Ciberseguridad Web con OWASP
Manual Técnico

Autor: JOAO DANIEL DELGADO TITO

Dirigido a: Estudiantes y Técnicos en Ciberseguridad

Edición: 2025

Índice

1. ¿Qué es OWASP?

OWASP (Open Web Application Security Project) es una comunidad global sin fines de lucro dedicada a mejorar la seguridad del software. Su misión es hacer que la seguridad en aplicaciones web sea visible, para que individuos y organizaciones puedan tomar decisiones informadas.

2. Importancia del OWASP Top 10

El OWASP Top 10 es un documento de referencia que enumera las 10 principales vulnerabilidades de seguridad en aplicaciones web. Es ampliamente utilizado por desarrolladores, testers, ingenieros y analistas como guía para evaluar y mejorar la ciberseguridad de los sistemas.

3. OWASP Top 10 Vulnerabilidades Explicadas

A01: Broken Access Control

Los usuarios pueden acceder a recursos sin la debida autorización. Ejemplo: Cambiar el ID de usuario en la URL para acceder a otra cuenta.

A02: Cryptographic Failures

Fallos en la protección de datos sensibles. Ejemplo: Transmisión de contraseñas sin cifrado HTTPS.

A03: Injection
Inserción de código malicioso a través de entradas no validadas. Ejemplo: SQL Injection.

A04: Insecure Design
Diseños arquitectónicos inseguros desde el inicio. Ejemplo: Falta de límites de control de acceso en APIs.

A05: Security Misconfiguration
Configuraciones incorrectas o por defecto. Ejemplo: Servidores con páginas de administración públicas.

A06: Vulnerable and Outdated Components
Uso de librerías o frameworks desactualizados. Ejemplo: Versiones antiguas de jQuery o Apache Struts.

A07: Identification and Authentication Failures
Errores en autenticación o gestión de sesiones. Ejemplo: Uso de contraseñas débiles sin 2FA.

A08: Software and Data Integrity Failures
Falta de validación de integridad en actualizaciones. Ejemplo: Actualización de software desde fuentes no verificadas.

A09: Security Logging and Monitoring Failures
Falta de monitoreo y registros para detectar ataques. Ejemplo: No detectar múltiples intentos fallidos de login.

A10: Server-Side Request Forgery (SSRF)
El servidor accede a recursos internos al ser manipulado por un atacante. Ejemplo: Un formulario web que permite ingresar una URL externa sin validación.

4. Ejemplos Reales y Casos Prácticos
- Equifax (2017): Exploited una vulnerabilidad de Apache Struts → Fuga de datos de 147 millones de personas.
- Twitter (2020): Ataque de ingeniería social + fallas en permisos internos → Acceso a cuentas verificadas.
- GitHub: Ha registrado múltiples ataques de tipo SSRF usados para escanear redes internas desde la nube.

5. Buenas Prácticas y Mitigación
- Aplicar principios de seguridad desde el diseño (Security by Design).
- Usar herramientas de análisis estático (SAST) y dinámico (DAST).
- Mantener componentes y librerías actualizados.
- Validar y sanear todas las entradas de usuarios.
- Configurar adecuadamente los servidores y deshabilitar funciones innecesarias.

- Implementar autenticación multifactor (MFA).
- Monitorear actividades sospechosas y mantener registros.

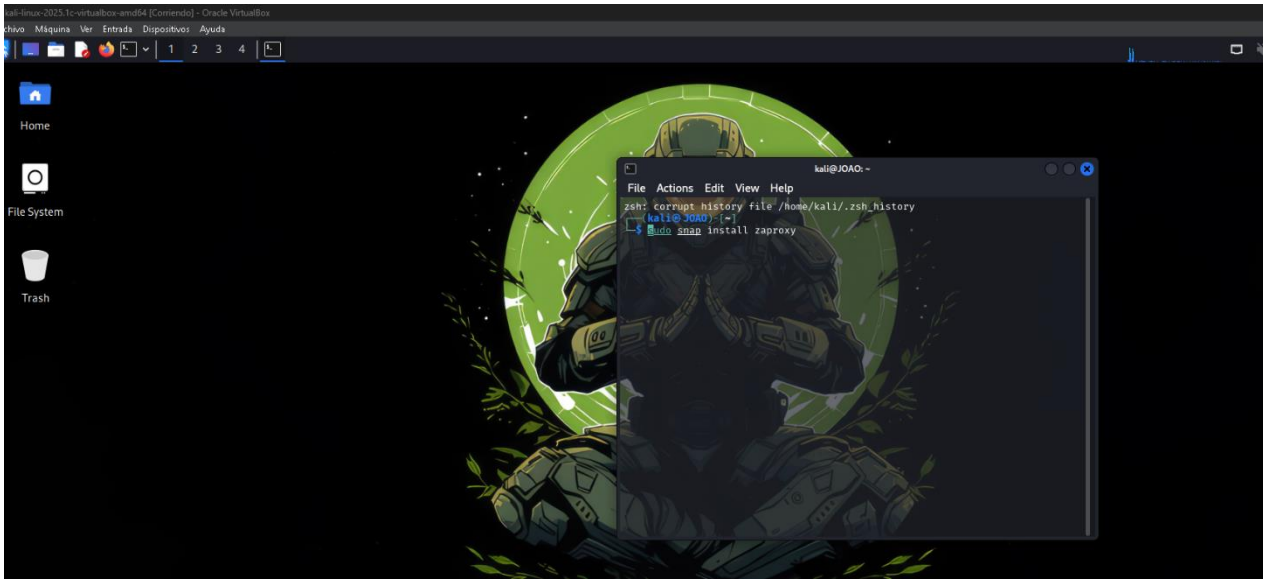
6. Herramientas y Recursos Recomendados

- OWASP ZAP: Escáner gratuito de vulnerabilidades.
- Burp Suite: Plataforma profesional para pruebas de seguridad web.
- SonarQube: Análisis estático de código.
- Security Headers: Verificación de cabeceras de seguridad.
- Have I Been Pwned: Revisión de filtraciones de credenciales.

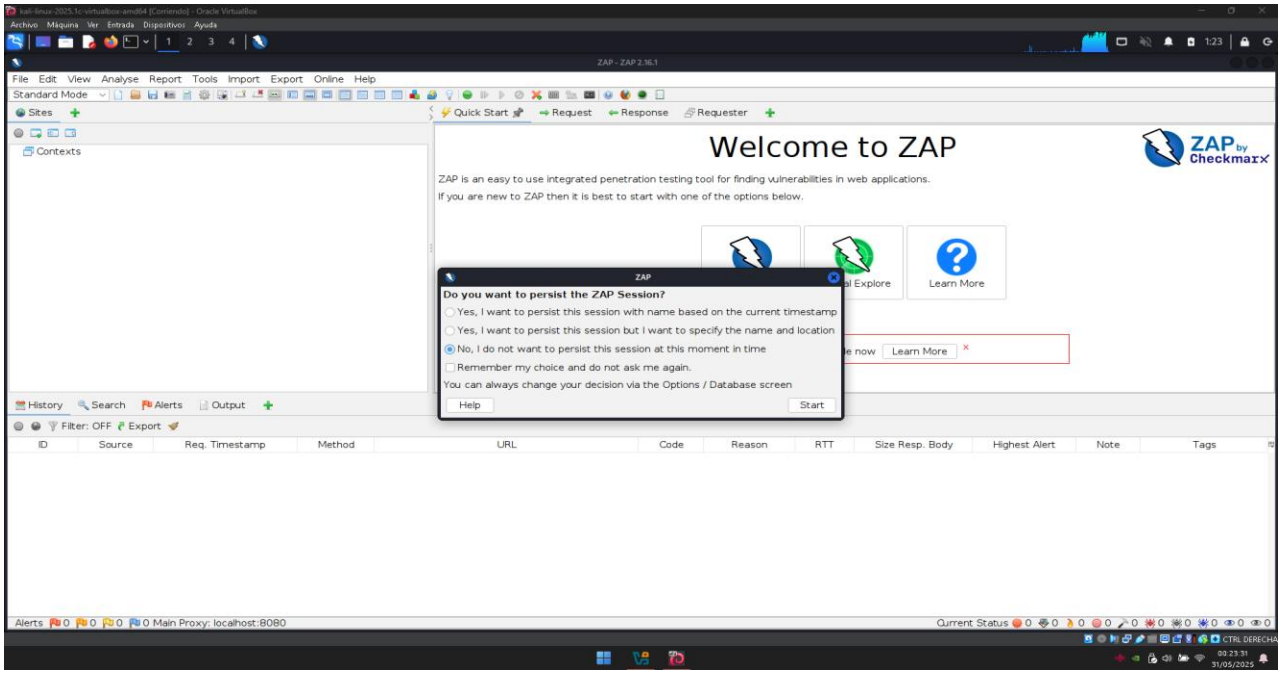
AHORA HAREMOS UNA PRACTICA

Usaremos owasp zap una herramienta en Kali

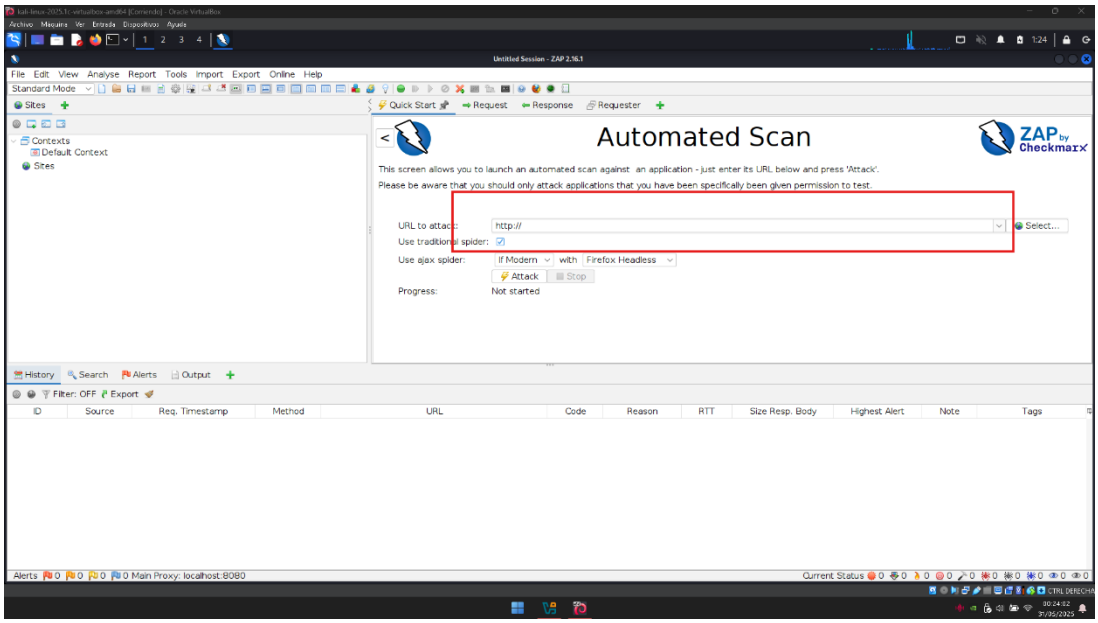
Como paso 1 comenzaremos instalando la aplicación que necesitaremos



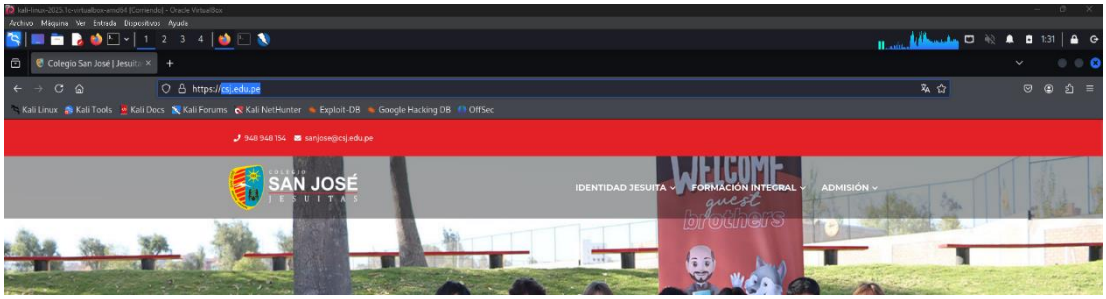
Como paso 2 iniciaremos el programa en esta Kali como zap



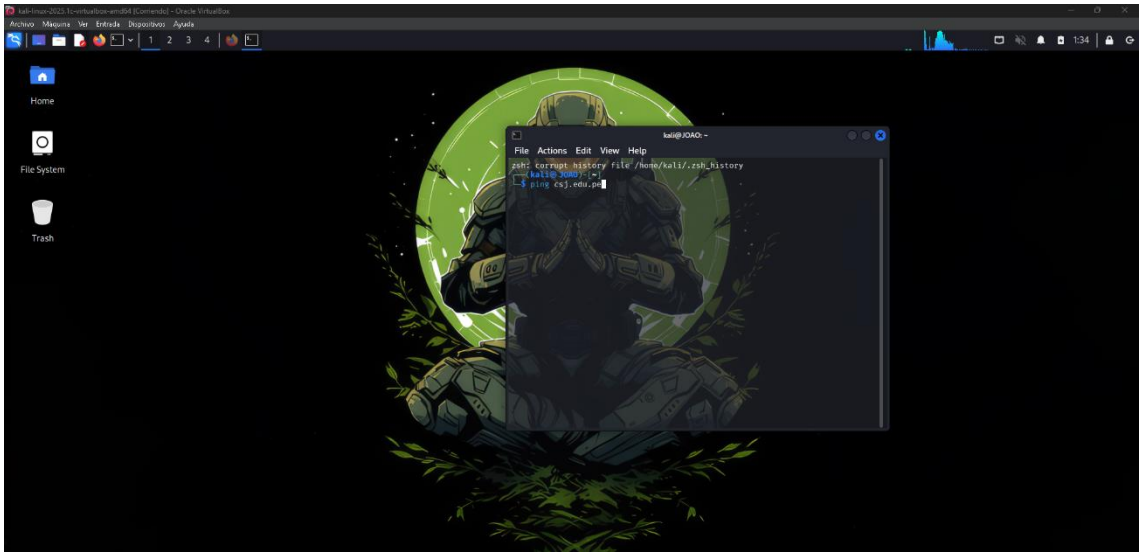
Como paso 3 tenemos que localizar la ip o el dominio que queremos escanear



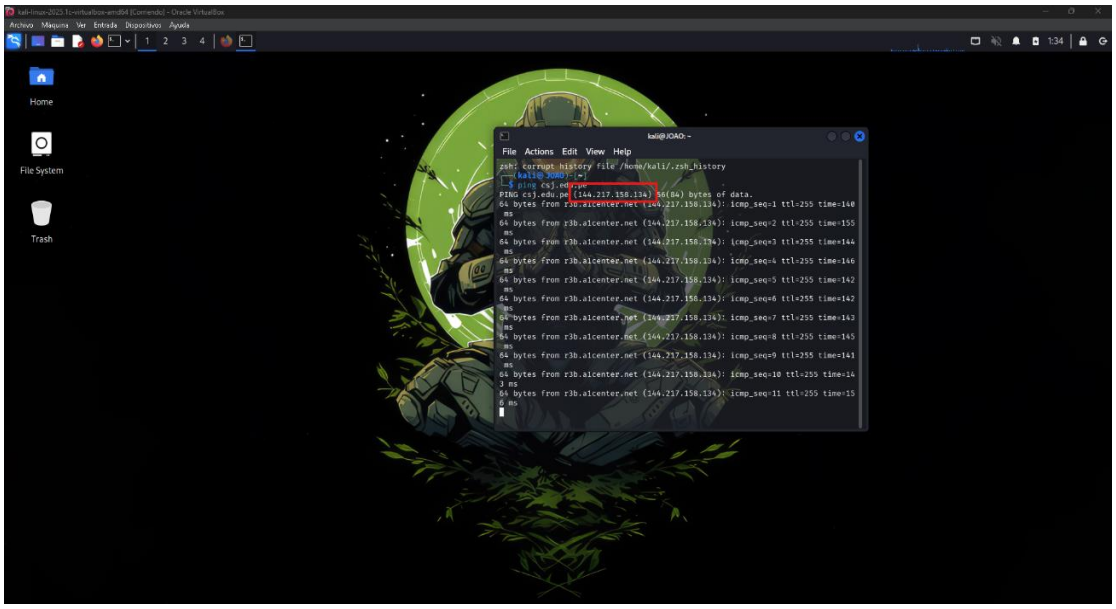
En este caso usaremos una URl para saber la ip que escanearemos



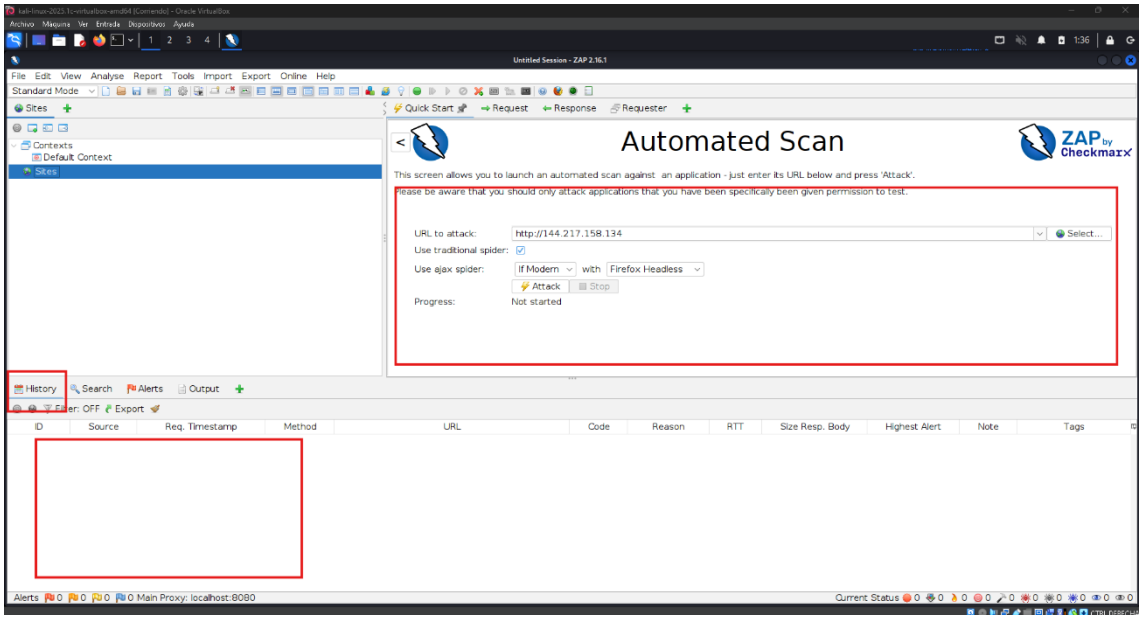
Una vez que sabemos que url usaremos iremos al hacer un ping a la url



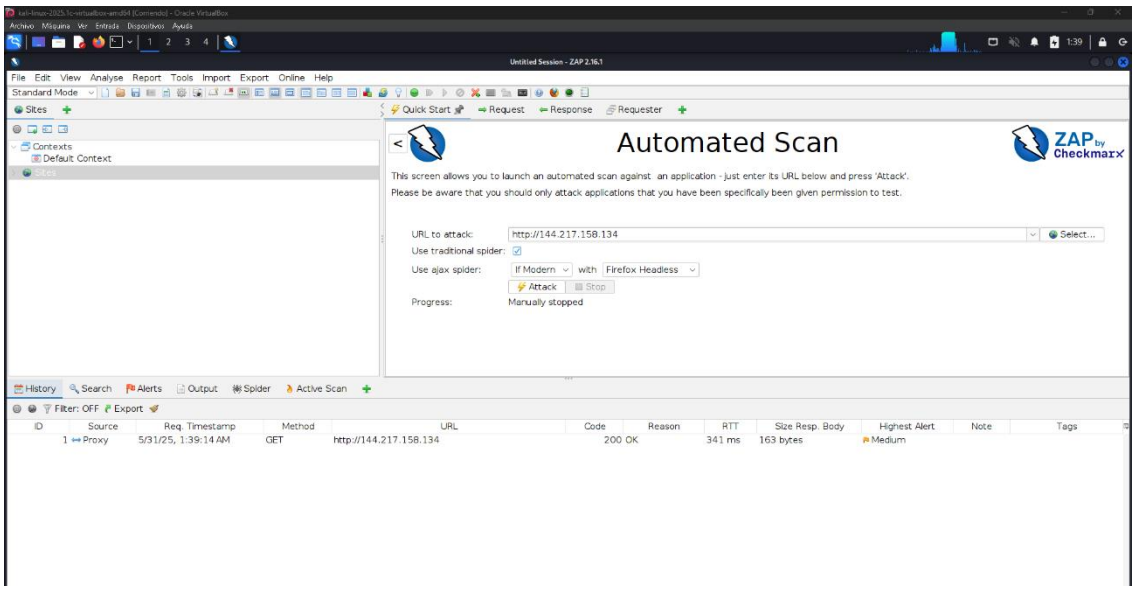
Como vemos ya sabemos que ip es la que usaremos para hacer el escaneo



Luego ya en la plataforma de las aplicaciones procederemos a colocar la ip

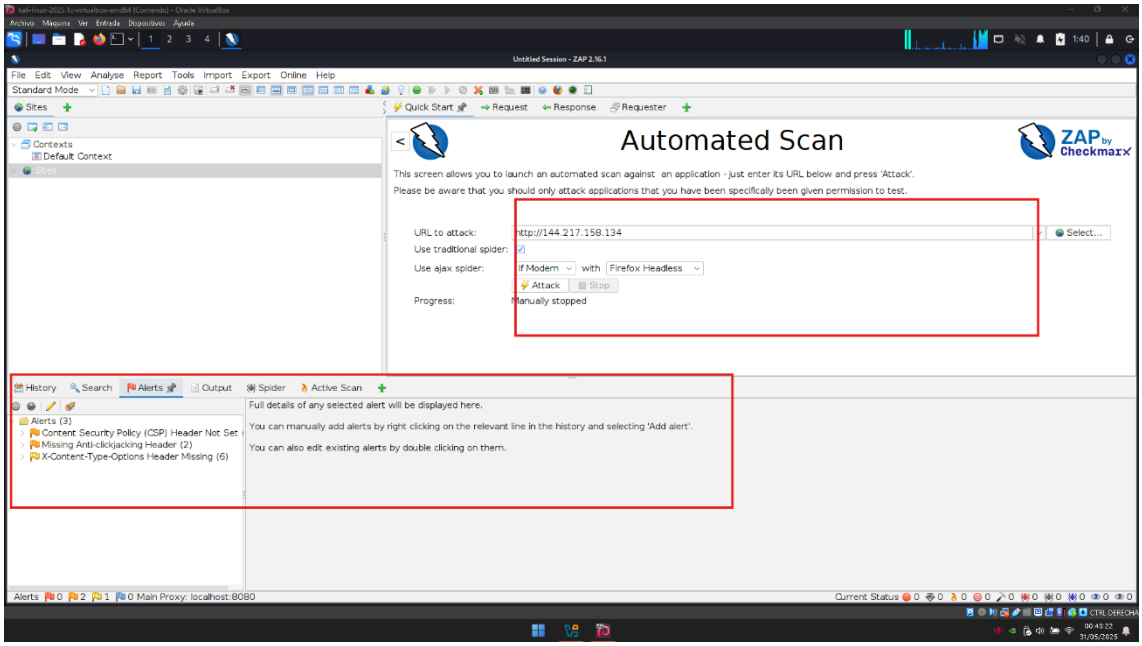


Nos dará como resultado en los campos lo que analizo el programa

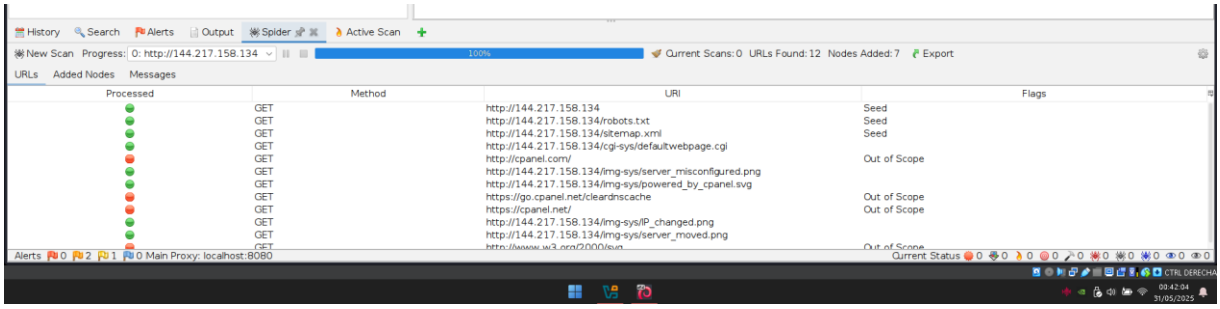


Como podemos ver el programa detecto un proxy más adelante veremos que mas pudimos ver en este escaneo

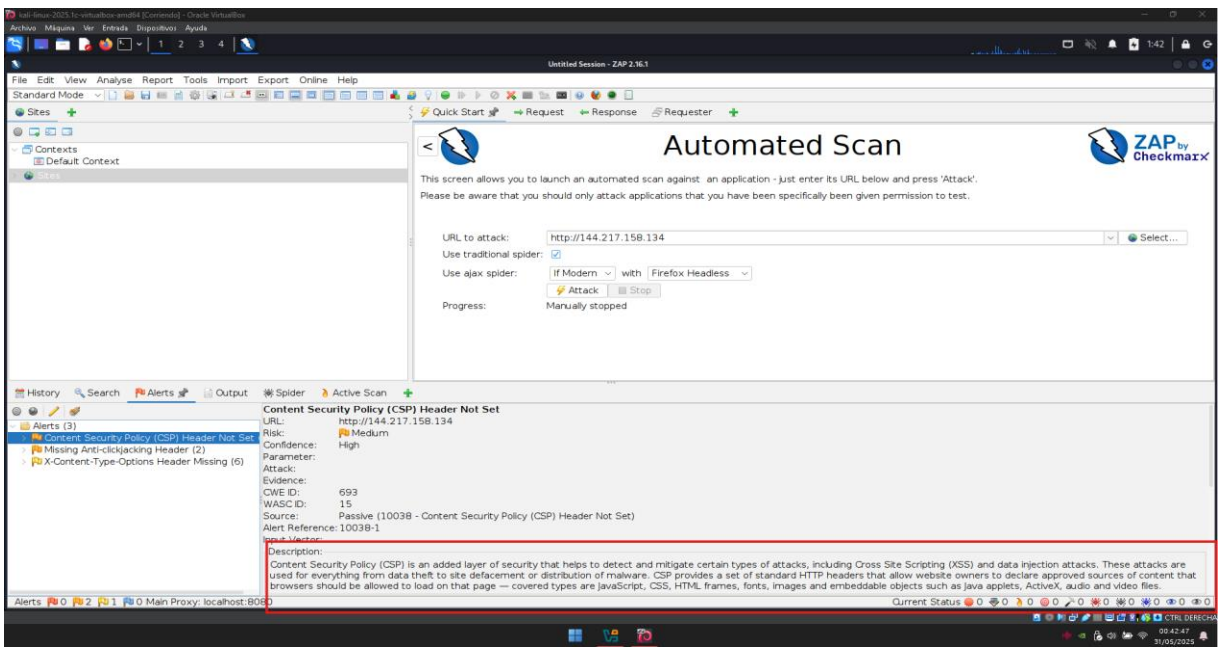
Como vemos hay tres posibles vulnerabilidades 2 son de medio y 1 de bajo nivel



De igual manera podemos ver el spider que nos permite ver los url y los sitios web



Lo más importante es que el programa nos da una descripción de las vulnerabilidades y una posible solución ante ello.



Conclusión sobre OWASP ZAP

OWASP ZAP es una herramienta muy práctica y accesible para quienes trabajan en la seguridad de aplicaciones web. Al ser gratuita y de código abierto, permite a desarrolladores y expertos detectar vulnerabilidades comunes de manera sencilla. Además, ofrece opciones para realizar pruebas más profundas y personalizadas, lo que ayuda a mejorar la protección de las aplicaciones. En definitiva, OWASP ZAP es una opción confiable para fortalecer la seguridad y evitar posibles ataques, aportando tranquilidad en el desarrollo y mantenimiento de software.