**OWASP ZAP**

**Enrique José Rodríguez Martín**

*Uo257565@uniovi.es*

**Resumen**

En este trabajo se va a realizar una pequeña introducción al uso de una de las herramientas de pentesting más utilizadas en la actualidad OWASP ZAP, se hablará de su instalación en un sistema Windows y como utilizar sus más básicas funciones siguiendo un tutorial de la web.

1. **Introducción**

OWASP ZAP se anuncia como una de las herramientas de seguridad de código abierto más utilizadas del mundo, mantenida gracias al esfuerzo de múltiples usuarios de todo el mundo. Entre sus funcionalidades destaca la capacidad de encontrar automáticamente vulnerabilidades en tus aplicaciones web mientras desarrollas y testeas las mismas. También es una excelente herramienta para pentesters experimentados que quieren hacer comprobaciones manuales sobre seguridad. [1] Esta herramienta está creada en Java por lo que puede ser utilizada desde cualquier sistema operativo y cuando se utiliza como servidor proxy es capaz de otorgar a los usuarios la capacidad de manipular todo el tráfico que pasa a través del proxy, incluido el tráfico del protocolo HTTPS. [2] Las siglas ZAP provienen del inglés *“Zed Attack Proxy”*.

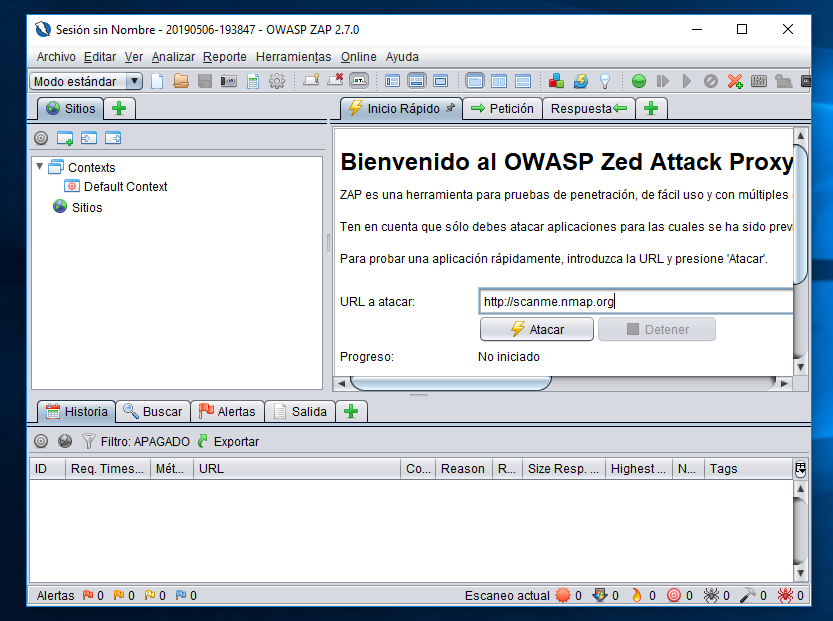
1. **Instalación**

Primero nos dirigimos a su página oficial [1], una vez allí pulsamos sobre el botón que nos redirige a su pagina de descarga, en nuestro caso, vamos a seleccionar el instalador de Windows de 64 bits, una vez con el ejecutable en nuestro poder, procedemos a la instalación, si la instalación se ha completado satisfactoriamente y hemos elegido la instalación estándar, deberíamos tener un nuevo icono de escritorio como este:



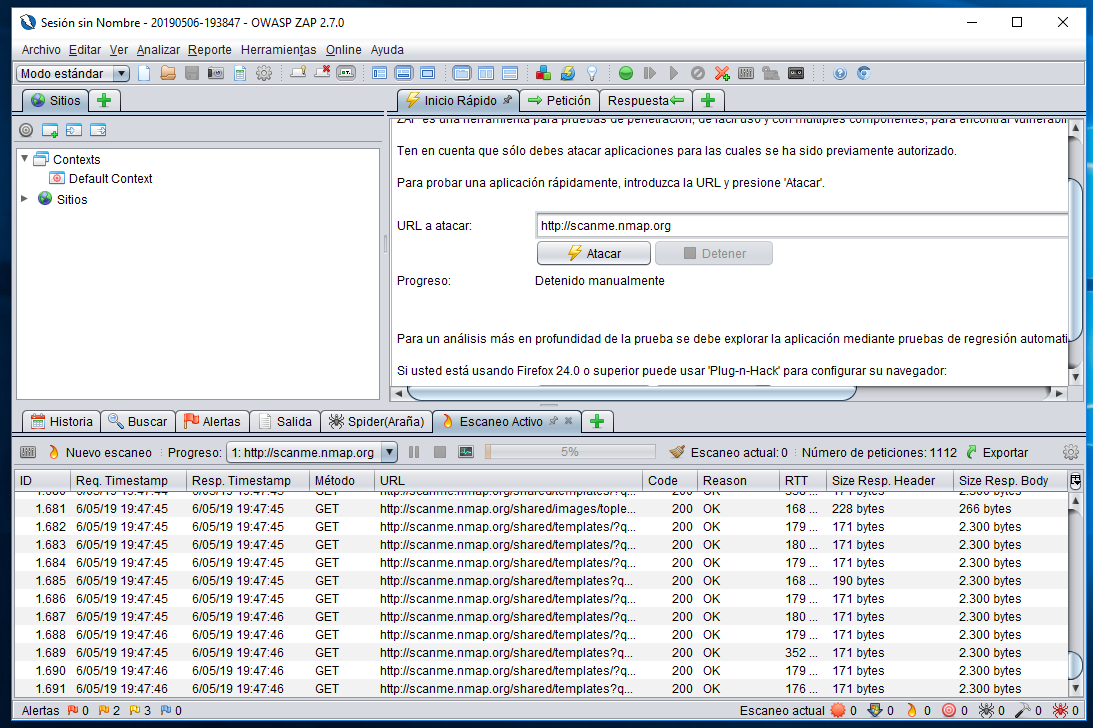
**Primeros pasos [3]**

Para este trabajo vamos a seguir el tutorial sobre la herramienta que podemos encontrar en la siguiente página web [4]. Iniciamos el programa y siguiendo los pasos del primer apartado del tutorial debemos ingresar la URL a atacar, en nuestro caso será “*<http://scanme.nmap.org>”*, hay que tener en cuenta que debemos ingresar en el campo del formulario el protocolo con el que se comunica el sitio a atacar ya que de no hacerse, se nos mostrará diálogo de error. Simplemente tenemos que pulsar el botón atacar de la pestaña inicio rápido.



Podemos detener el escaneo activo cuando queramos e incluso exportar los resultados a un .csv:

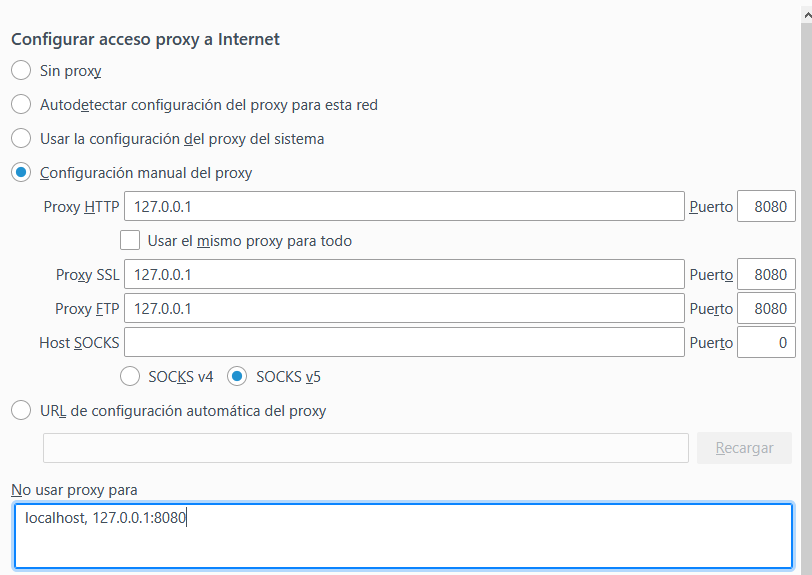




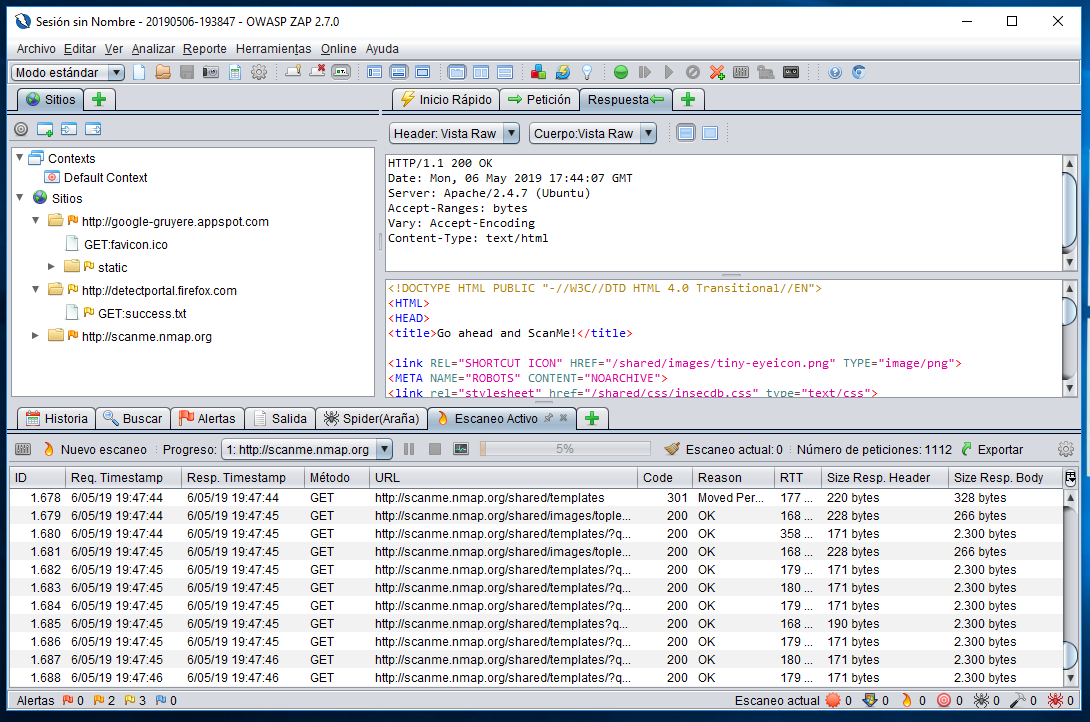
Esto nos permite analizar con mayor detenimiento los datos obtenidos incluso realizar una serie de filtrados posteriores utilizando otras herramientas que nos pueden esclarecer o precisar la búsqueda.

**Interceptación por proxy: [5]**

Para poder realizar este apartado modificaremos nuestra configuración, para ello vamos a utilizar el navegador Firefox ya que es un navegador más cómodo a lo que editar la configuración se refiere:



Ahora accedemos a *“Google Gruyere”* [6] desde Firefox. Si teníamos previamente ZAP abierto deberíamos apreciar varios cambios ya que ZAP ha comenzado a analizar los sitios que hemos visitado, pudiendo nosotros observar el contenido de las peticiones y respuestas entre nuestro navegador y el servidor que aloja la web.



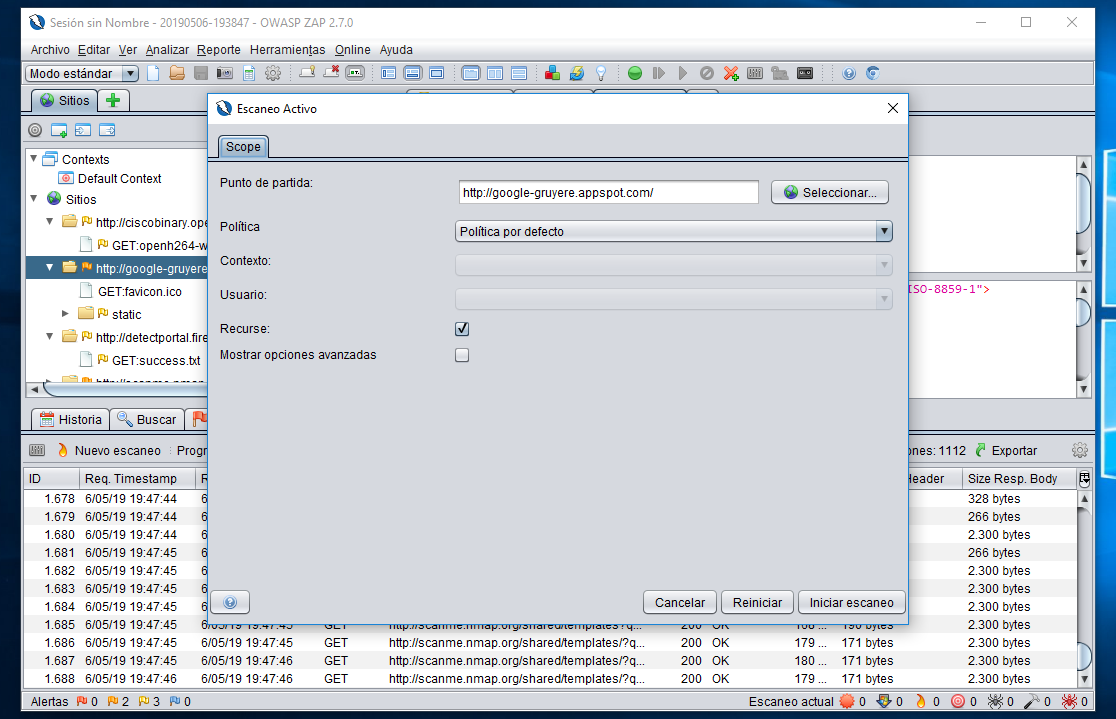
ZAP nos permite en este momento analizar el tráfico con este sitio o con otros si queremos simultáneamente analizar varios sitios, nos permite, por ejemplo comprobar cómo se envían las credenciales a nuestro servidor para así evitar exponer a nuestros usuarios, nos permite filtrar las peticiones con el sitio para tener un entorno de trabajo más organizado o simplemente queremos afinar nuestra búsqueda, también podemos utilizar las funciones de *“breakpoints”* es decir podemos detener peticiones antes de enviarlas para un análisis paso a paso más exhaustivo, esta funcionalidad nos permite modificar las peticiones y comprobar como reaccionaría un sistema en caso de ataque.

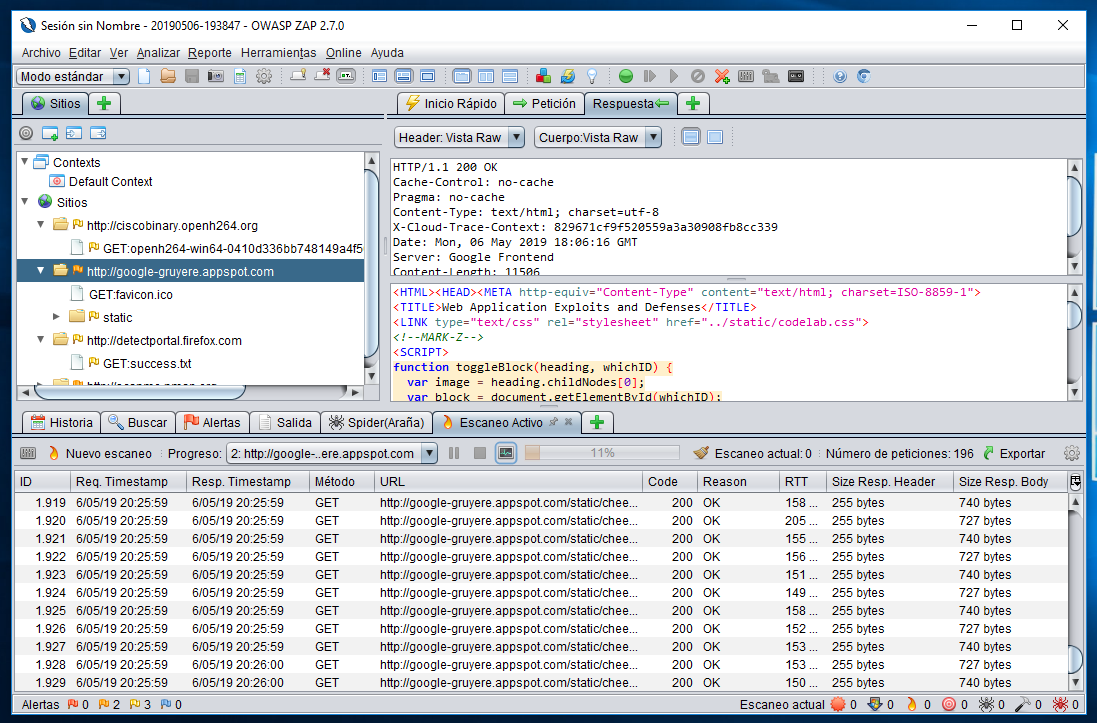
**Escaneo pasivo con ZAP [7]**

ZAP nos permite escanear de forma pasiva las respuestas del servidor de forma no intrusiva, podemos definir las reglas que utiliza usando la ventana de ajustes, pudiendo añadir, modificar y eliminar dichas reglas. Esto nos perite analizar con muchísima más precisión la interacción de nuestro buscador con el servidor. Estas reglas tienen múltiples funcionalidades como detectar comentarios, direcciones de correo, cookies entre otros elementos de interés para nuestro análisis.

**Escaneo activo con ZAP [7]**

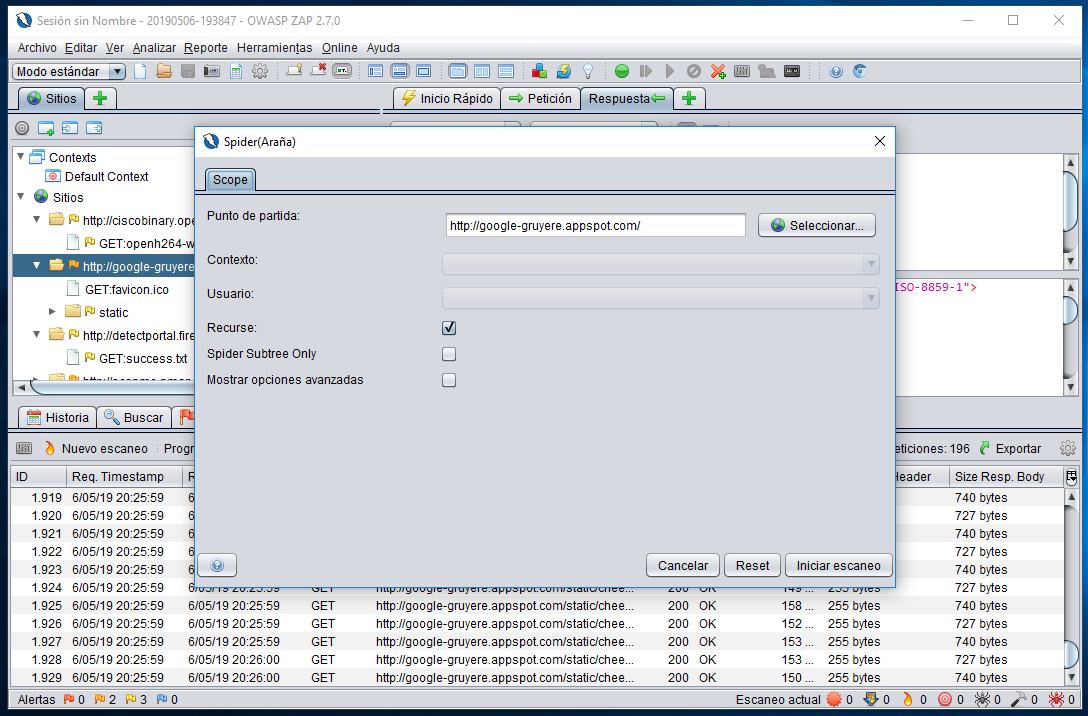
De la misma forma que hemos realizado con anterioridad un escaneo pasivo podemos realizarlo de manera activa contra un sitio, para ello solo tenemos que haber visitado la página, es decir, tenerla en nuestra pestaña de sitios y mediante clic derecho -> atacar -> Iniciar escaneo … el programa iniciará el escaneo. Se recomienda antes de realizar un escaneo activo haber realizado con anterioridad un escaneo pasivo aumentando la granularidad de nuestro ataque.

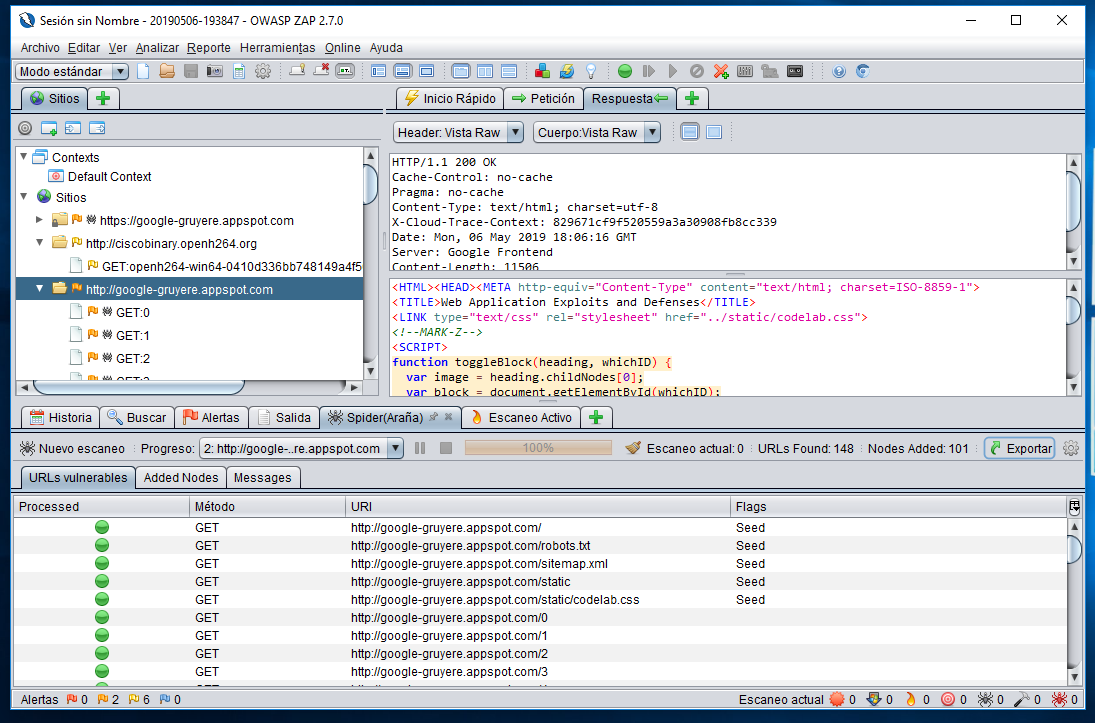




**ZAP web crawling [8]**

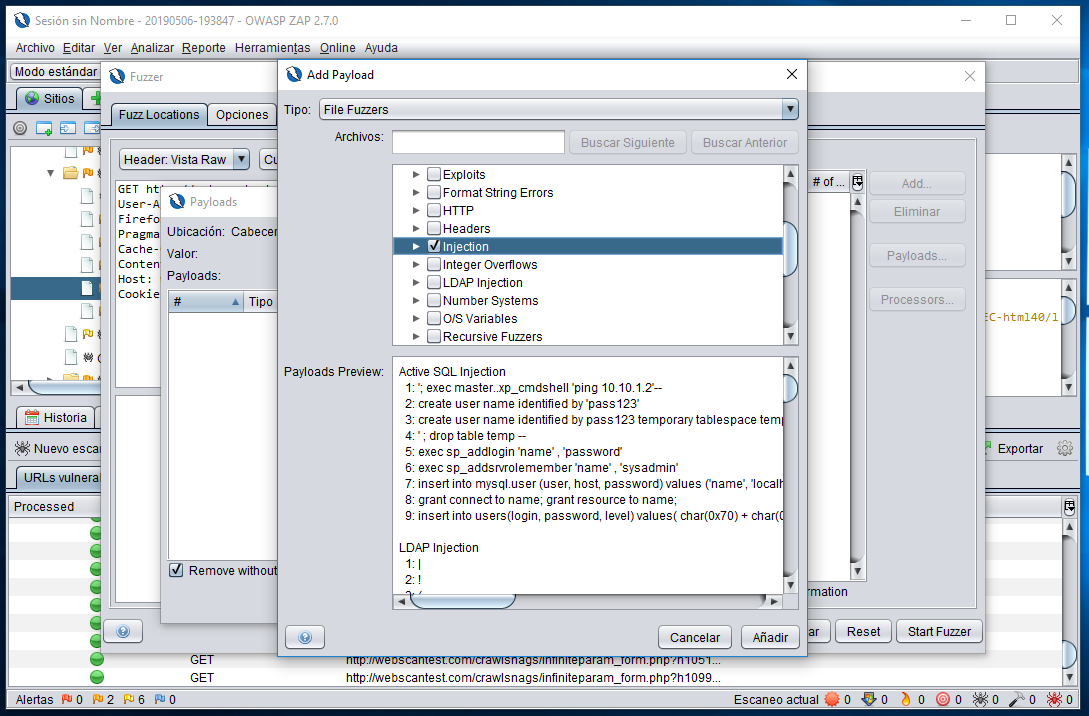
ZAP cuenta con funcionalidades de web crawling, esto nos permite identificar los enlaces que están presentes en un sitio web, con esto conseguimos analizar cómo se compone nuestro sitio web y desentrañar cómo se estructuran tanto los directorios como posibles archivos existentes en los cuales podemos centrar nuestra auditoría, el web crawler de ZAP funciona de forma recursiva, eso quiere decir que cada vez que encuentra un nuevo enlace lo va siguiendo, identificando los elementos; existe la posibilidad de una mayor granularidad que puede ser definida en la ventana de configuración; de esta forma iremos afinando aún más nuestra búsqueda y otorgándonos datos de mayor relevancia para nuestro análisis. Si buscamos un análisis general del sitio solo tenemos que realizar un escaneo pasivo y dirigirnos a la pestaña de sitios -> clic derecho -> Spider URL. Para una mayor precisión se recomienda usar el resto de las funciones Spider.



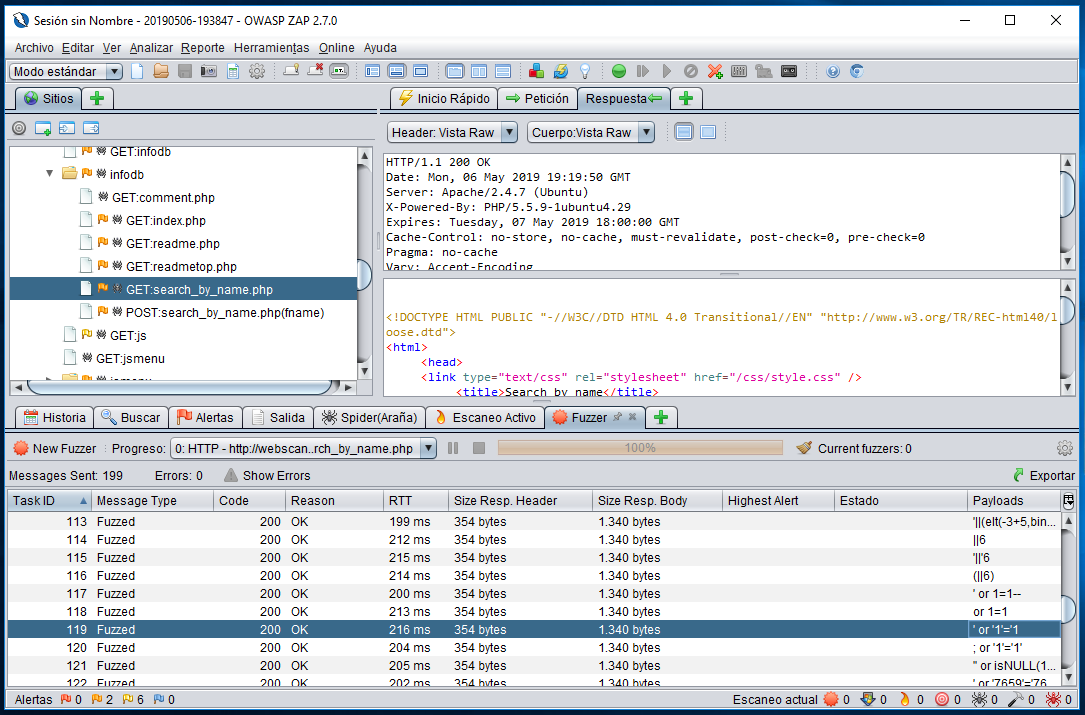


**Fuzzing [9]**

El fuzzing son las diferentes técnicas que permiten generar y enviar datos aleatorios y secuenciales a una aplicación con el fin de conocer sus vulnerabilidades [10], estás técnicas previenen a nuestras aplicaciones por ejemplo de ser susceptibles a ataques de inyección SQL al permitirnos comprobar con anterioridad al despliegue si es nuestro software vulnerable a este tipo de ataques. Vamos a realizar un ataque de inyección SQL contra el sitio “[*http://webscantest.com*](http://webscantest.com)*”*. Después de realizar un ataque con la herramienta de web crawling antes descrita procedemos a analizar los posibles puntos débiles de la aplicación. Si nos fijamos en el árbol post análisis del sitio veremos en la carpeta *“infodb”* un método GET con el nombre *“search by name”*. Ahora pasaremos al ataque con las funciones de fuzzing para realizar una inyección SQL. Para ello realizamos clic derecho -> atacar fuzz -> Add … -> File Fuzzers -> jbrofuzz -> injection.



Una vez configurado el fuzz simplemente seleccionamos la opción Start fuzz de la primera ventana de fuzzing.

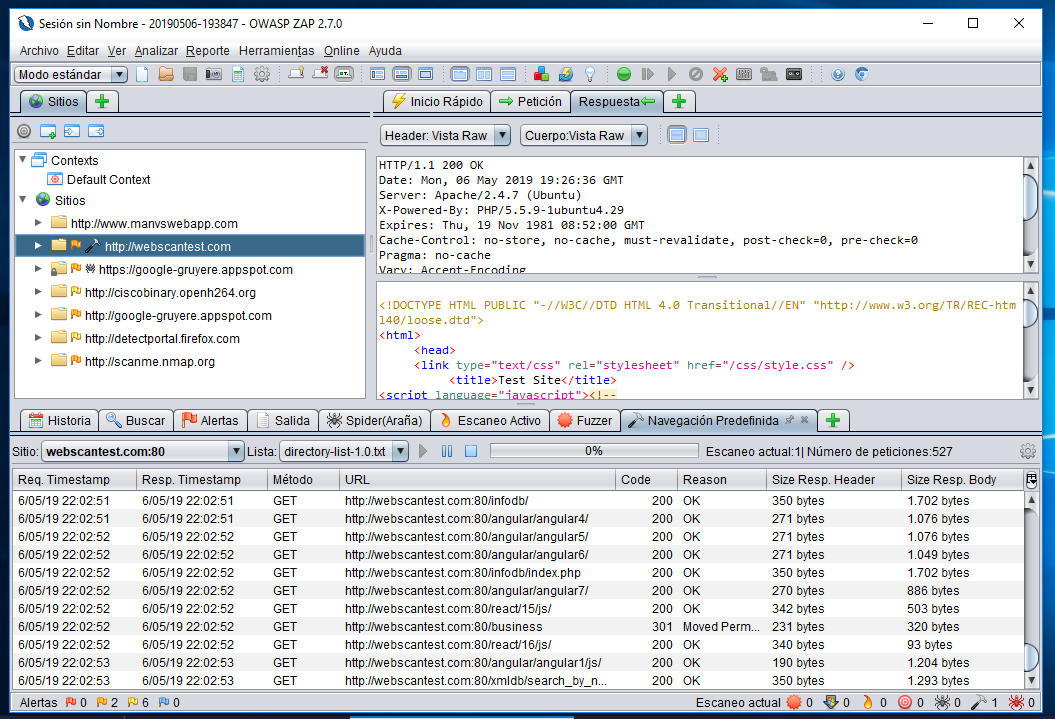


Comprobamos que en este caso nos retorna un código OK la sentencia SQL que nos mostraría el contenido completo de nuestra base de datos. Comprobamos que el ZAP está en lo correcto dirigiéndonos al sitio con nuestro navegador e introduciendo manualmente la cadena maliciosa.

**Forced Browse [11]**

Entendemos por forced browsing por el ataque que se centra en enumerar los accesos a recursos que no son referenciados por una aplicación [12], es decir, elementos que no son accesibles desde el uso normal de una web por un usuario, pero que conociendo la URL pueden ser accedidos con una intencionalidad maliciosa, es por ello por lo que es muy importante mantener la estructura de directorios limpia para evitar este tipo de ataques.

ZAP nos permite realizar este tipo de ataques, simplemente una vez visitado el sitio, clic derecho -> Atacar … -> Sitio de navegación predefinida. Cuando termine deberíamos obtener algo similar a esto.



Es posible realizar un ataque más granular con las funciones que nos otorga la propia herramienta si queremos por ejemplo conocer qué esconde un directorio concreto.

1. **Conclusión**

Podemos concluir que esta aplicación es fundamental en la “caja de herramientas” de todo pentester, sus funcionalidades descritas en los apéndices anteriores son muy variadas y versátiles, además cuenta con un elevado nivel de personalización que permiten al usuario, no solo analizar los datos obtenidos, sino que además permite realizar ataques y exploraciones de forma precisa y rápida; ofrece una variedad de opciones por defecto que cumplirán los requisitos mínimos que podamos requerir y debido a su buen diseño, esta herramienta puede ser mejorada gracias a addons realizados por la comunidad, los cuales permiten expandir las posibilidades de la herramienta aumentando su versatilidad.

1. **Referencias**
2. <https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project>
3. <https://es.wikipedia.org/wiki/OWASP_ZAP>
4. <https://www.sniferl4bs.com/2015/07/owasp-zaproxy-i-introduccion-zaproxy.html>
5. <https://blog.segu-info.com.ar/2015/09/tutorial-de-uso-owasp-zaproxy.html>
6. <https://www.sniferl4bs.com/2015/07/owasp-zaproxy-ii-intercept-proxy.html>

1. <http://google-gruyere.appspot.com/>
2. <https://www.sniferl4bs.com/2015/08/owasp-zaproxy-iii-escaneos-pasivo-y.html>
3. <https://www.sniferl4bs.com/2015/08/owasp-zaproxy-iv-zap-web-crawling.html>
4. <https://www.sniferl4bs.com/2015/08/owasp-zaproxy-v-fuzzing-con-zap.html>
5. <https://blog.segu-info.com.ar/2007/08/qu-es-fuzzing.html>
6. <https://www.sniferl4bs.com/2015/09/owasp-zaproxy-vi-zap-forced-browse.html>
7. <https://www.owasp.org/index.php/Forced_browsing>