasoft C/C++Test |
| Redirecciones y reenvíos no validados | | SCA |
| | | ITS4 |
| | | RATS |

# Ataques basados en vulnerabilidades

CIMAT

- El proyecto abierto de seguridad en aplicaciones web (**OWASP** por sus siglas en inglés) emite el **top 10 de las vulnerabilidades** más graves de **aplicaciones web**. El objetivo principal es educar a las organizaciones y desarrolladores sobre las consecuencias de las vulnerabilidades de seguridad en aplicaciones web más importantes.

# Ataques basados en vulnerabilidades

CIMAT

- *Inyección*: Las fallas de inyección, ocurren cuando **datos no confidenciales** son enviados a un **interprete** como parte de un comando o consulta, tratando de **engañar** al intérprete en ejecutar comandos no intencionados o **acceder datos no autorizados**.

- *Secuencia de Comandos en Sitios Cruzados:* Las fallas de XSS ocurren cuando una aplicación toma **información** originada por **un usuario** y la envía a un navegador Web **sin** primero **validarl**a o codificando el contenido.

- *Configuración de Seguridad Incorrecta:* Una buena seguridad requiere tener definidas e implementada una **configuración segura para la aplicación**, marcos de trabajo, servidores de aplicación, servidores web, base de datos, y plataformas. Todas estas configuraciones deben ser definidas, implementadas, y mantenidas ya que **por lo general no son seguras por defecto**.

# Ataques basados en vulnerabilidades

CIMAT

- *Exposición de datos sensibles:* Muchas aplicaciones web **no protegen** adecuadamente **datos sensibles** tales como números de tarjetas de crédito, o credenciales de autenticación. Los datos sensibles **requieren de métodos de protección** adicionales tales como el cifrado de datos, así como también de precauciones especiales en un intercambio de datos con el navegador.

- *Falsificación de Petición en Sitios Cruzados (CSRF):* Un ataque CSRF obliga al navegador de una víctima autenticada a enviar una **petición HTTP falsificado**, incluyendo la sesión del usuario y cualquier otra información de autenticación incluida automáticamente, a una aplicación web vulnerable.

# Herramientas para la detección de vulnerabilidades

CIMAT

VULNERABILITY MANAGEMENT
TRIPWIRE IP360

tripwire

beyondtrust
Beyond Traditional Security
Retina Network Security Scanner

QUALYS
CONTINUOUS SECURITY

WAS Web Application Scanning

BEYOND security
Web Site Security Audit - WSSA by Beyond Security

WhatWeb

McAfee
Vulnerability Manager

Nessus
vulnerability scanner

ITS4: SOFTWARE SECURITY TOOL

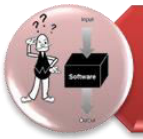
PARASOFT.
C/C++test

frama C
Software Analyzers

# Técnicas para la detección de vulnerabilidades

CIMAT

**De acuerdo a los resultados obtenidos de la revisión sistemática realizada sobre las herramientas y técnicas utilizadas para detección de vulnerabilidades, se puede establecer las técnicas utilizadas.**




- Black-box

- Pruebas de penetración


- White-box

- Pruebas pasivas

- Análisis estático de código (auditoria de código fuente)

- Pruebas activas

- Análisis dinámico de código


- Fuzz testing (pruebas de caja negra)

# Técnicas para la detección de vulnerabilidades

CIMAT

- **Black-box:** Es una técnica basada para descubrir vulnerabilidades en aplicaciones web, probando la aplicación desde el punto de vista del atacante.

- **White-box:** Está del lado del servidor. En este tipo de enfoque se tiene acceso a información relevante de la organización.

- **Análisis estático de código (auditoria de código fuente):** Es un método en el que no se requiere ejecutar el programa, este realiza un análisis de código fuente directo para determinar huecos en la seguridad.

- **Análisis dinámico de código:** Se comunica con la aplicación web a través de front-end de la aplicación en orden de identificar vulnerabilidades de seguridad potenciales y debilidades en la arquitectura de la aplicación web.

# Técnicas para la detección de vulnerabilidades

CIMAT

- **Pruebas de penetración:** Consiste en la simulación de un ataque de los maliciosos outsiders (que no tienen un medio autorizado de acceder a los sistemas de la organización) y de maliciosos insiders (que tienen algún nivel de acceso autorizado).

- **Pruebas pasivas:** Las pruebas pasivas están diseñadas para el análisis del tráfico de telecomunicaciones. Permite detectar fallas y defectos de seguridad mediante el examen de los paquetes capturados (livetrafficor log files).

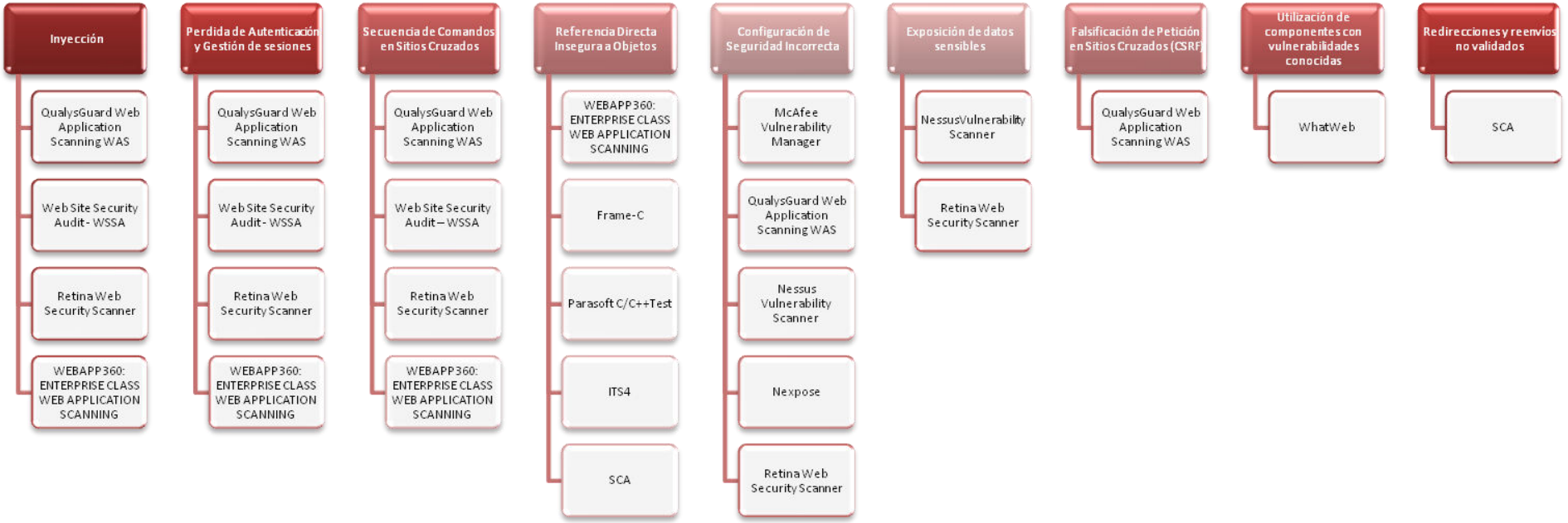
- **Pruebas activas:** Utiliza un programador de subprocesos asignados al azar para verificar si las advertencias comunicadas por un análisis predictivo de programa son errores reales (simulaciones).

- **Fuzz testing (pruebas de caja negra):** Consiste en **estimular el sistema** bajo prueba, utilizando **datos aleatorios o mutados** queridos, con el fin de detectar comportamientos no deseados como violación de confidencialidad.

# Propuesta de utilización de herramientas

CIMAT

En base a los resultados obtenidos de la revisión sistemática, se propone la utilización de herramientas para los ataques basados en vulnerabilidades más comunes.



# Conclusiones

CIMAT

Al realizar el análisis de las herramientas y técnicas utilizadas para la detección de vulnerabilidades, se concluye que existen muchas herramientas que proporcionan la detección para diferentes propósitos, es decir, algunas herramientas cubren desde escaneo de vulnerabilidades en aplicaciones web, hasta escaneo de vulnerabilidades en dispositivos móviles, un ejemplo de este tipo de herramientas es Nessus Vulnerability Scanner, además de muchas otras funcionalidades.

De igual manera existen herramientas muy específicas para la detección de problemas de seguridad muy específica, como por ejemplo WhatWeb [27] en el que solamente se enfoca en el escaneo de sitios web.