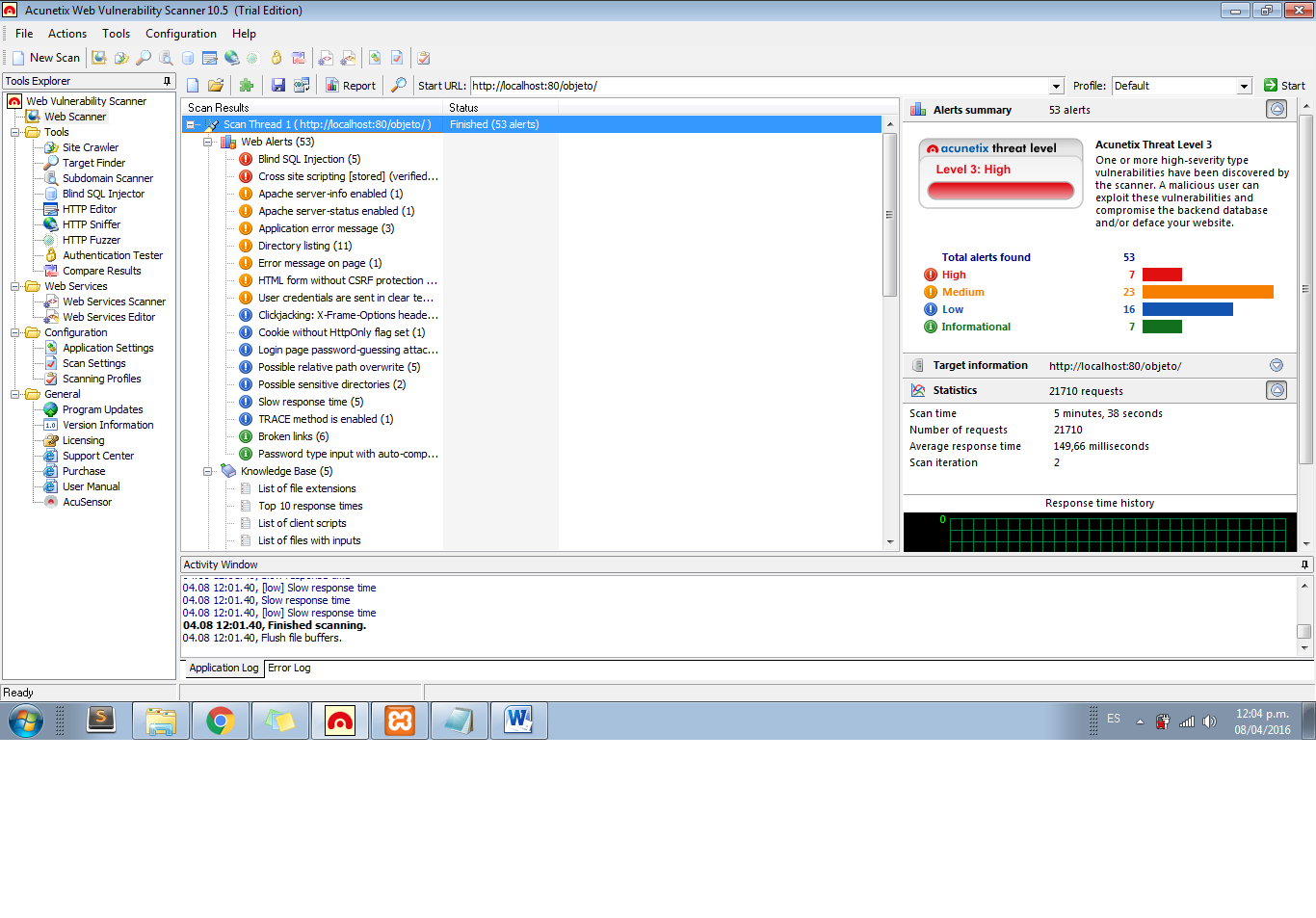
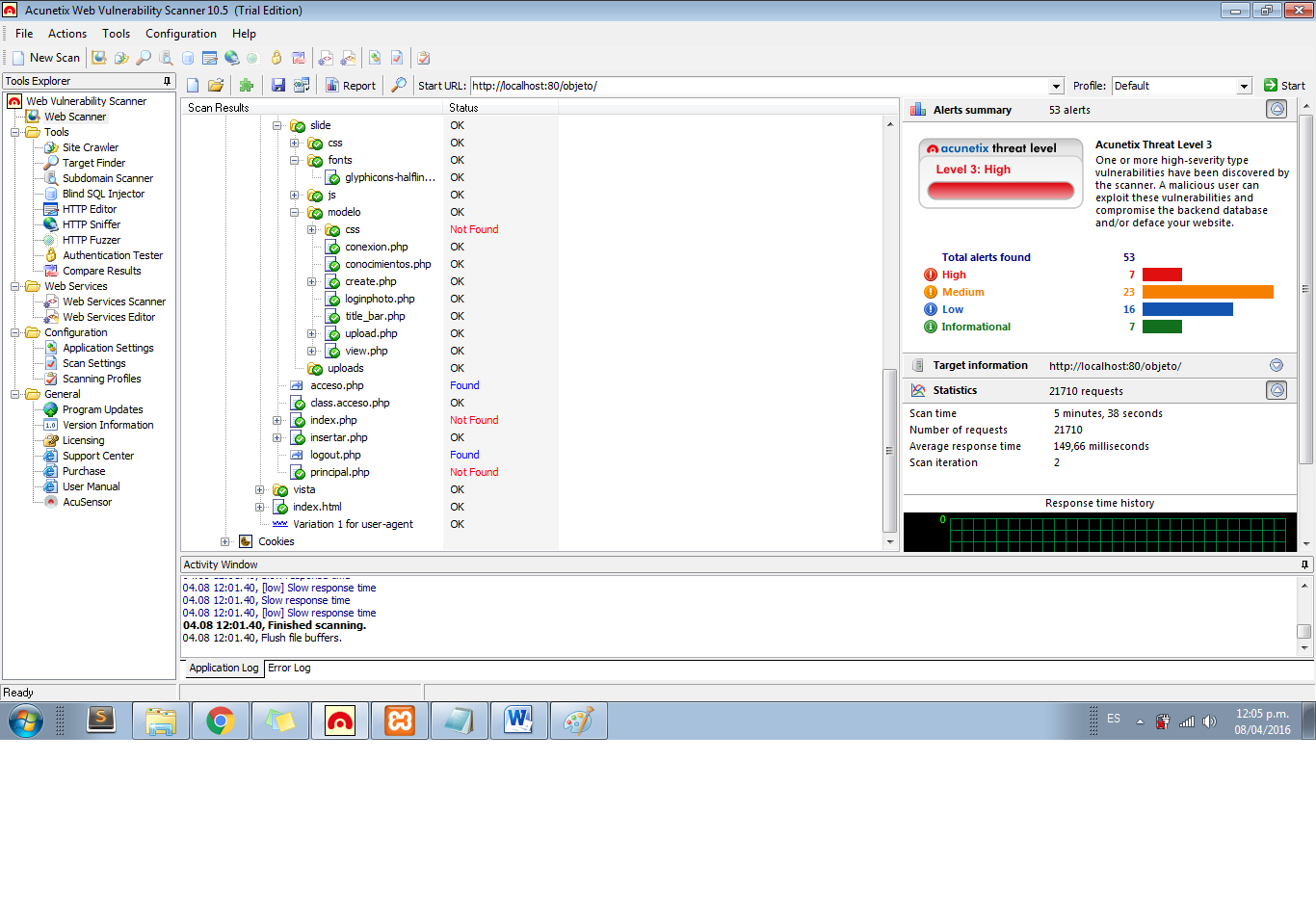
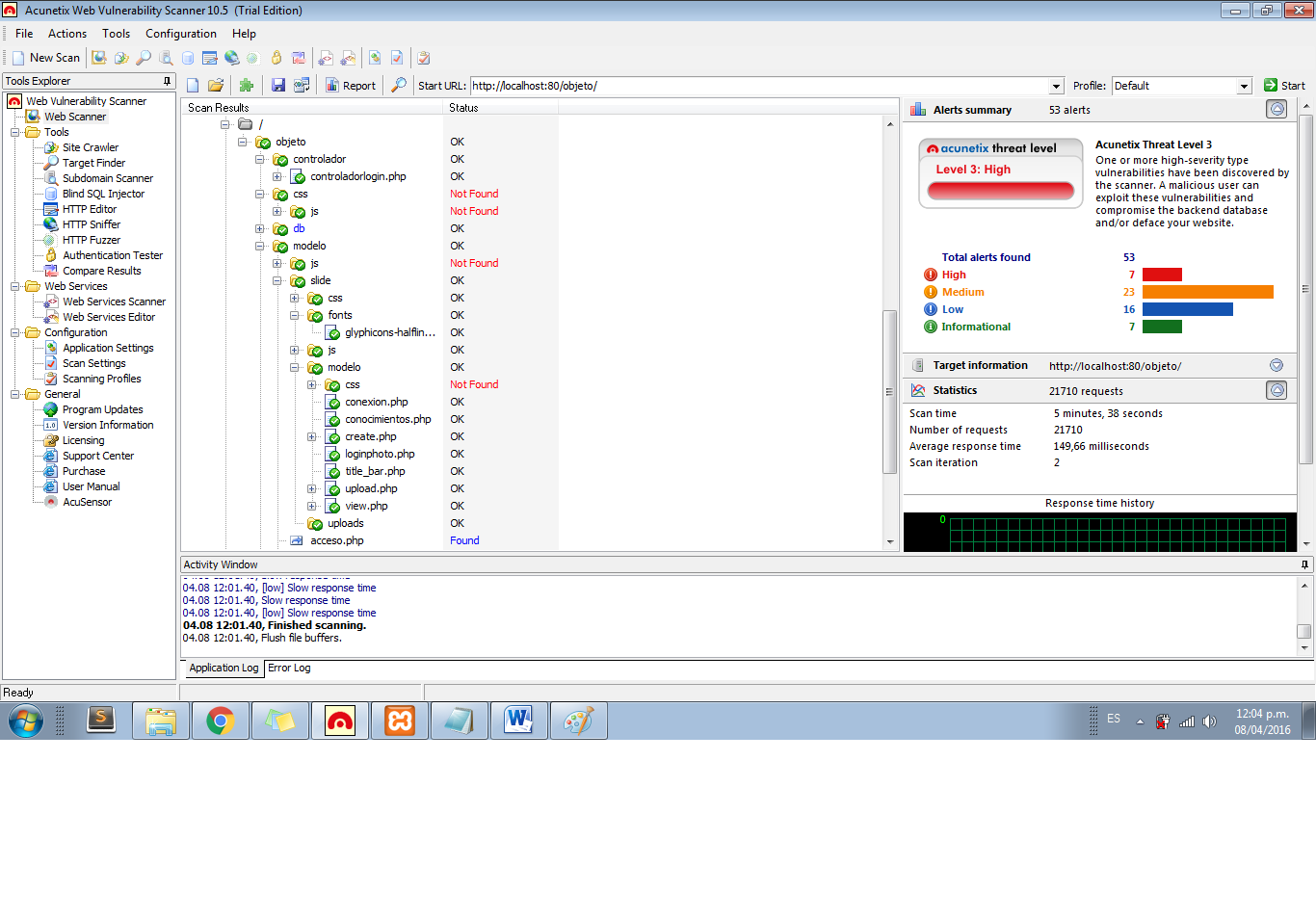
**Resumen vulnerabilidades en la aplicación de álbum de fotos**

Luego de haber analizado la aplicación con el contenido actual (posee login con MVC, y aplicación sin MVC). Se encontraron en total 53 vulnerabilidades en un tiempo de escaneo de 5 min y 38 segundos. En los siguientes screenshot mostramos el tiempo y la lista de vulnerabilidades encontradas.

**Tiempo promedio de escaneo :**



**Rutas de la aplicación 1 y 2 :**



Luego de mostrar los resultados y la estructura se puede hablar de las diferentes vulnerabilidades encontradas. En general, “ACUNETIX” puede diferenciar las distintas vulnerabilidades en grupos dependiendo de su complejidad y daño a realizar en las diferentes estructuras del proyecto.

**LVL 0 información:** Se encontraron dos errores, uno de ellos es que los links “están rotos” es decir no redirigen hacia ninguna parte, habría que mirar las vistas de estos y ver si hay que eliminarlos o no, el otro error se enfoca en la parte de que si la contraseña es guardada puede ser rellenada automáticamente por el navegador, la solución presentada seria la siguiente

<INPUT TYPE="password" AUTOCOMPLETE="off">

Donde el campo input no se pueda autocompletar.

**LVL 1 bajo:** Primera capa de errores que puede afectar la aplicación y que puede ser aprovechados por personas malintencionadas para fines indebidos.

**HTTP TRACE ENABLED :** El método trace, puede verse como un “echo” en php pero como cabecera de http. En general podemos inyectar código malicioso por medio de un código que el navegador pueda interpretar con el objetivo de un ataque.

Solucion : Quitar el método trace del servidor web atravez del apache en el archivo **httpd.conf.**

**SLOW RESPONSE TIME : tiempo de respuesta lento, en general se puede generar un ataque de denegación de servicio para sobrecargar el servidor, no hay solución genérica para esto. Hay que ver en la arquitectura si podemos mejorar nuestro código.**

**POSSIBLE SENSITIVE FILES : Archivos posiblemente sensibles, informacion importante se puede mostrar al atacante la cual puede ser backup, datos sql entre otros. La solución principal es sacar los archivos del web server o restringuir la llamada con el archivo .htaccess**

**POSSIBLE RELATIVE PATH OVERWRITE : Se puede sobreescribir código o informacion importante de algunas direcciones relativas utilizando html o javascript para esto. La solución seria mirar que huecos de seguridad se están dejando y reforzarlos.**

**LOGIN PAGE PASSWORD-GUESSING ATTACK : El ataque de fuerza bruta, adivinar contraseñas atravez de un diccionario u otro metodo. Una solución seria el numero de intentos que tiene el usuario.**

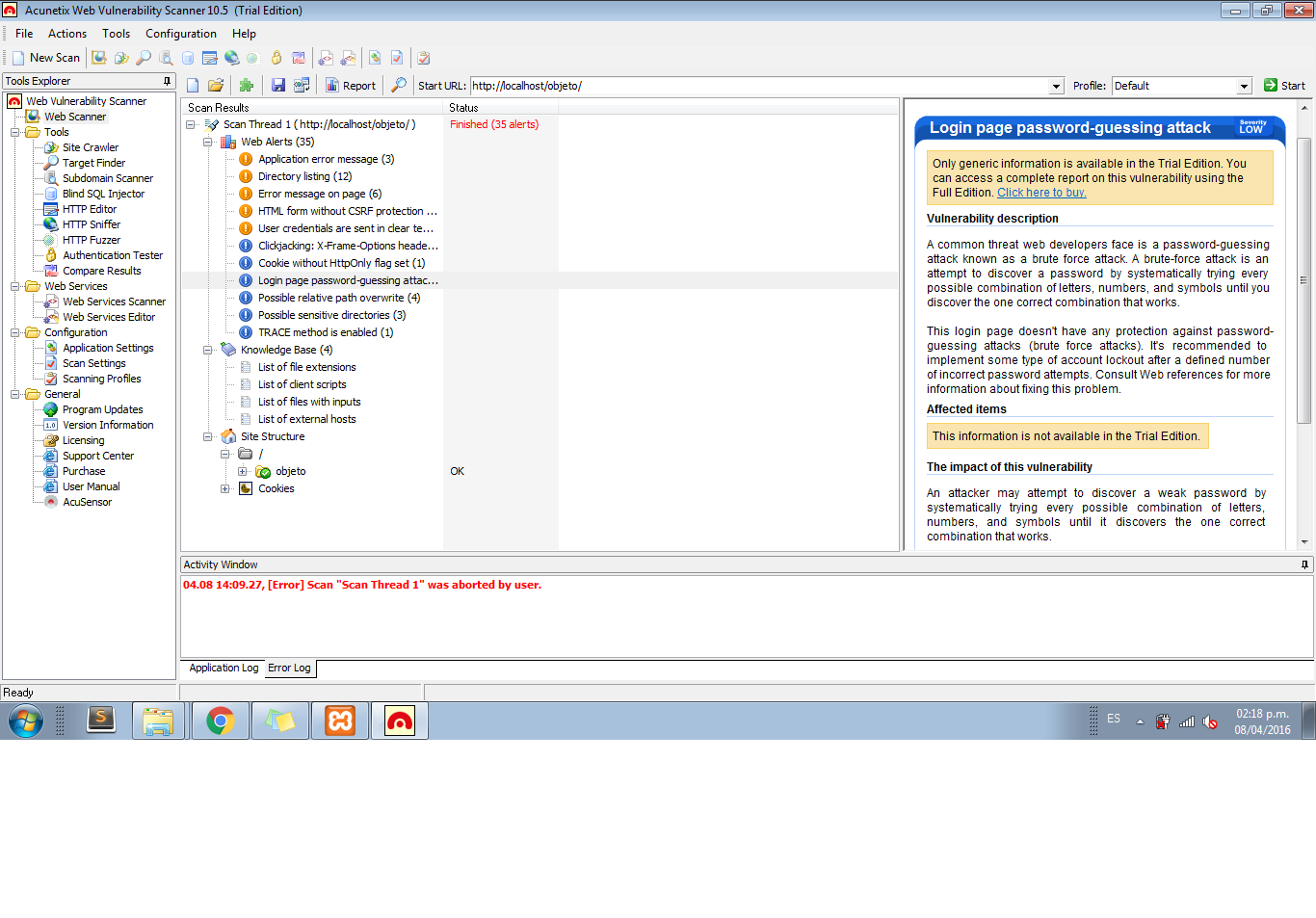
**COOKIE WITHOUT HTTPONLY FLAG SET: Generalmente esta bandera se active para saber que la informacion debe de manejarse por el servidor y no por el cliente. En general se debe restablecer esta bandera.**

**Mas informacion sobre como restaurar aca :** <https://edalmava.wordpress.com/2012/11/18/http-only-mitigando-el-cross-site-scripting-con-cookies/>

**CLICKJACKING : X-FRAME-OPTIONS HEADER MISSING : Generalmente este tipo de vulnerabilidad lo que busca es incrustar codigo malicioso en el servidor web con el objetivo de robar informacion del usuario atravez del cual el usuario realiza una acción la cual cree que es correcta pero como esta inyectada con código malicioso puede revelar informacion importante al atacante. Una solución seria incluir dentro del web server una opción de cabecera X-FRAME-HEADER.**

**El resto de los tipos de ataque se encuentran la pagina oficial de acunetix. Pasando de este tema, uno de los valores importante es revisar porque salieron estos errores. Algunos de ellos es porque esta en “PROCESO DE FABRICA”, se pueden mostrar los errores o acceder por puertos gracias al apache. Implementando algunas políticas rapidas de seguridad se logro mitigar algunas cosas.**

**Uno de los errores medios o lvl2 , era la inclusión de informacion del apache. Esto se controla desde el httpd.conf deshabilitando algunas opciones como la info o el status**



**Otras ataques como el CLICKJACKING con el siguiente código no funciono, se busca nuevas maneras de evadir dichos ataques.**

<httpProtocol>

<customHeaders>

<add name="X-Frame-Options" value="SAMEORIGIN" />

</customHeaders>

</httpProtocol>