Asignatura	Datos del alumno	Fecha
Seguridad en Aplicaciones Online	Apellidos: Mehrez Garcia	28-05-2022
	Nombre: Amir Fernando Mamdouh	

Actividad: Test de penetración a la aplicación web Badstore utilizando un scanner de aplicaciones web

Nombres y Apellidos: Amir Fernando Mamdouh Mehrez Garcia

Asignatura	Datos del alumno	Fecha
Seguridad en Aplicaciones Online	Apellidos: Mehrez Garcia	28-05-2022
	Nombre: Amir Fernando Mamdouh	

Objetivos

- Vas a aprender a encontrar vulnerabilidades de seguridad en una aplicación web.
- Vas a explotar vulnerabilidades de seguridad en una aplicación web.
- Vas a aprender a utilizar un scanner de vulnerabilidades de aplicaciones web a través de un procedimiento por fases.

Descripción

Realización de un test de penetración a la aplicación web Badstore.

Descarga:
ORACLE Virtualbox desde https://www.virtualbox.org/ e instala ZAP desde:
https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

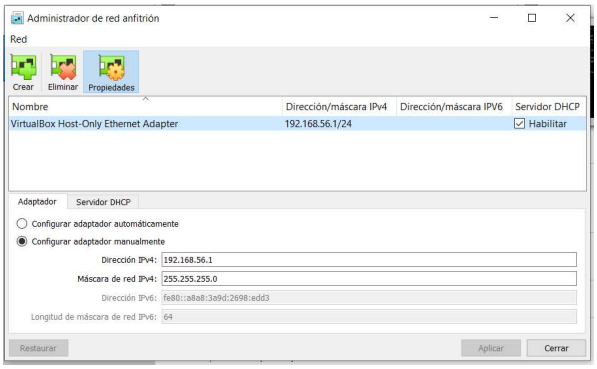
La máquina virtual con la aplicación BADSTORE, desde:
https://www.dropbox.com/sh/7ewzuosszqslkok/AADL6CSiXkoFPWdmfnwjHDLYa?dl=0

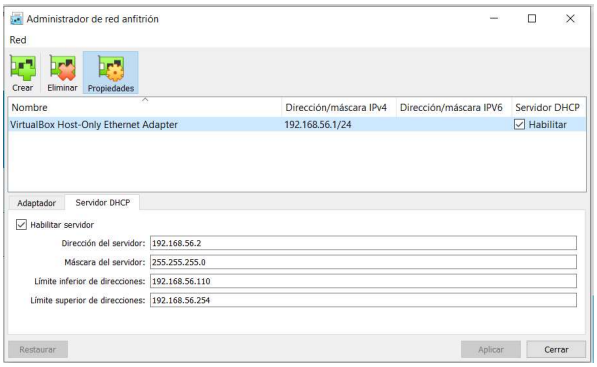
- Importa el servicio virtualizado badstore.ova desde ORACLE virtualbox.
  En configuración - almacenamiento, asocia la imagen BadStore-212.iso en el controlador IDE (cdrom) y configura la máquina virtual para que arranque primero desde el cdrom.

Como se ha llevado a cabo el procedimiento de test

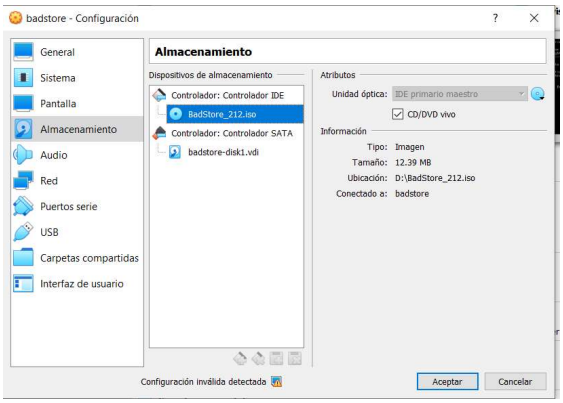
Asignatura	Datos del alumno	Fecha
Seguridad en Aplicaciones Online	Apellidos: Mehrez Garcia	28-05-2022
	Nombre: Amir Fernando Mamdouh	

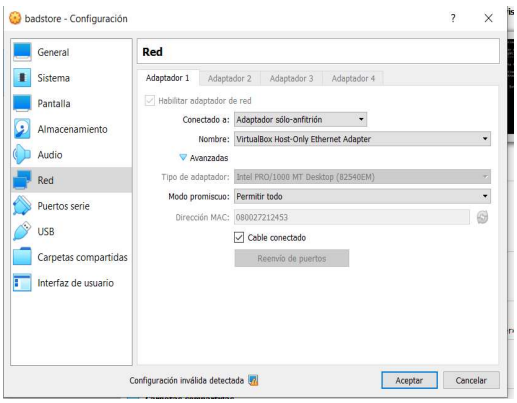
Lo primero es la creación de un adaptador ethernet del host-virtual, siguiendo los pasos de la guía, dentro de Oracle VM-> Archivo-> Administrador de red anfitrión y configuramos según lo solicitado.





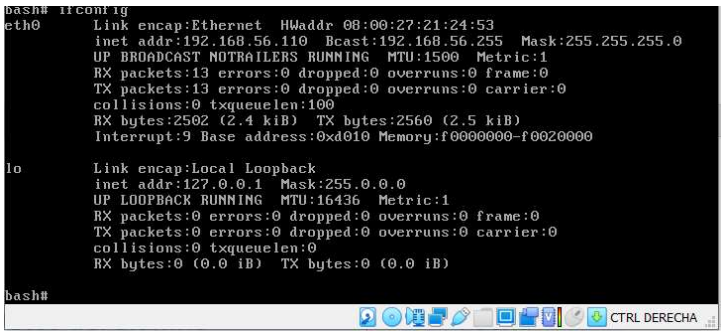
Luego de esto se procede con Archivo-> Importar un Servicio Virtualizado-> seleccionamos badstore.ova del link que se nos brindó https://www.dropbox.com/sh/7ewzuosszqslkok/AADL6CSiXkoFPWdmfnwjHDLYa?dl=0, configuramos adicionalmente antes del primer inicio. Como se muestra a continuación



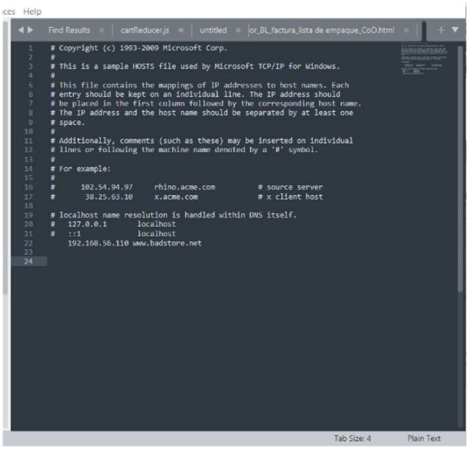


Despues del primer inicio, revisamos la ip asignada con el comando **ifconfig**.

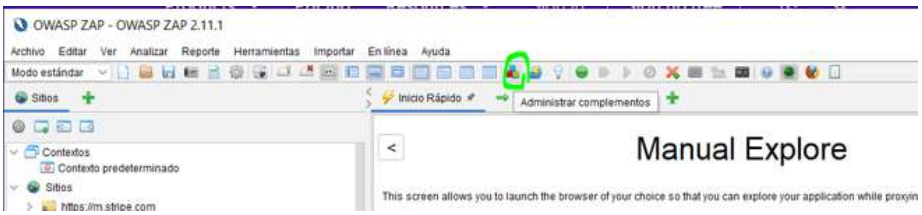
Asignatura	Datos del alumno	Fecha
Seguridad en Aplicaciones Online	Apellidos: Mehrez Garcia	28-05-2022
	Nombre: Amir Fernando Mamdouh	



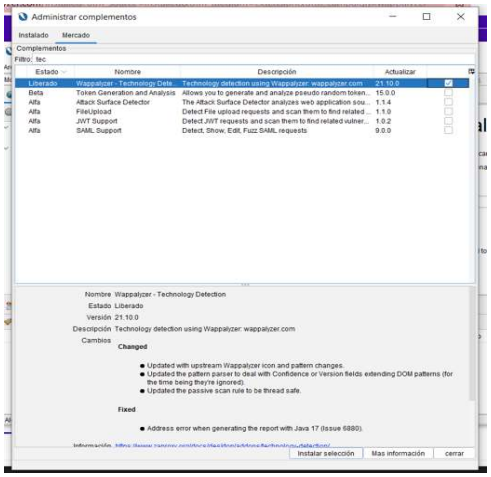
Lo que sigue es iniciar dentro de la maquina anfitriona y configurar el archivo C:\Windows\System32\drivers\etc\hosts (Windows 10) pero para esto previamente le damos anticlick propiedades->seguridad(aquí editamos usario y le damos permiso de modificar el archivo), luego agregamos la resolucion de la dns de badstore.net. Con esto al digitar la ip en este caso 192.168.56.110 nos redireccionara a badstore.net

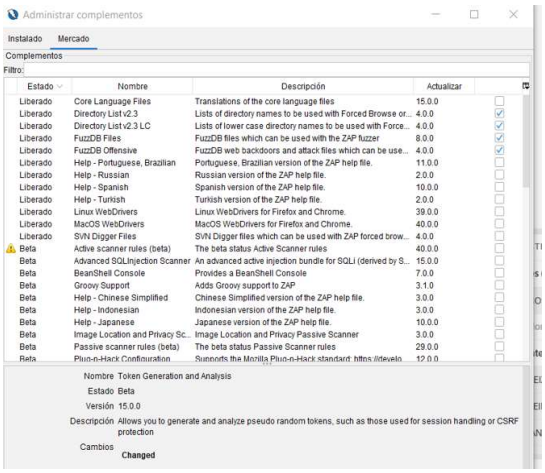


Lo siguiente se realizará dentro de OWASP ZAP. Accedemos a administrar complementos (click donde muestra la imagen)->Mercado->Wappalyzer. Esto nos permite ver la tecnología al momento de escanear, solo debemos agregarla.

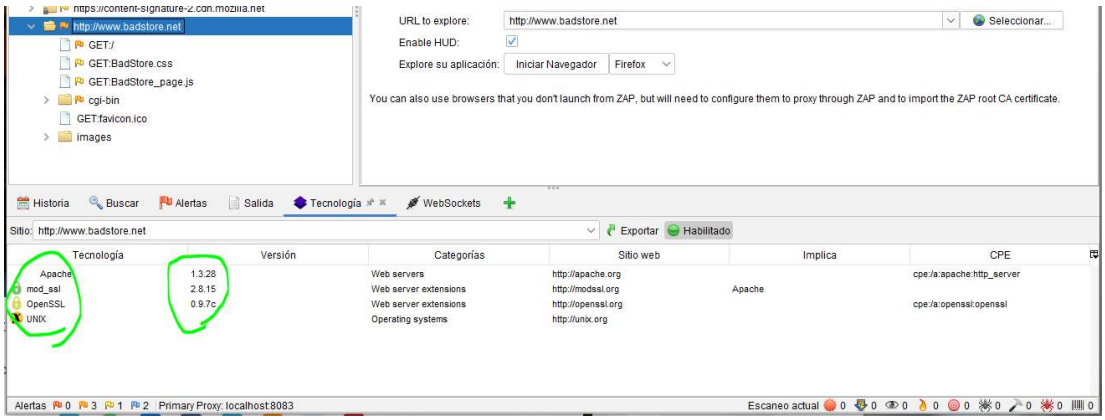


Lo siguiente es agregarle algunos plugins adicionales dentro de mercado como ya mencioné wappalyzer, directory list, fuzzdb files y offensive.



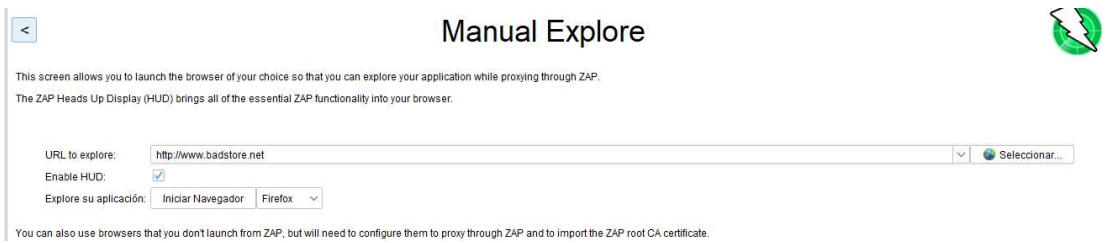


Agrego la pestaña tecnología del plugin de wappalyzer que instale previamente y obtengo la tecnología importante de badstore. En si que es un apache v1.3, tiene ssl v2.8 y un openssl v0.9; además de que se encuentra dentro de una versión de unix, específicamente trinux

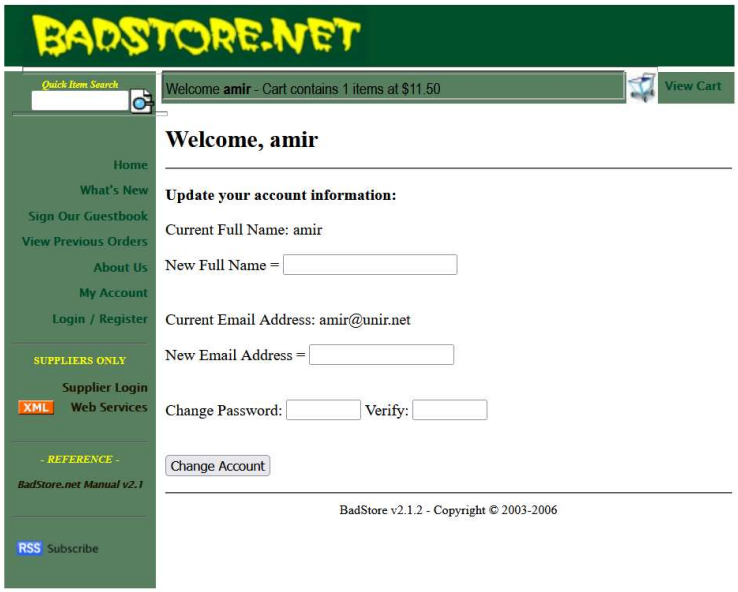


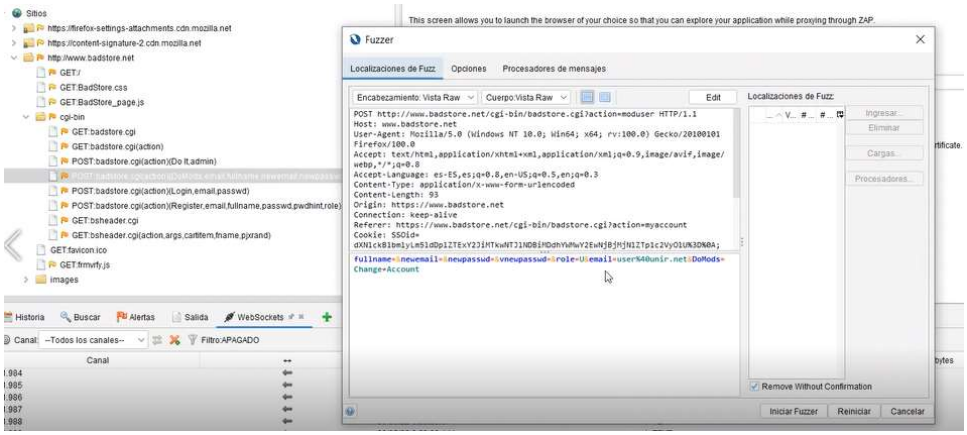
Hacking de Usuario Administrador

Para este caso intente 3 métodos, pero solamente me funciono uno, de todas formas mostrar un poco de cada uno de estos

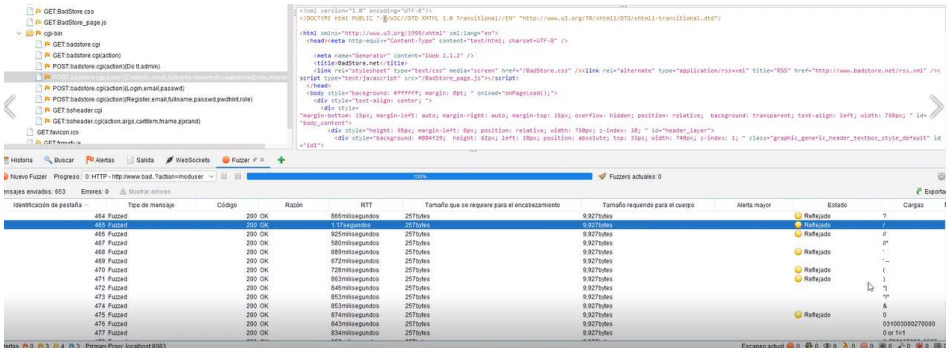


En el escaneo manual inicio con el navegador Firefox y al ir abriendo uno por uno los archivos, luego del registro me tope con una pagina llamada My account



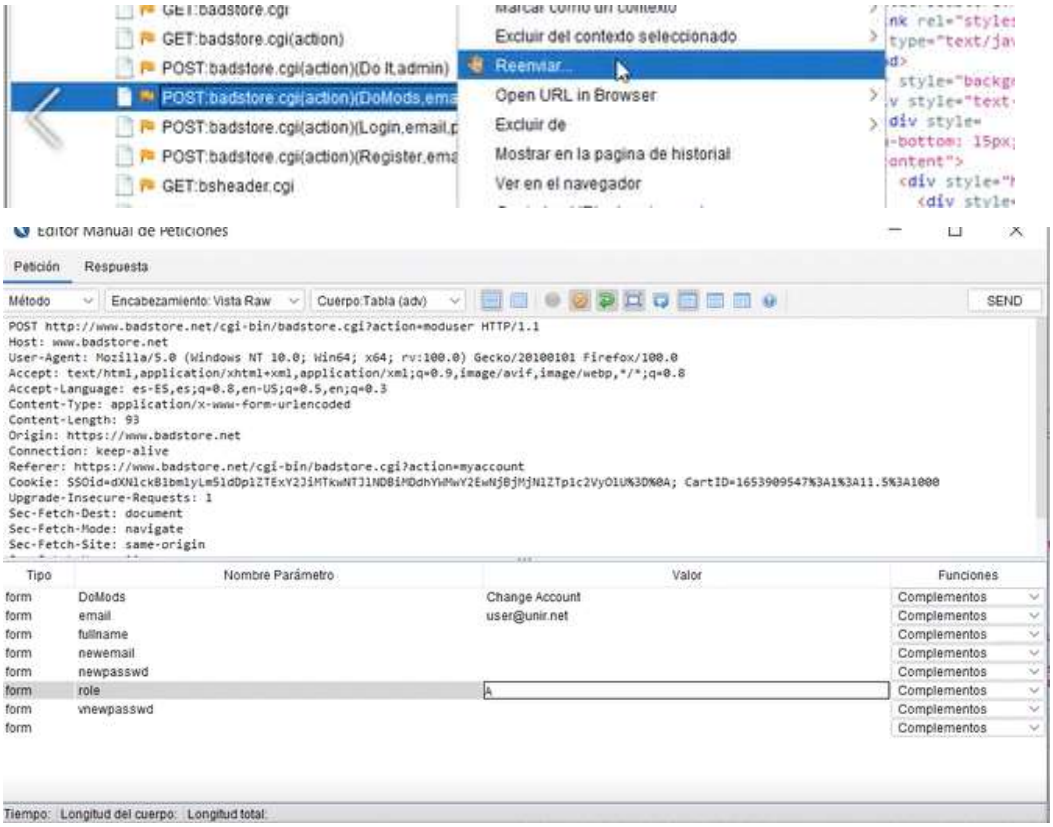


Teniendo este método Post intente reemplazar el valor de role U por A, ya que investigue un poco sobre badstore para el trabajo debido a que los 2 tipos de categoría de usuario son U de normal y A de administrador.



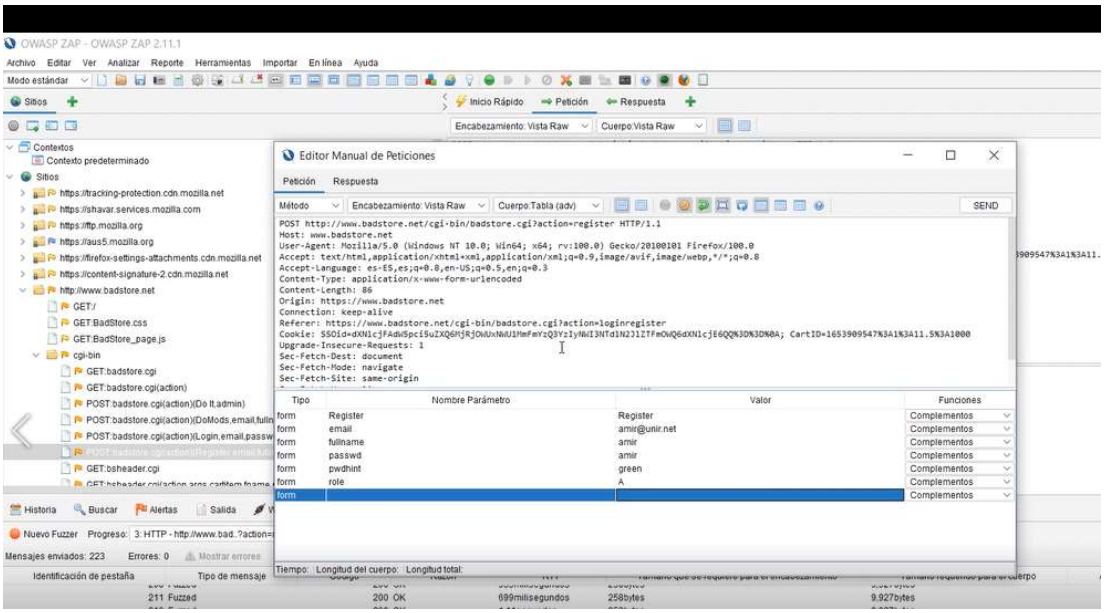
Al intentar hacer fuzzer, no hubo ningún cambio e intente ver por los estados reflejados, lamentablemente fuzzer no me sirvió en ningún contexto.

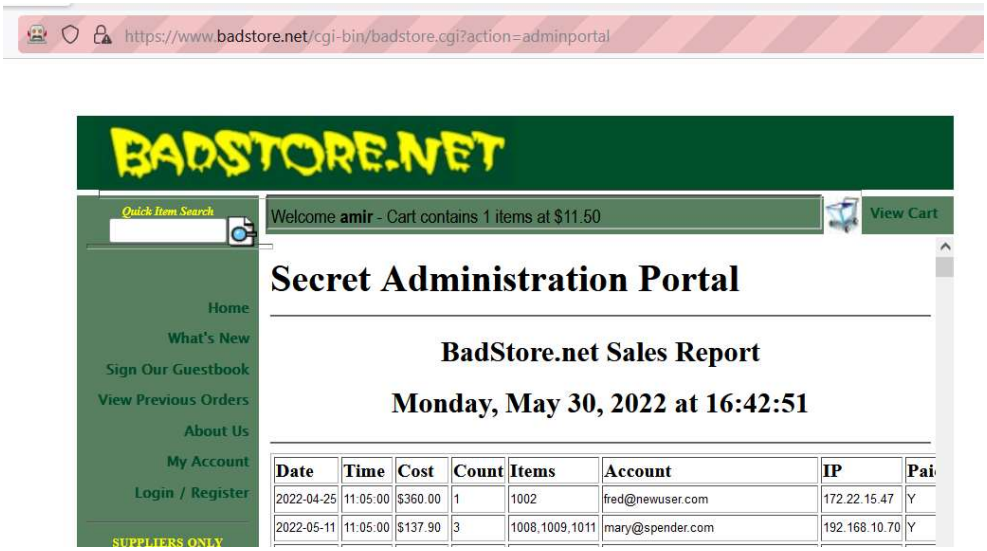
Lo siguiente fue intentarlo con una opción que vi llamada “Reenviar”.



Para probar si el usuario cambio de rol a un administrador se debe ingresar a un menú secreto el cual está en action=admin. Luego de intentar con send, realmente no hubo un cambio con la que intente cambiar el usuario. Por esto al intentar acceder a una opción del menú de admin me manda un error

Lo siguiente fue intentarlo con otra página en esta ocasión fue con la opción Register, ya que al agregar se usaba el role = U lo cambie por role= A

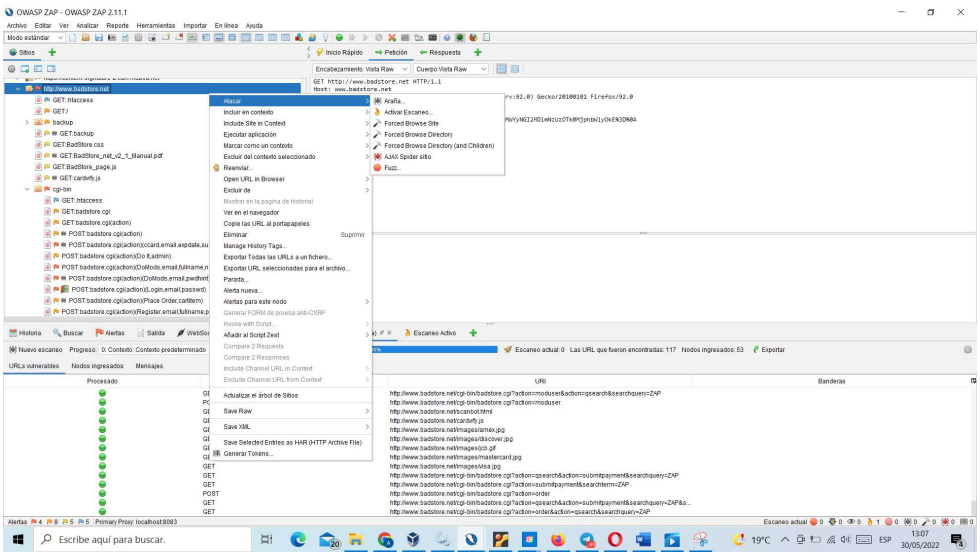




Como se observa en este caso si se puede acceder a cualquiera de las opciones de administrador. Por ende, lo siguiente sería el realizar las 2 actividades adicionales (una es obligatoria, escaneo activo) por hacer según lo que se nos indicó.

Spider

El spider es un descubridor de páginas dentro de un dominio accesos específicos, esto se realiza ya que en algunos casos algunos links son ocultos, pero estos pueden contener información muy importante y deberían ser bloqueados por el administrador. Como se muestra en la siguiente imagen, las páginas descubiertas con el spider aparecen con un símbolo de araña al costado a diferencia del primer escaneo manual que se realizó en el navegador abierto por ZAP

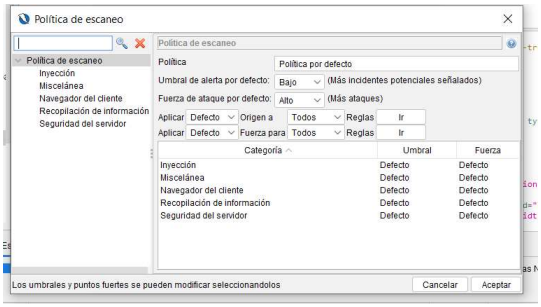


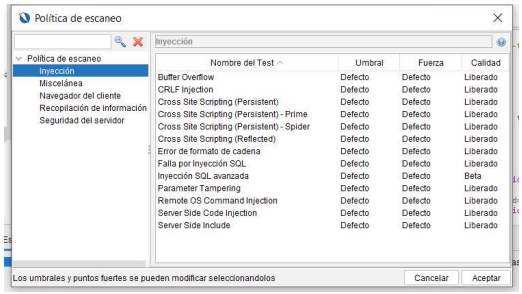
Asignatura	Datos del alumno	Fecha
Seguridad en Aplicaciones Online	Apellidos: Mehrez Garcia Nombre: Amir Fernando Mamdouh	28-05-2022

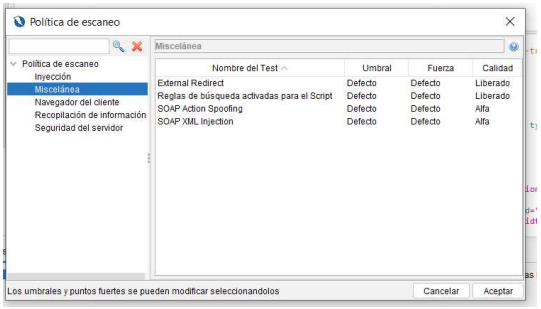
Resultados de vulnerabilidades encontradas

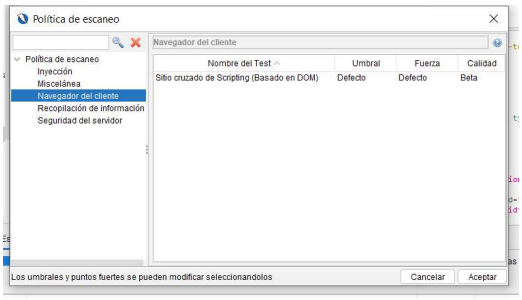
Escaneo Activo

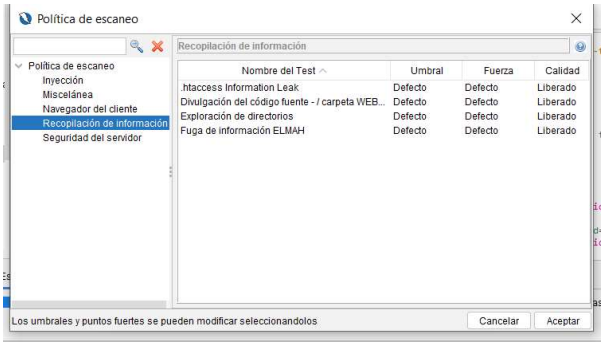
El escaneo activo permite el visualizar las vulnerabilidades a más detalle, ya que se realiza paso por paso en cada una de las paginas obtenidas en los escaneos manuales. Para realizar esto se deben configurar los siguientes menús.



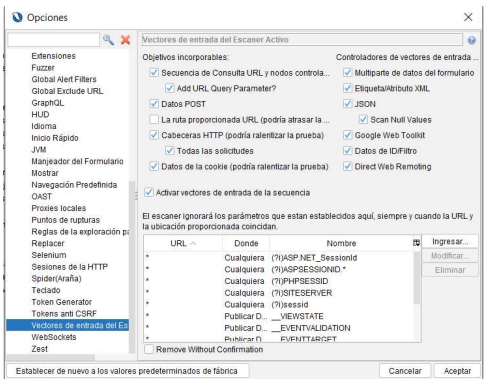


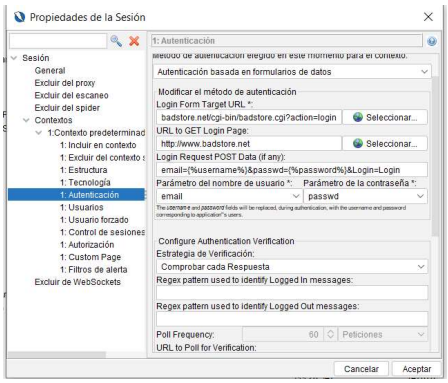






Para los vectores de Entrada seleccione las opciones que muestro y añadí el contexto con el usuario “amir” con el método de autenticación en basado en formulario.

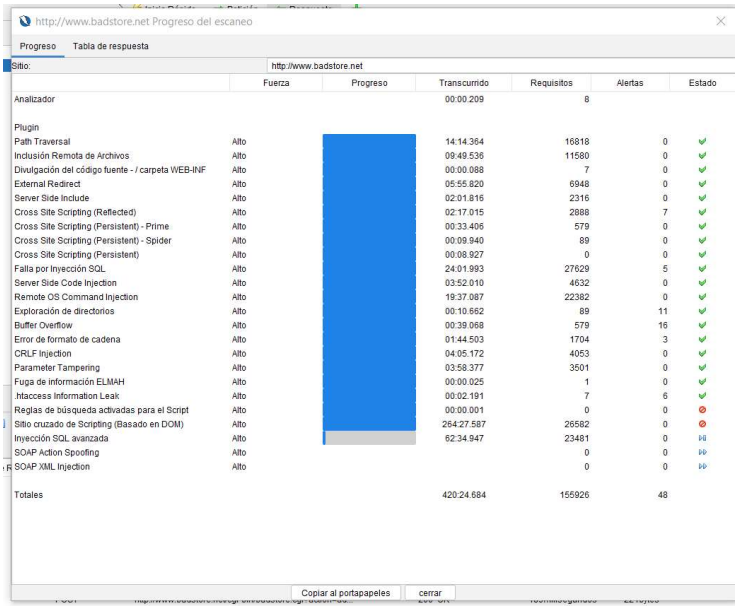




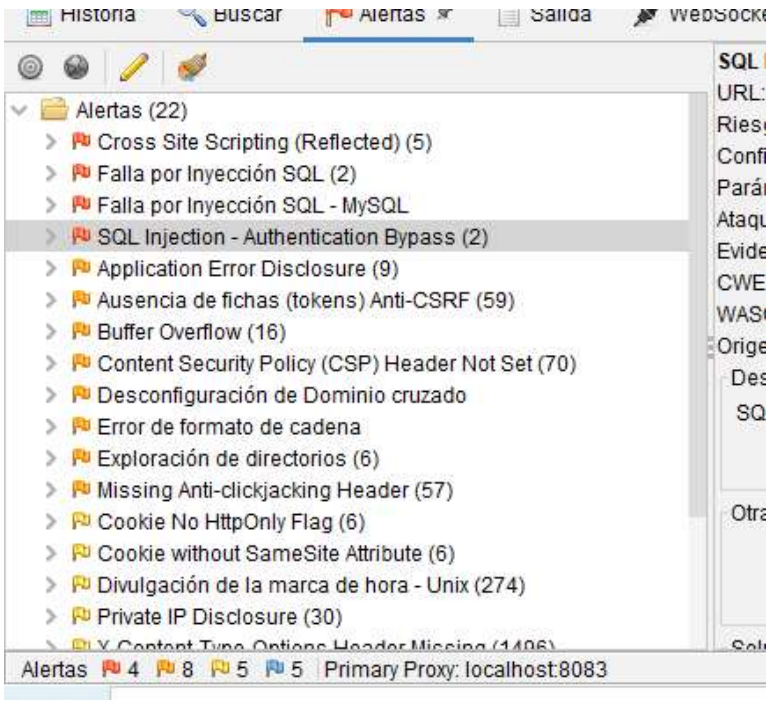
Asignatura	Datos del alumno	Fecha
Seguridad en Aplicaciones Online	Apellidos: Mehrez Garcia	28-05-2022
	Nombre: Amir Fernando Mamdouh	

Es la tercera vez que realizo el escaneo, pero en las 2 primeras tuve un error al intentar hacer fuzz al mismo tiempo, ya que el tiempo me quedo corto daré los resultados como están por que ya lleva más de 7 horas de análisis, que incluso tuve que saltar el XSS basado en DOM al demorar tanto tiempo.

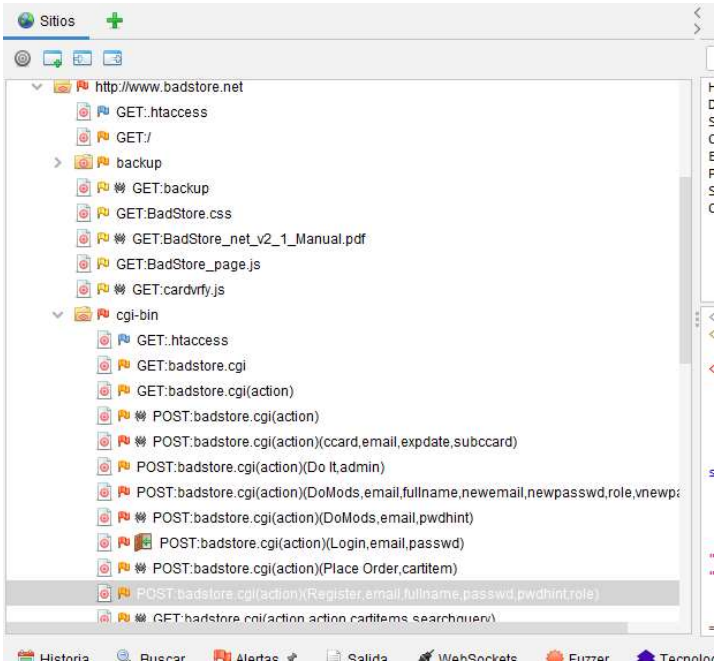
Reportes



Alertas



Sitios



Hasta el momento de la realización de este informe se tienen 22 categorías de alertas, siendo las más relevantes XSS e Inyecciones SQL.

Para la creación de un nuevo usuario administrador lo que hice fue intervenir la petición de Registro de la pagina Register y cambiar el rol, como ya a sido esto mencionado antes.

Adicionalmente anexare los resultados y el reporte generado por la herramienta

Auditoría de las vulnerabilidades encontradas

Al tratar de ejecutar alguno de los ataques de la alerta me volvió a aparecer este error de la bd, error el cual hizo que las 2 primeras veces se trabara el escaneo activo.

Por este motivo simplemente mencionare las 3 vulnerabilidades que debemos auditar, usando la información proporcionada por OWASP ZAP

Asignatura	Datos del alumno	Fecha
Seguridad en Aplicaciones Online	Apellidos: Mehrez Garcia Nombre: Amir Fernando Mamdouh	28-05-2022

Categoria: Cross Site Scripting (Reflected)
Riesgo: High
Parametro: fullname
Descripción: Cross_site Scripting (XSS) es una técnica de ataque que comprende hacer eco del código que fue proporcionado por el atacante en la instancia del navegador de un usuario. Una instancia de navegador puede ser un cliente de navegador web corriente, o un objeto de navegador integrado e un producto de software, como el navegador que se encuentra dentro de WinAmp, un lector de RSS o un cliente de correos electrónicos. El código por sí mismo se encuentra escrito en HTML/JavaScript, pero también puede extenderse a VBScript, ActiveX, Jave, Flash o cualquier otra tecnología que sea compatible con el navegador.
Solución: **Fases: Implementación; Arquitectura y Diseño** **Comprenda el contexto en el que se va a utilizar sus datos y la condificación que se va a esperar. Esto es fundamentalmente importante cuando se transmiten los datos entre diferentes componentes o cuando se generan las salidas que pueden comprender múltiples codificaciones al mismo tiempo, como paginas web o mensajes de correos de varias zonas. Estudie todos los protocolos de comunicacón y representaciones de los datos que son esperadas para poder determinar las estrategias de codificación que son necesarias.** **Por cualquier dato que se enviará a otra página web, en especial cualquier dato recibido de las entradas externas, utiice la codificación que sea conveniente en todos los caracteres que no sean alfanuméricos.** **Consulte la hoja de referencia de prevención de CSS para poder obtener más información detallada de los diferentes tipos de condificación y escape que se requieren.** **Fase: Arquitectura y Diseño** **Cualquier comprobación de seguridad que se vaya a realizar en el lado del cliente, asegúrese de que estas comprobaciones se encuentre duplicadas en el lado del servidor, para evitar el CWE-602. Los atacantes pueden eludir las comprobaciones**

del lado del cliente modificando los valores después de que se hayan realizado las comprobaciones, o cambiando al cliente para poder eliminar de forma completa las comprobaciones del lado del cliente. Después, estos valores que fueron modificados serán enviados al servidor.

Si se encuentra disponible, utilice los mecanismos estructurados que apliquen de forma automática la separación entre los datos y códigos. Estos mecanismos pueden otorgar la cotización, codificación y validación relevantes de manera automática, en lugar de confiar en que el desarrollador proporcione esta capacidad en cada uno de los puntos donde se origina la salida.

Fase: Implementación
Para cada una de las páginas web que se origina, utilice y especifique una codificación de caracteres como ISO-8859 o UTF-8. Cuando no se puede especificar una condificación, el navegador web podría selaccionar una codificación distinta adivinando que codificiación está siendo utilizada en verdad por la página web. Esto puede permitir que el navegador web trate varias secuencias como especiales, abriendo al cliente a leves ataques XSS. Consulte CWE-116 para conseguir más mitigaciones con respecto a la codificación/escape.

Para ayudar a mitigar los ataques XSS contra las cookies de la sesión del usuario, es necesario establecer que la cookie de la sesión sea HttpOnly. En navegadores que son compatibles con la característica HttpOnly (como las versiones más actualizadas de internet explorer y firefox), esta característica puede prevenir que la cookie de sesión del usuario sea accesible para las secuencias de comandos del lado del cliente malignas que utilizan document.cookie. Esta no es una solución muy completa, ya que HttpOnly no es compatible con todos los navegadores que hay. Más importante aún, XMLHTTPRequest y otras tecnologías poderosas de navegador otorgan acceso de lectura a los encabezados HTTP, incluido el encabezado Set-Cookie en el cual se establece el indicador HttpOnly.

Asuma que toda la entrada es maliciosa. Use an "accept known good" input validation strategy, i.e., use an allow list of acceptable inputs that strictly conform to specifications. Rechace cualquier entrada que no se adapte de forma estricta a las especificaciones, o cambielas por algo que sí lo haga. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a deny list). However, deny lists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright.

Al realizar la validación de entrada, usted debe considerar todas las propiedades potencialmente destacadas, incluida la longitud, el tipo de entrada, el rango completo de valores aceptables, las entradas faltantes o adicionales, la sintaxis, el sentido entre los campos que se encuentran relacionados y la conformidad con todas las reglas comerciales. As an example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it is not valid if you are expecting colors such as "red" or "blue."

Ensure that you perform input validation at well-defined interfaces within the application. Esto ayudará a cuidar la aplicación, incluso si un elemento se utiliza de nuevo o traslada a otro sitio.

Parte de Código:

<HTML><HEAD><TITLE>BadStore.net - Update User Information</TITLE> </HEAD><BODY><H2> Account Information for: </H2> Full Name: <scrIpt>alert(1);</scRipt><P> Email: ZAP<P> Password: ZAP<P><H3> Has been updated!</H3><HR><Center><FONT SIZE=2, FACE='Times'>BadStore v2.1.2 - Copyright © 2003-2006</Center>

Asignatura	Datos del alumno	Fecha
Seguridad en Aplicaciones Online	Apellidos: Mehrez Garcia	28-05-2022
	Nombre: Amir Fernando Mamdouh	

Categoria: Falla por Inyeccion SQL
Riesgo: High
Parametro: email
Descripción: Inyección SQL puede ser posible. **Otra Info: The page results were successfully manipulated using the boolean conditions [foo-bar@example.com' AND '1'='1' --] and [foo-bar@example.com' OR '1'='1' --]** **The parameter value being modified was NOT stripped from the HTML output for the purposes of the comparison** **Data was NOT returned for the original parameter**
Solución: **Do not trust client side input, even if there is client side validation in place.** **In general, type check all data on the server side.** **If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?'** **If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries.** **If database Stored Procedures can be used, use them.** **Do *not* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality!** **Do not create dynamic SQL queries using simple string concatenation.** **Escape all data received from the client.** **Apply an 'allow list' of allowed characters, or a 'deny list' of disallowed characters in user input.** **Apply the principle of least privilege by using the least privileged database user possible.** **In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact.** **Grant the minimum database access that is necessary for the application.**
Parte de Código:

Asignatura	Datos del alumno	Fecha
Seguridad en Aplicaciones Online	Apellidos: Mehrez Garcia	28-05-2022
	Nombre: Amir Fernando Mamdouh	

Ya que consiste en el login, solo debería cambiar el nombre de usuario secuestrando la petición.

Categoria: Application Error Disclosure

Riesgo: Medium

Evidencia: Parent Directory

Descripción: This page contains an error/warning message that may disclose sensitive information like the location of the file that produced the unhandled exception. This information can be used to launch further attacks against the web application. The alert could be a false positive if the error message is found inside a documentation page.

Solución:

Review the source code of this page. Implement custom error pages. Consider implementing a mechanism to provide a unique error reference/identifier to the client (browser) while logging the details on the server side and not exposing them to the user.

Parte de Código:

```
<HTML>
 <HEAD>
  <TITLE>Index of /ws</TITLE>
 </HEAD>
 <BODY>
<H1>Index of /ws</H1>
<PRE><IMG SRC="/icons/blank.gif" ALT="     "> <A HREF="?N=D">Name</A>                    <A HREF="?M=A">Last modified</A>
<HR>
<IMG SRC="/icons/back.gif" ALT="[DIR]"> <A HREF="/">Parent Directory</A>        30-May-2022 11:05      -
<IMG SRC="/icons/unknown.gif" ALT="[    ]"> <A HREF="NetBadstoreSoapSearch.java">NetBadstoreSoapSearc..&gt;</A> 09-Nov-2006 14:51
<IMG SRC="/icons/unknown.gif" ALT="[    ]"> <A HREF="NetBadstoreSoapSearchLocator.java">NetBadstoreSoapSearc..&gt;</A> 09-Nov-200
<IMG SRC="/icons/unknown.gif" ALT="[    ]"> <A HREF="NetBadstoreSoapSearchSoap.java">NetBadstoreSoapSearc..&gt;</A> 09-Nov-2006 1
<IMG SRC="/icons/unknown.gif" ALT="[    ]"> <A HREF="NetBadstoreSoapSearchSoapStub.java">NetBadstoreSoapSearc..&gt;</A> 09-Nov-20
<IMG SRC="/icons/p.gif" ALT="[    ]"> <A HREF="soapSearchClient.pl">soapSearchClient.pl</A>     09-Nov-2006 14:46     1k
<IMG SRC="/icons/p.gif" ALT="[    ]"> <A HREF="wsdlSearchClient.pl">wsdlSearchClient.pl</A>     09-Nov-2006 14:46     1k
</PRE><HR>
<ADDRESS>Apache/1.3.28 Server at 192.168.56.110 Port 80</ADDRESS>
</BODY></HTML>
```

Asignatura	Datos del alumno	Fecha
Seguridad en Aplicaciones Online	Apellidos: Mehrez Garcia Nombre: Amir Fernando Mamdouh	28-05-2022

Actividad 2 (valor real: 5 puntos)	Descripción	Puntuación máxima (puntos)	Peso %
Criterio 1	Como se ha llevado a cabo el procedimiento de test	3	30%
Criterio 2	Resultados de vulnerabilidades encontradas	3	30%
Criterio 3	Auditoría de las vulnerabilidades encontradas	3	30%
Criterio 4	Calidad de la memoria	1	10%
		10	**100 %**