Introducción a ZAP

Pablo Gómez - OWASP Logroño - La Rioja

Mayo 2025

Pablo Gómez Sánchez

- Padre de 2
- Co-Fundador @ Redsauce
- Co-Líder de OWASP Logroño - La Rioja
- Desarrollador, QA automation, cybersec...
- En el Ebro, con la familia o los amigos

pgomez@redsauce.net
https://www.linkedin.com/in/pablogomezsanchez/

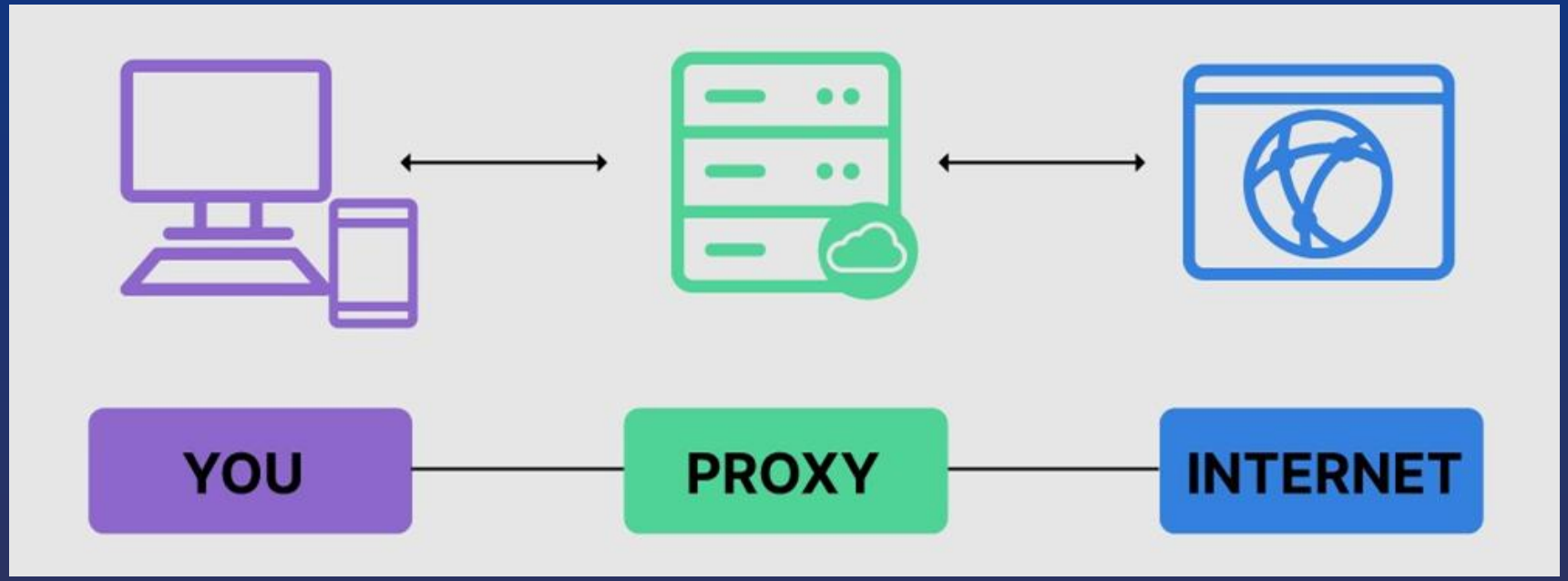
¿Qué es ZAP (Zed Attack Proxy)?

Herramienta Open-Source

Encuentra vulnerabilidades

Análisis web manual y automatizado

Entendiendo el Proxy



Navegador

HTTP/HTTPS

ZAP

Intercepta, analiza y registra

Servidor Web

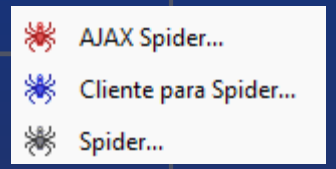
Responde a las peticiones procesadas

Caso de uso 1: mapeo del sitio

Descubrimiento de la estructura

Lanzamiento del Spider

ZAP inicia el rastreo automático del sitio web.



Identificación de enlaces

Descubre URLs, formularios y recursos ocultos.

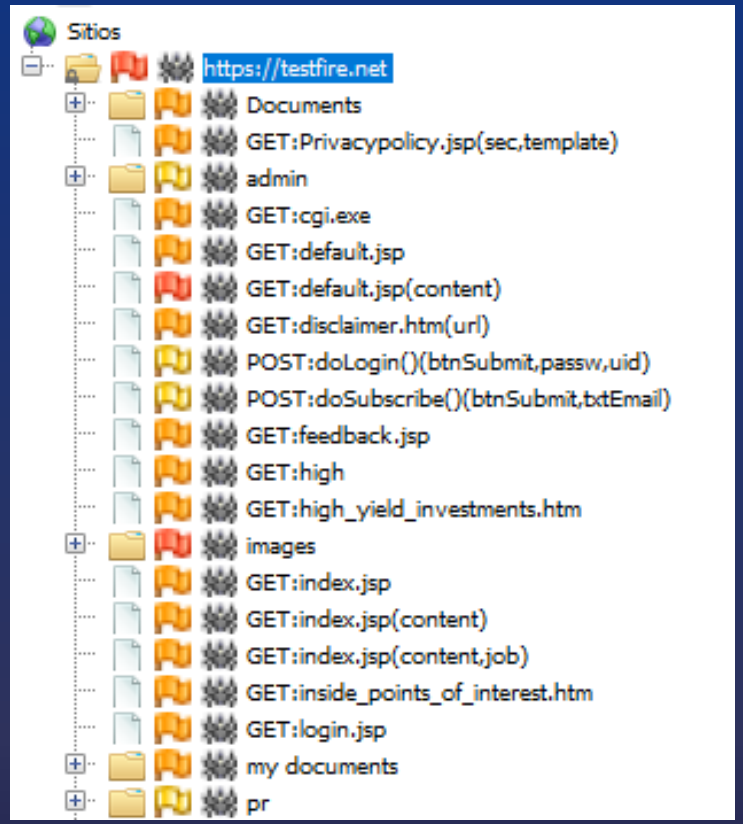
Construcción del árbol

Genera una visualización completa de la estructura web.

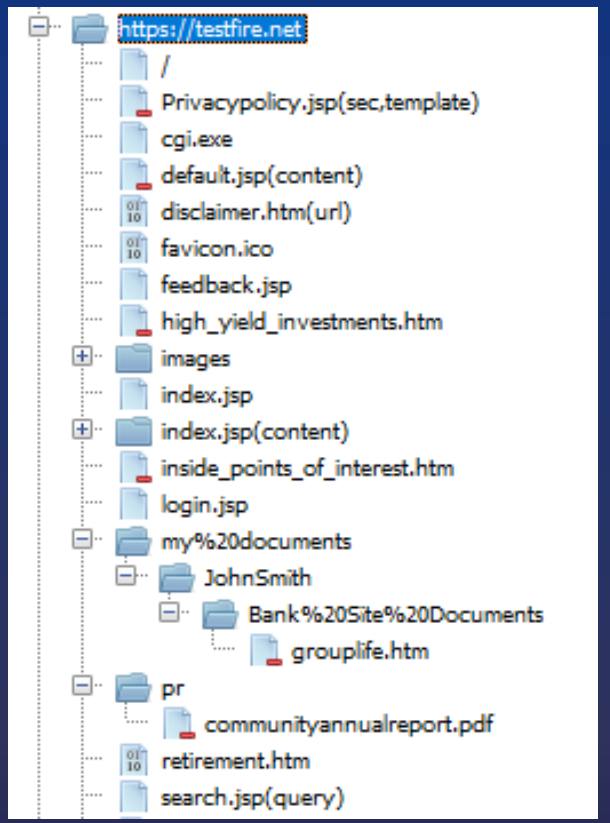
Caso de uso 1: mapeo del sitio

Descubrimiento de la estructura

Spider y AJAX Spider



Client Spider



Caso de uso 2: escaneo activo

Encontrando debilidades ocultas (y sus vulnerabilidades asociadas)

XSS

Cross-Site Scripting

SQL Injection

Manipulación de bases de datos

CSRF

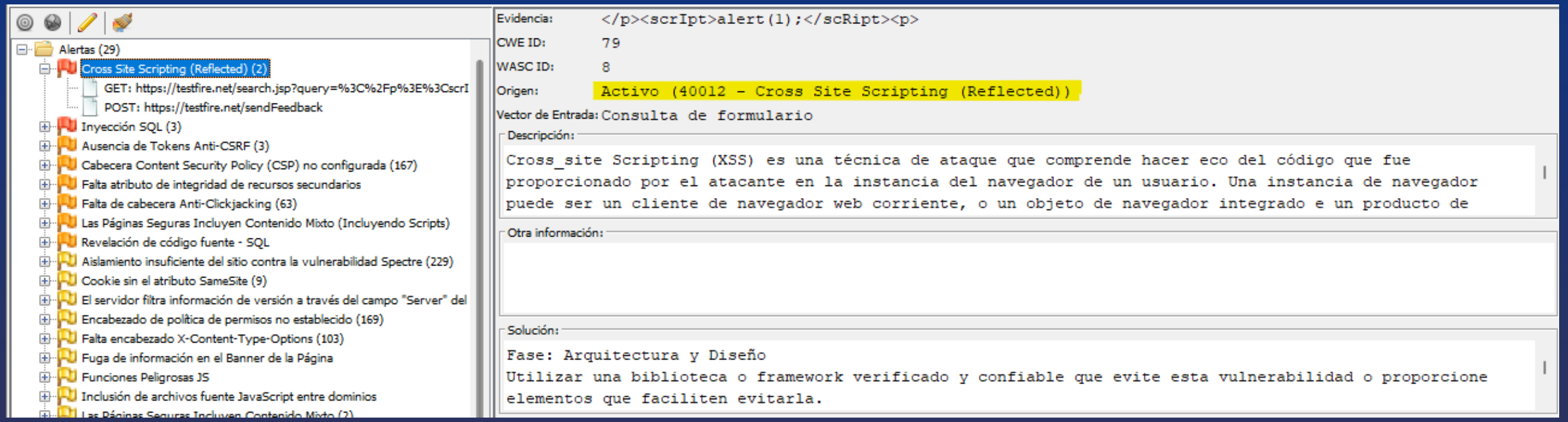
Cross-Site Request Forgery

Path Traversal

Acceso a ficheros del sistema

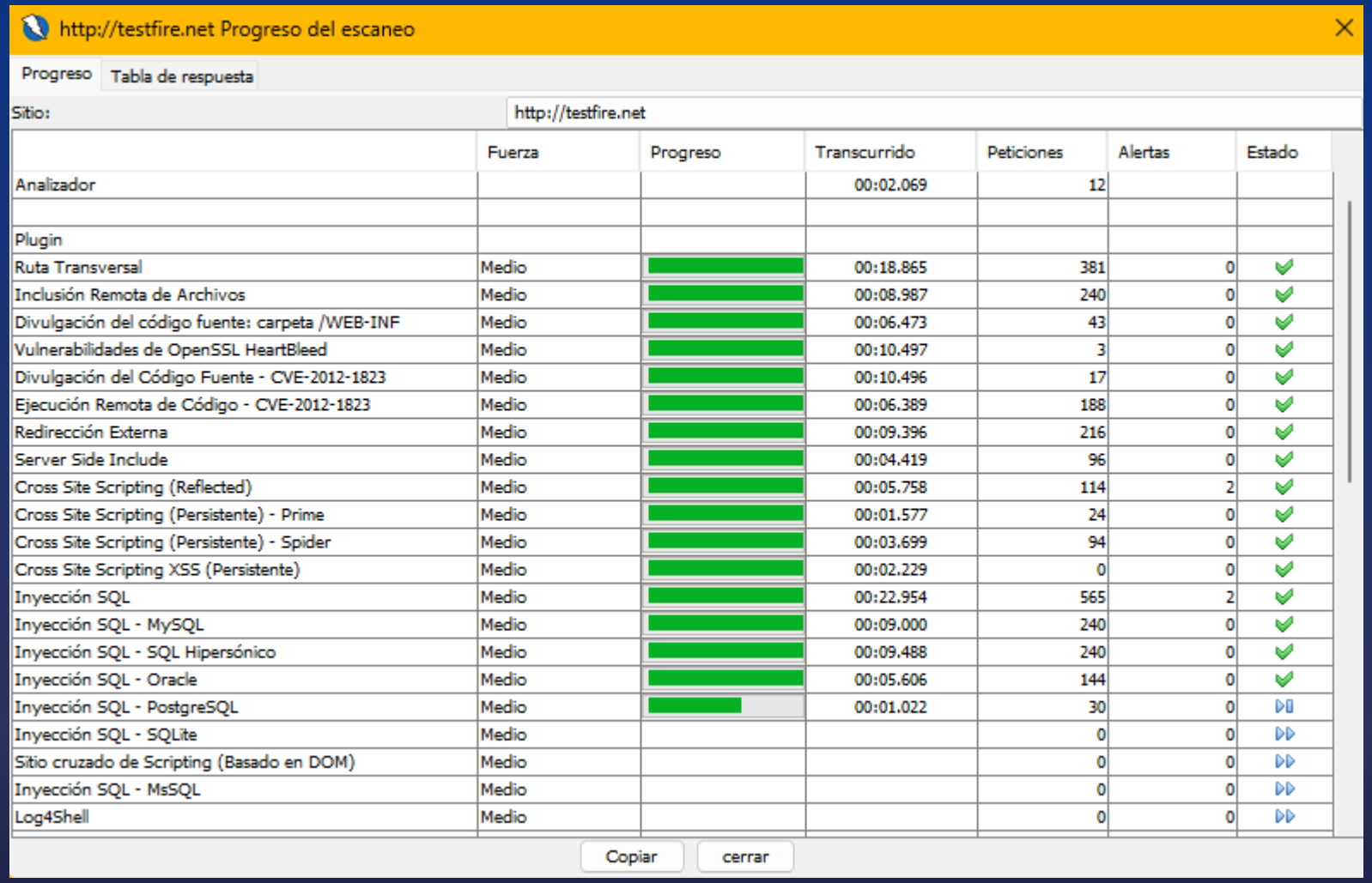
Caso de uso 2: escaneo activo

Encontrando debilidades ocultas (y sus vulnerabilidades asociadas)



Caso de uso 2: escaneo activo

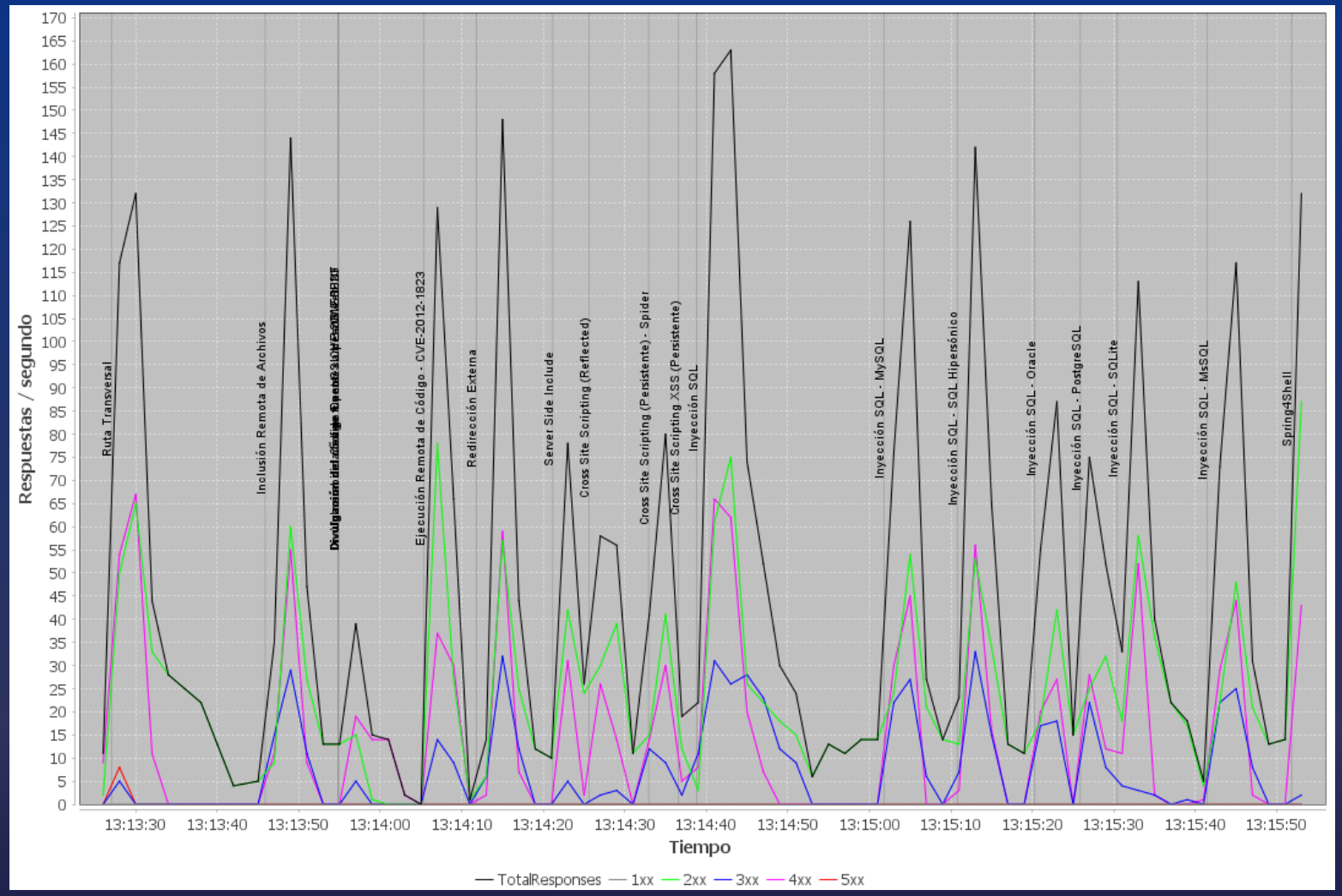
Encontrando debilidades ocultas (y sus vulnerabilidades asociadas)



	Fuerza	Progreso	Transcurrido	Peticiones	Alertas	Estado
Analizador			00:02.069	12		
Plugin						
Ruta Transversal	Medio		00:18.865	381	0	✓
Inclusión Remota de Archivos	Medio		00:08.987	240	0	✓
Divulgación del código fuente: carpeta /WEB-INF	Medio		00:06.473	43	0	✓
Vulnerabilidades de OpenSSL HeartBleed	Medio		00:10.497	3	0	✓
Divulgación del Código Fuente - CVE-2012-1823	Medio		00:10.496	17	0	✓
Ejecución Remota de Código - CVE-2012-1823	Medio		00:06.389	188	0	✓
Redirección Externa	Medio		00:09.396	216	0	✓
Server Side Include	Medio		00:04.419	96	0	✓
Cross Site Scripting (Reflected)	Medio		00:05.758	114	2	✓
Cross Site Scripting (Persistente) - Prime	Medio		00:01.577	24	0	✓
Cross Site Scripting (Persistente) - Spider	Medio		00:03.699	94	0	✓
Cross Site Scripting XSS (Persistente)	Medio		00:02.229	0	0	✓
Inyección SQL	Medio		00:22.954	565	2	✓
Inyección SQL - MySQL	Medio		00:09.000	240	0	✓
Inyección SQL - SQL Hipersónico	Medio		00:09.488	240	0	✓
Inyección SQL - Oracle	Medio		00:05.606	144	0	✓
Inyección SQL - PostgreSQL	Medio		00:01.022	30	0	▷‖
Inyección SQL - SQLite	Medio			0	0	▷▷
Sitio cruzado de Scripting (Basado en DOM)	Medio			0	0	▷▷
Inyección SQL - MsSQL	Medio			0	0	▷▷
Log4Shell	Medio			0	0	▷▷

Caso de uso 2: escaneo activo

Encontrando debilidades ocultas (y sus vulnerabilidades asociadas)



Caso de uso 1 y 2: comparativa

🎯 En resumen:

Función	Spider	Escaneo Activo
¿Qué hace?	Descubre estructura	Busca vulnerabilidades
¿Lanza ataques?	❌ No	✅ Sí
¿Afecta a la app?	✅ Mínimamente	✅ Potencialmente destructivo
¿Cuándo usarlo?	Antes de escanear o manual	Tras mapear, para auditar seguridad

Caso de uso 3: escaneo automatizado

El “Modo fácil”: lanzar, esperar y revisar.

Spider(s) + Escaneo activo

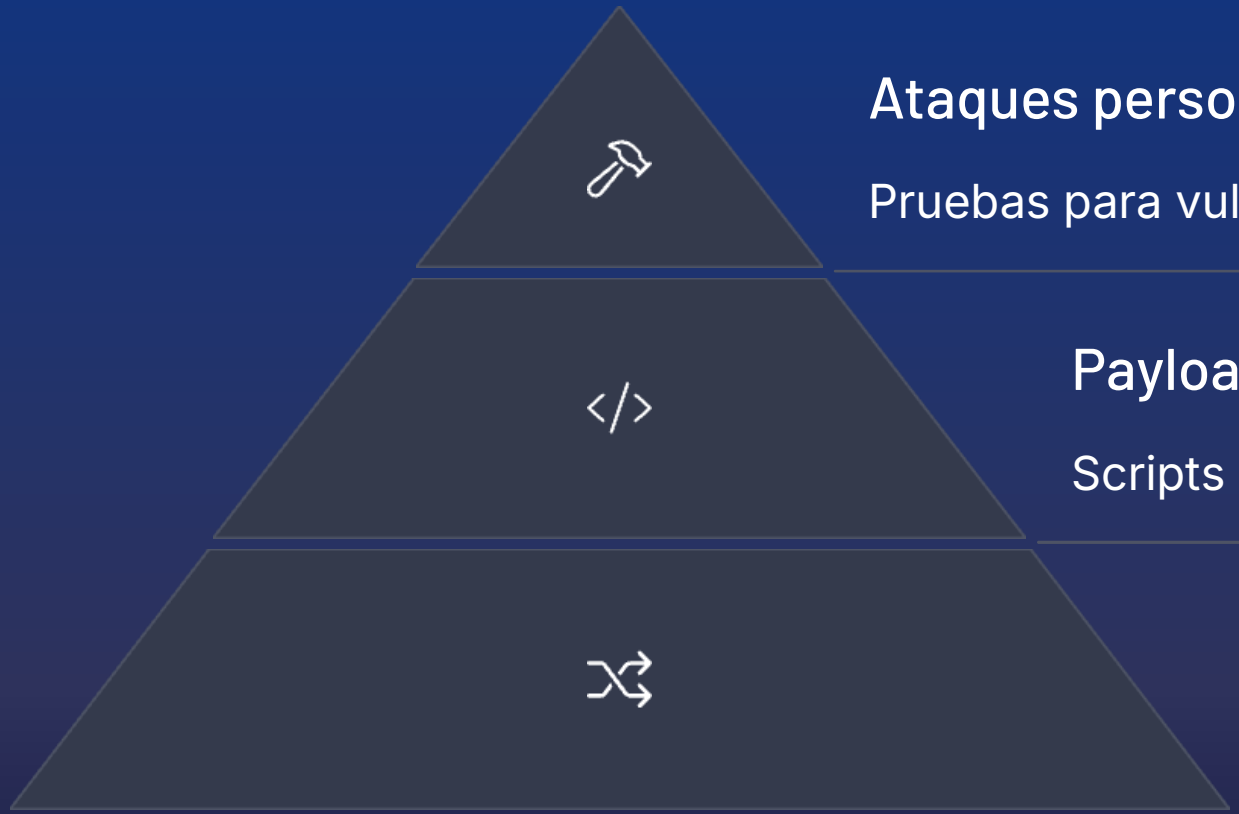
- ❌ No se muestran detalles de la ejecución
- ❌ No es posible pausar ni revisar en tiempo real
- ☑ Sencillo y rápido de usar
- ☑ No requiere configuración ni personalización

Caso de uso 2 y 3: comparativa

Característica	Escaneo Automatizado	Escaneo Activo
¿Diseñado para...?	Simplicidad / principiantes	Control / usuarios técnicos
¿Feedback en tiempo real?	❌ No	✅ Sí
¿Control de parámetros?	❌ Automático	✅ Total
¿Visible en árbol de sitios?	❌ Limitado	✅ Completo
¿Personalización de reglas?	❌ No	✅ Sí

Caso de uso 4: fuzzing

Probando los límites del sistema



Ataques personalizados

Pruebas para vulnerabilidades específicas y complejas

Payloads maliciosos

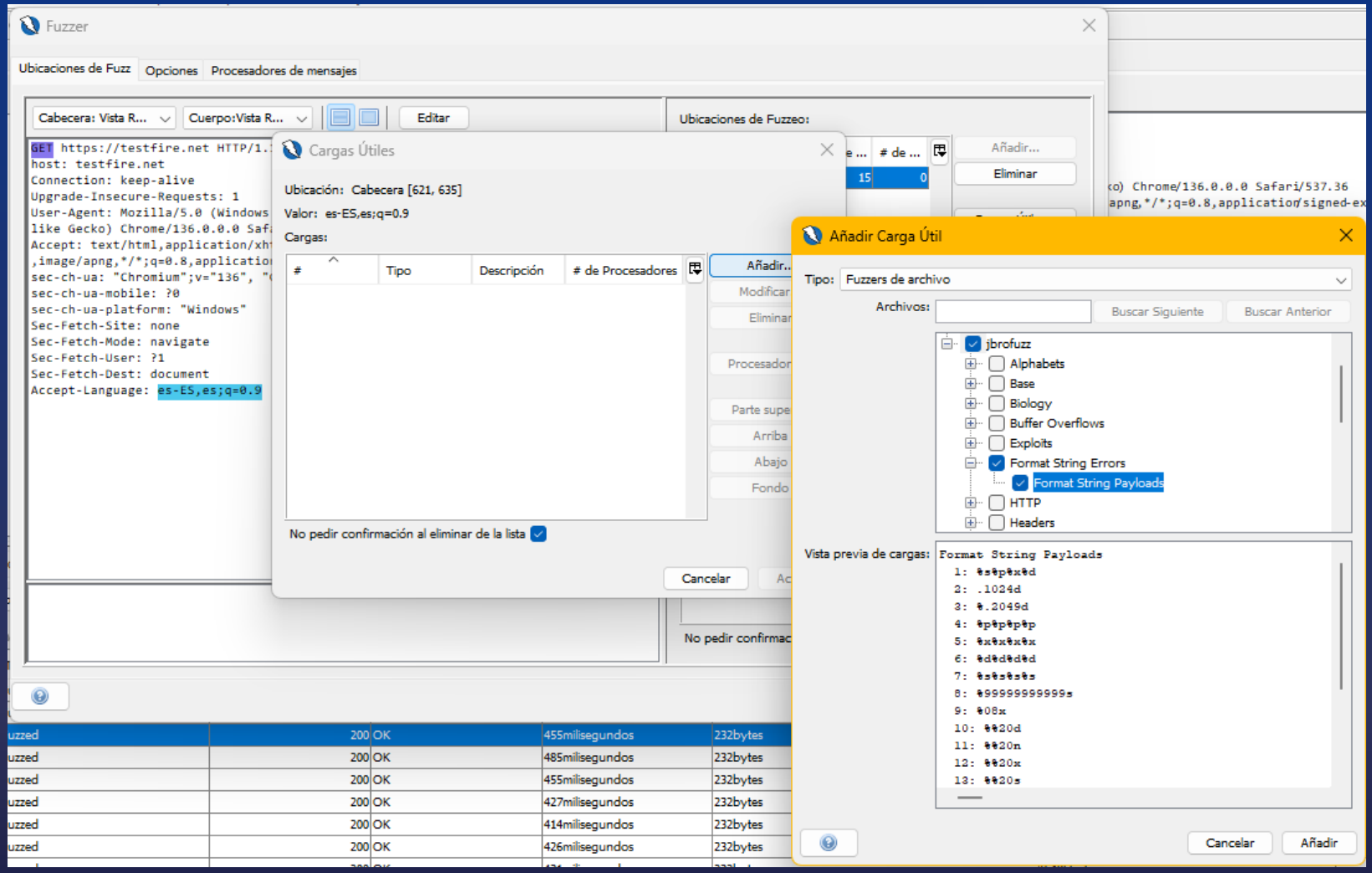
Scripts y comandos potencialmente dañinos

Datos aleatorios

Entradas inesperadas en campos y formularios

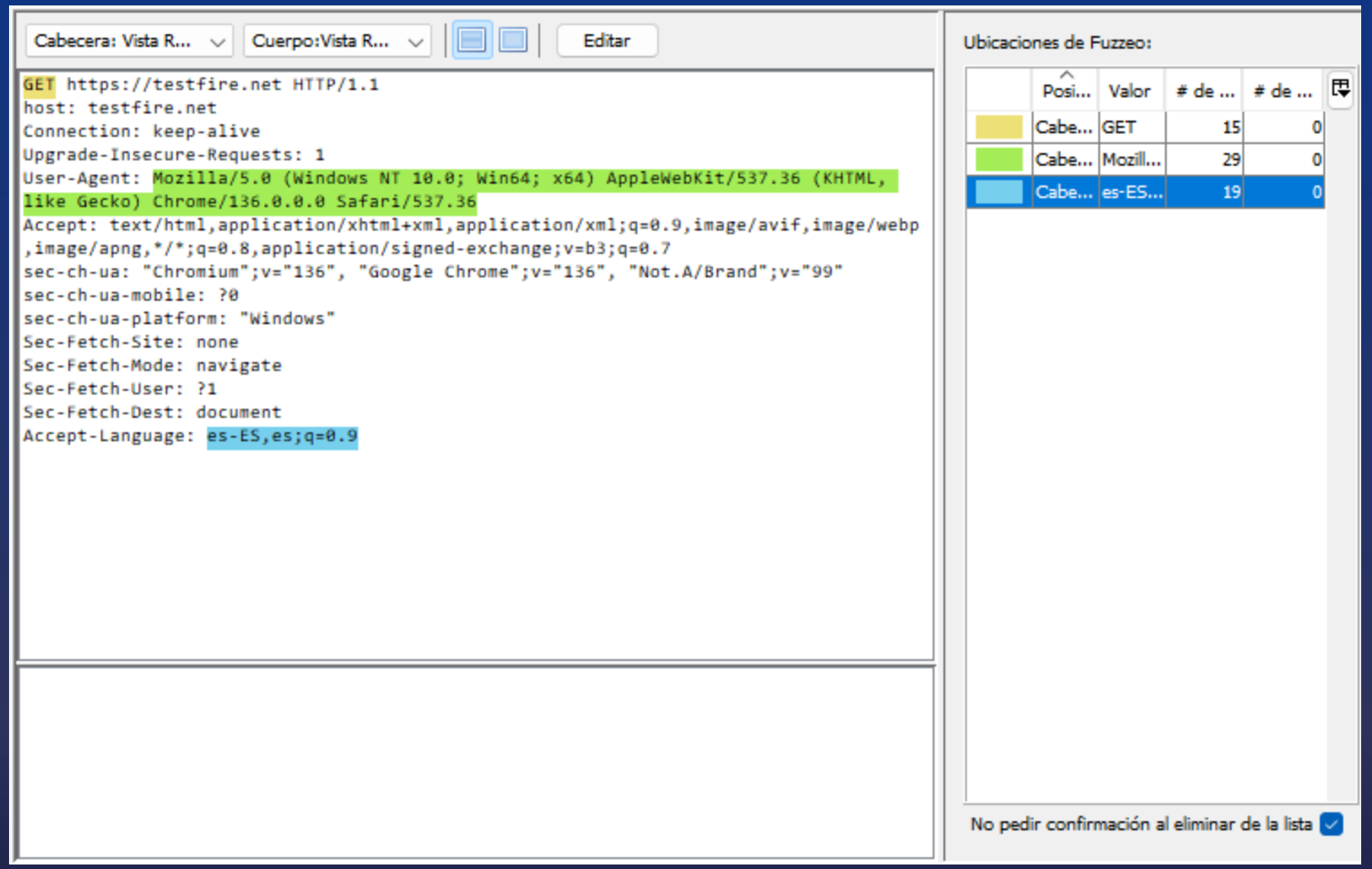
Caso de uso 4: fuzzing

Probando los límites del sistema



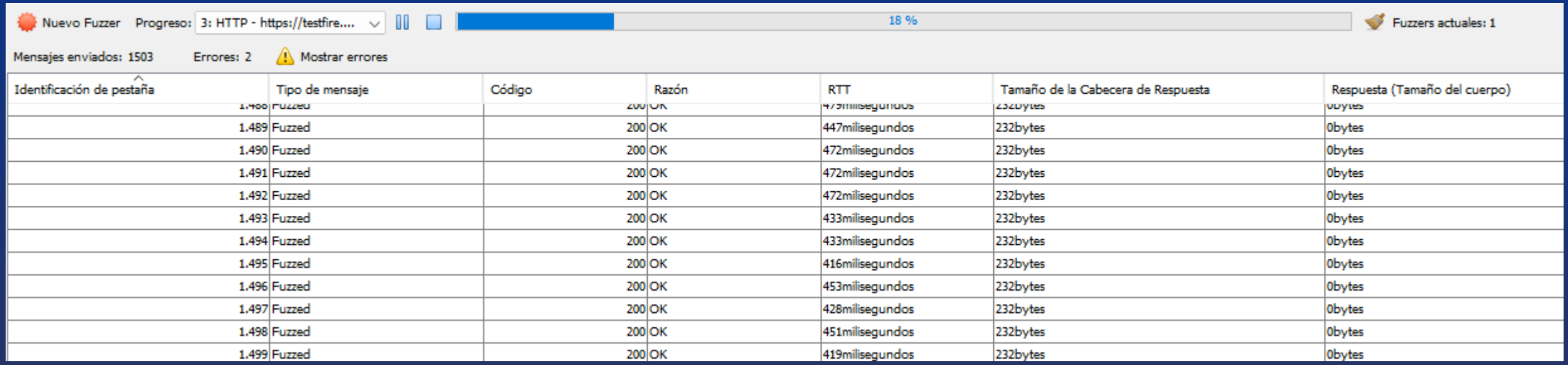
Caso de uso 4: fuzzing

Probando los límites del sistema



Caso de uso 4: fuzzing

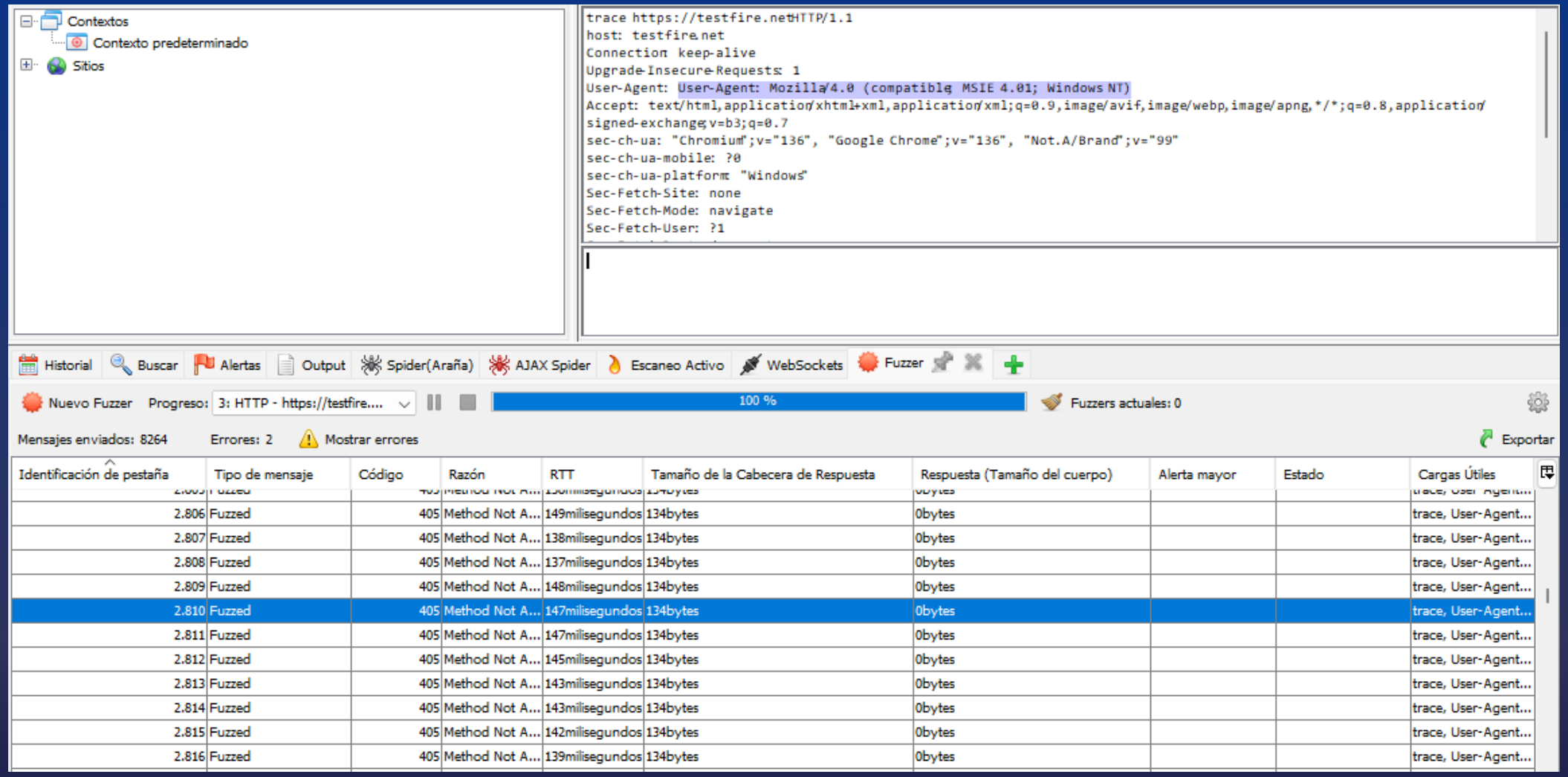
Probando los límites del sistema



Identificación de pestaña	Tipo de mensaje	Código	Razón	RTT	Tamaño de la Cabecera de Respuesta	Respuesta (Tamaño del cuerpo)
1.489	Fuzzed	200	OK	447milisegundos	232bytes	0bytes
1.490	Fuzzed	200	OK	472milisegundos	232bytes	0bytes
1.491	Fuzzed	200	OK	472milisegundos	232bytes	0bytes
1.492	Fuzzed	200	OK	472milisegundos	232bytes	0bytes
1.493	Fuzzed	200	OK	433milisegundos	232bytes	0bytes
1.494	Fuzzed	200	OK	433milisegundos	232bytes	0bytes
1.495	Fuzzed	200	OK	416milisegundos	232bytes	0bytes
1.496	Fuzzed	200	OK	453milisegundos	232bytes	0bytes
1.497	Fuzzed	200	OK	428milisegundos	232bytes	0bytes
1.498	Fuzzed	200	OK	451milisegundos	232bytes	0bytes
1.499	Fuzzed	200	OK	419milisegundos	232bytes	0bytes

Caso de uso 4: fuzzing

Probando los límites del sistema



Identificación de pestaña	Tipo de mensaje	Código	Razón	RTT	Tamaño de la Cabecera de Respuesta	Respuesta (Tamaño del cuerpo)	Alerta mayor	Estado	Cargas Útiles
2.806	Fuzzed	405	Method Not A...	149milisegundos	134bytes	0bytes			trace, User-Agent...
2.807	Fuzzed	405	Method Not A...	138milisegundos	134bytes	0bytes			trace, User-Agent...
2.808	Fuzzed	405	Method Not A...	137milisegundos	134bytes	0bytes			trace, User-Agent...
2.809	Fuzzed	405	Method Not A...	148milisegundos	134bytes	0bytes			trace, User-Agent...
2.810	Fuzzed	405	Method Not A...	147milisegundos	134bytes	0bytes			trace, User-Agent...
2.811	Fuzzed	405	Method Not A...	147milisegundos	134bytes	0bytes			trace, User-Agent...
2.812	Fuzzed	405	Method Not A...	145milisegundos	134bytes	0bytes			trace, User-Agent...
2.813	Fuzzed	405	Method Not A...	143milisegundos	134bytes	0bytes			trace, User-Agent...
2.814	Fuzzed	405	Method Not A...	143milisegundos	134bytes	0bytes			trace, User-Agent...
2.815	Fuzzed	405	Method Not A...	142milisegundos	134bytes	0bytes			trace, User-Agent...
2.816	Fuzzed	405	Method Not A...	139milisegundos	134bytes	0bytes			trace, User-Agent...

Caso de uso 5: exploración manual

Análisis silencioso del tráfico

- Análisis sin intrusión
- Reducción de falsos positivos
- Detección de fallos lógicos
- Útil en acciones complejas (autenticación)

Exploración manual VS Escaneo activo

Tipo de escaneo	Manual	Automático / Activo
Intrusión	❌ No	☑ Sí
Seguridad para producción	☑ Alta	❌ Arriesgado
Detecta configuración insegura	☑ Sí	☑ Sí
Detecta vulnerabilidades técnicas (XSS, SQLi, etc.)	❌ No	☑ Sí
Carga sobre el servidor	◍ Nula	◍ Alta
Cobertura completa	◍ Limitada	☑ Amplia (si se configura bien)

Caso de uso 6: integración continua

Seguridad desde tu CI/CD

Desarrollo

Programadores escriben nuevo código

Control de Versiones

Cambios enviados al repositorio

Despliegue

Código seguro en producción

ZAP Automático

Disparo del escaneo de seguridad

https://plugins.jenkins.io/zap/

https://github.com/zaproxy/action-baseline/tree/master

https://gitlab.com/gitlab-org/security-products/dependencies/zaproxy

ZAP Proxy (@ zaproxy.org)

100%

Gratuito

Código abierto y accesible para todos

1.000+

Casos de uso

Desde desarrollo hasta auditorías profesionales

24/7

Comunidad

Soporte global y actualizaciones constantes

¡Gracias a nuestro colaborador!

SDi Digital Group

...por invitarnos a sus instalaciones.


OWASP
LOGROÑO - LA RIOJA