

**CHALLENGE\_1 \_ SPRINT\_6**

**ANALISIS VULNERABILIDADES**

**SERVIDOR REDWEB**

**APLICACIONES DIRB Y OWASH ZAP**

EJERCICIO 1

**INTRODUCCION**

Pare este ejercicio, se ha analizado el servidor objeto de estudio con las aplicaciones dirb y Owasp ZAP, aplicación de escáner y análisis de vulnerabilidades y riesgos para servidores web.

-. APLICACIÓN\_DIRB. -

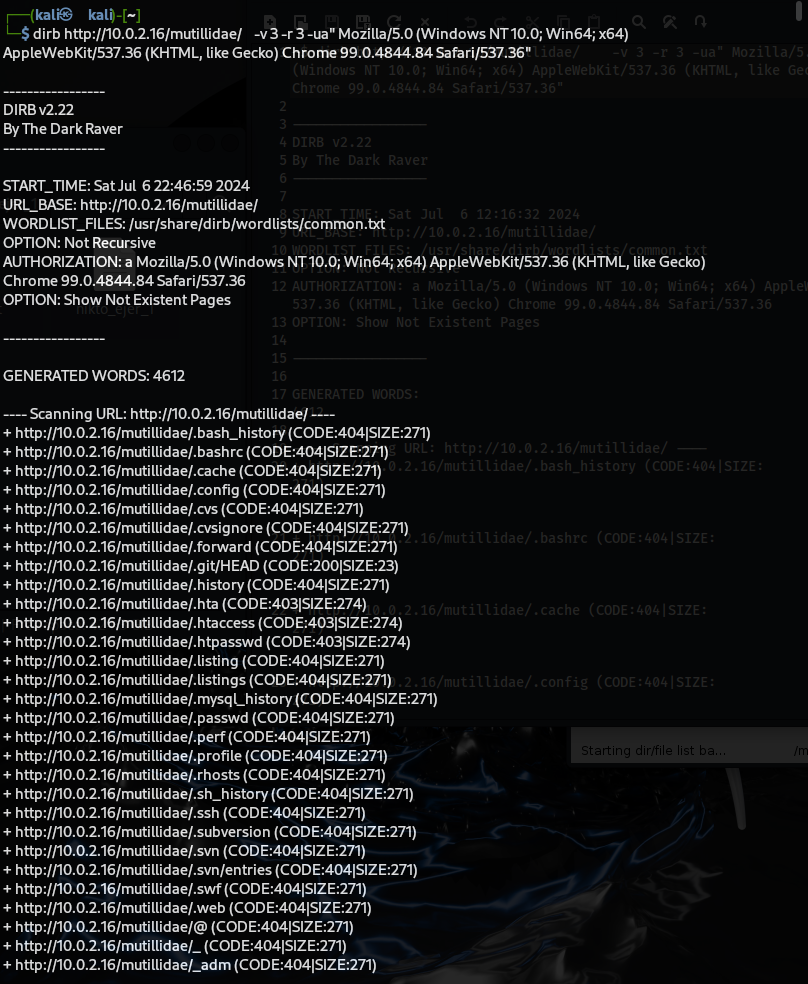
Aplicación para análisis de vulnerabilidades ejecutado directamente en CLI (terminal) realizando su ataque mediante uso de diccionarios, a través del cual, analiza todos los objetos disponibles en la web para conocer cuáles son vulnerables a técnicas hacking.

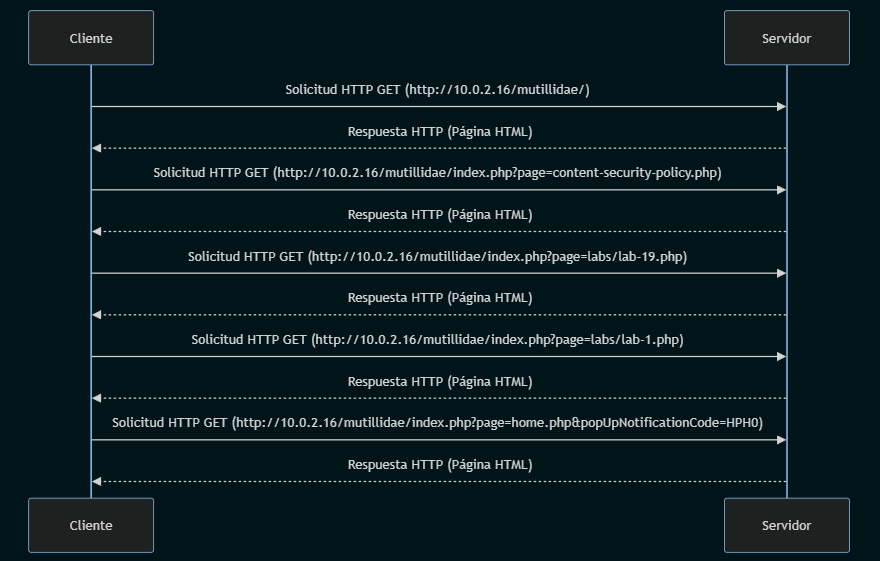
Se ha ejecutado el comando con varias opciones que viene en el apartado “*help”* del comando dirb:

*“dirb http://10.0.2.16/mutillidae/ -v 3 -r 3 -ua" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome 99.0.4844.84 Safari/537.36"*

Se ha lanzado el comando con la opción de verbosidad y profundidad de análisis máxima e intentado ocultar el ataque para camuflarse como si fuera la actividad de un navegador Mozilla 5.0.

Como resultado se han generado 4612 objetos, como se puede observar en la imagen y el esquema web:



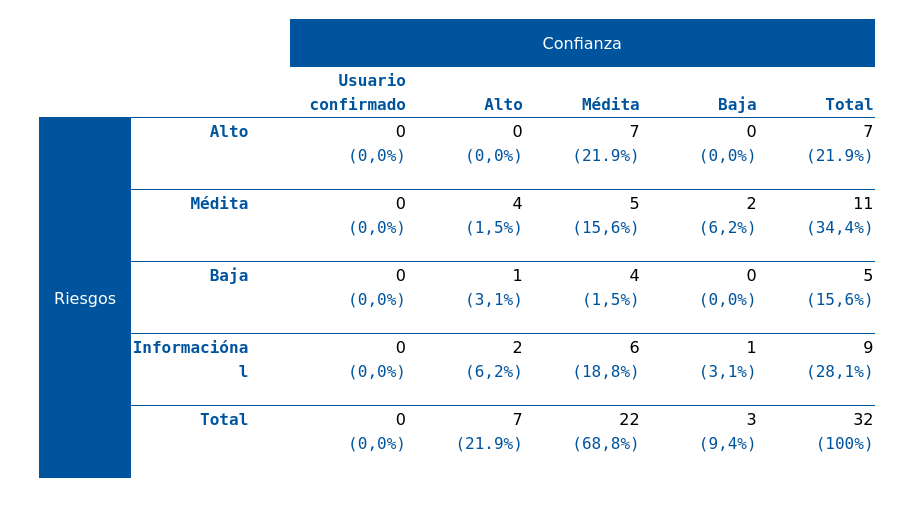


EJERCICIO 2

-. APLICACIÓN OWASP ZAP. –

Se realiza un nuevo escáner de seguridad de la web indicada en el ejercicio, con la aplicación Owasp Zap, la cual aporta muchos mas detalles en el análisis efectuado destacando:

Se han analizado el servidor <http://10.0.2.16> y <https://10.0.2.16>, estratificándolo en diferentes niveles, tanto de riesgo: Alto, medio, bajo e informativo, como de confianza: Usuario confirmado, alto , medio y bajo.



Como se puede observar en la gráfica, no se ha confi8rmado ningún Usuario, pero si se han detectado 11 de niveles medio-alto, concretamente 4 de riesgo medio y de confianza alto y 7 de riesgo alto y confianza media. Como mas graves, siendo el porcentaje mayor detectado en riesgo medio, en general.

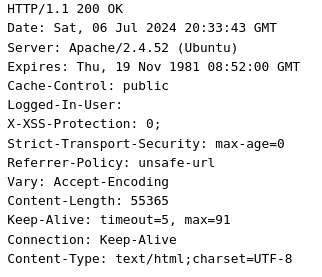


En esta pantalla, se puede observar las vulnerabilidades de riesgo alto detectadas y las partes del código afectadas. Analizando cada una de alertas generadas, destacan:

* [**Scripting de cross Site (Persistente)**](file:///\\home\kali\2024-07-06-ZAP-Report-.html#alert-type-0) **(1)**. – Un tipo de ataque que utiliza código malicioso por actor maliciosos en una instancia del navegador del usuario, pudiendo ser un cliente estándar del navegador, un objeto del mismo incrustado en un software del navegador (lector RSS) o cliente email. EL código, generalmente HTML/JavaScript, permite leer, modificar y transmitir cualquier dato sensible del navegador (robo de cuentas a través de las cookies, etc), siendo necesario el uso de bibliotecas anti-XSS para prevenir y mitigar estos ataques.

Solicitud

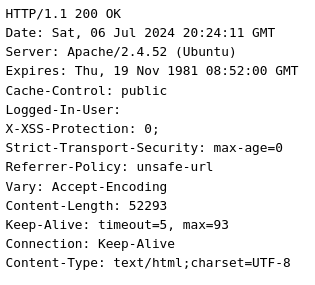
  
 Respuesta



* **Path Traversal. -** Este ataque permite el acceso a archivos, directorios y comandos que están fuera del directorio raíz o CGI del sitio web, manipulando una URL del servidor usando caracteres tipo *“/..”*, 0 caracteres Unicode, etc).Cualquier dispositivo, cuya interfaz este basada en HTTP es potencialmente vulnerable. Para evitar estos ataques, se pueden usar listas con permisos de entrada aceptables, que limiten el numero de caracteres “” o /, aunque la lista de denegación puede ser útiles para detectar posibles ataques.

**Solicitud**

**Respuesta**



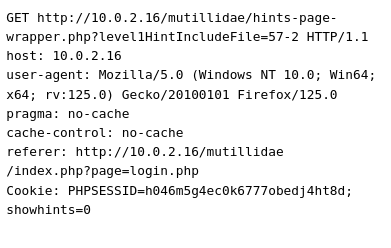
* **Remote File Include. –** Los ataques RFI son usados para la explotación de archivos dinámicos en aplicaciones web, que permiten incluir archivos. Estos archivos incluidos, pueden ejecutarse llamando a procedimientos específicos, tanto en el servidor como en el cliente, y si el módulo a cargar se basa en HTTP, la aplicación web es vulnerable a RFI. EL PHP es particularmente vulnerable a este ataque. Para prevenir los RFI, hay varias medias entre las que están usar S.= específicos (Unix, SElinux, etc), pero lo más fácil y accesible es montar el servidor en un entorno tipo sandbox que imponen limites entre el proceso y el S.O.

**º **

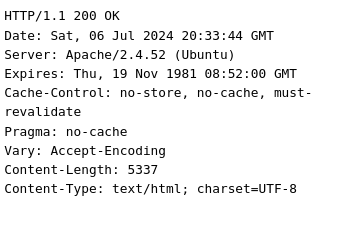
****

* **SQL Injection. -**  Este ataque consiste en la inserción p consulta SQL maliciosa a través de los datos de entrada del cliente, pudiendo acceder a datos sensibles contenidas en las BBDD. Para evitar estos ataques es necesario el uso de declaraciones preparadas, procedimientos almacenados, y escapando de toda entrega del usuario (Strongly Disccouraged), que deben incluir los desarrolladores web, así como incluir listas de permisos con caracteres permitidos y listas de denegados

**1 – Solicitud tipo MySQL**



**2.-Respuesta**



**1.- Solicitud tipo Oracle**



**2.- Ataque**

