**Tarea 4 Test de penetración**

Presentado Por:

Jose Fernando Ararat Moreno

Presentado a:

Mag. Cesar Antonio Villamizar

Universidad Nacional abierta y a distancia-UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería

Information Security

2024

**Objetivos**

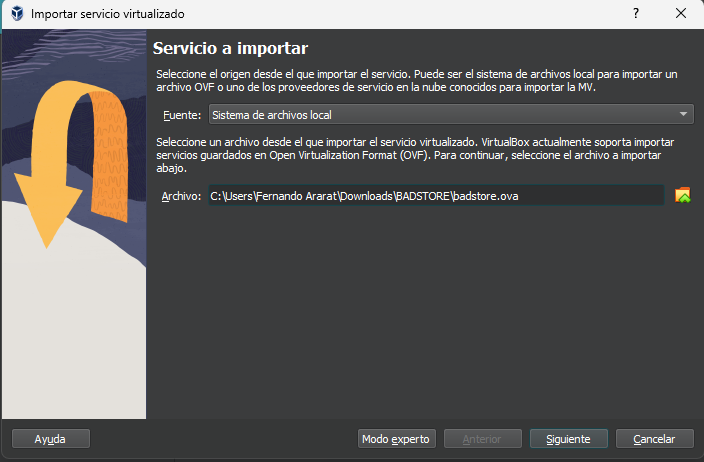
* Formular medidas de mitigación de riesgos de seguridad de la información en productos de software de acuerdo con metodologías, técnicas y buenas prácticas de desarrollo seguro.
* Hacer una revisión de las lecturas correspondientes a la unidad 3 que se encuentran en el entorno de aprendizaje.
* Realizar un test de penetración a la aplicación web BADSTORE.
* Descargar e instalar ZAP.
* Descargar la máquina virtual con la aplicación BADSTORE.
* Importar el servicio virtualizado.
* Configurar la máquina de VirtualBox asociada con BADSTORE.
* Realizar un test de penetración de la aplicación BADSTORE con el scanner de vulnerabilidades ZAP atacando el nombre asociado a la dirección del dispositivo.
* Auditar manualmente tres vulnerabilidades para comprobar la veracidad de las alertas por parte de ZAP e indicar la forma de como mitigarla.
* Guardar el informe de la herramienta ZAP en formato HTML.

**Enlace de la presentación**

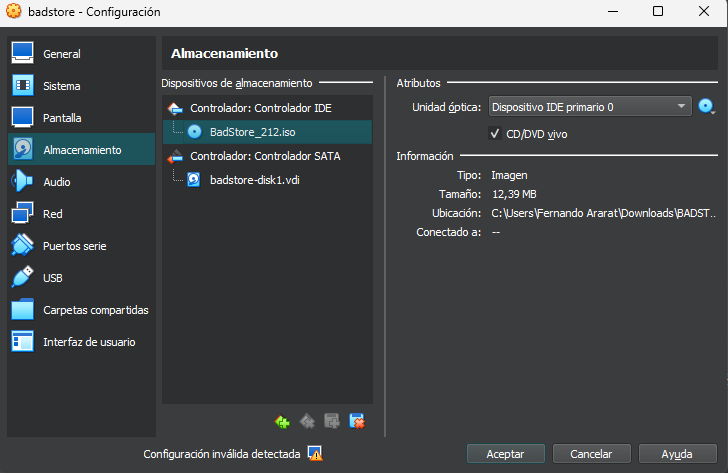
<https://unadvirtualedu-my.sharepoint.com/:p:/g/personal/mnceronh_unadvirtual_edu_co/EXDIEhS9299OpO9w-t0-hrMBvb8TCwy-NHDjD9XvD5nL0Q?e=dWKFjl>

**Trabajo Individual**

Configuración máquina virtual.

****

Importación del .ova de BADSTORE a la máquina virtual.

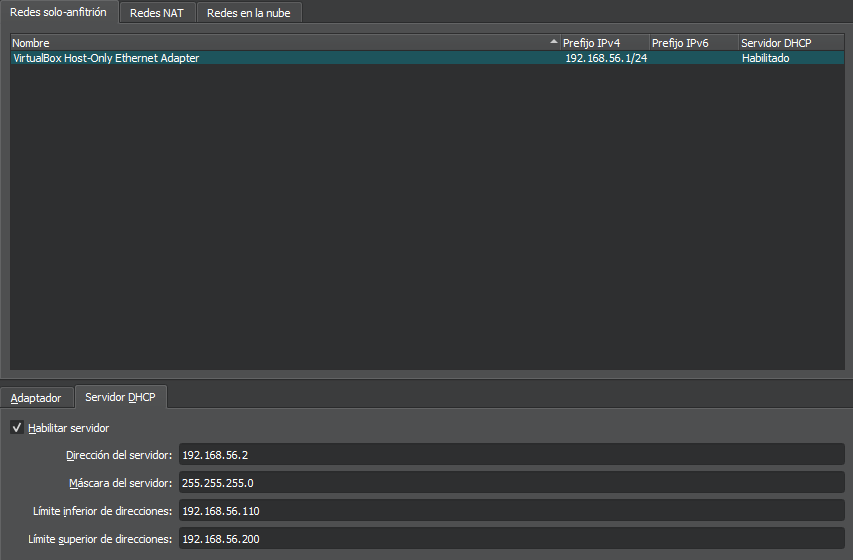


Cargar imagen ISO de BADSTORE en la máquina virtual.

Configuración de red de la maquina BADSTORE.

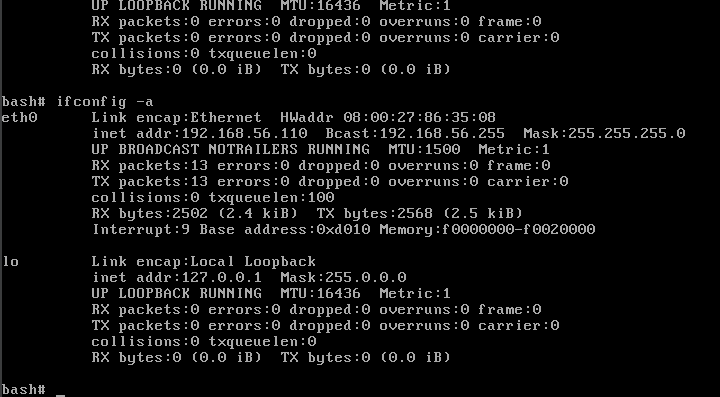


Configuración del adaptador.



Configuración servidor DHCP

Inicio de BADSTORE en virtualBox



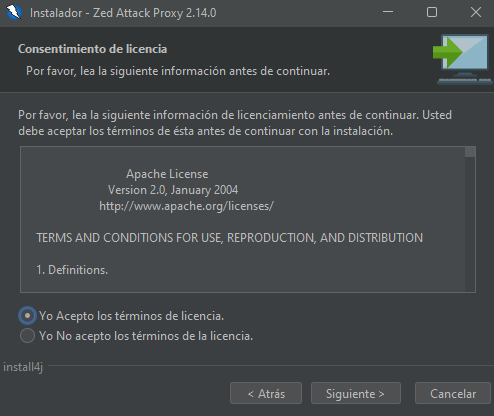
Dirección IP de la maquina

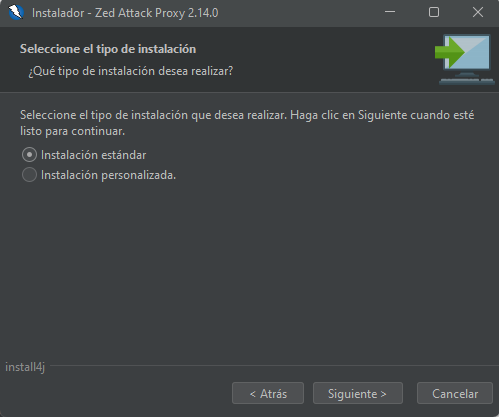


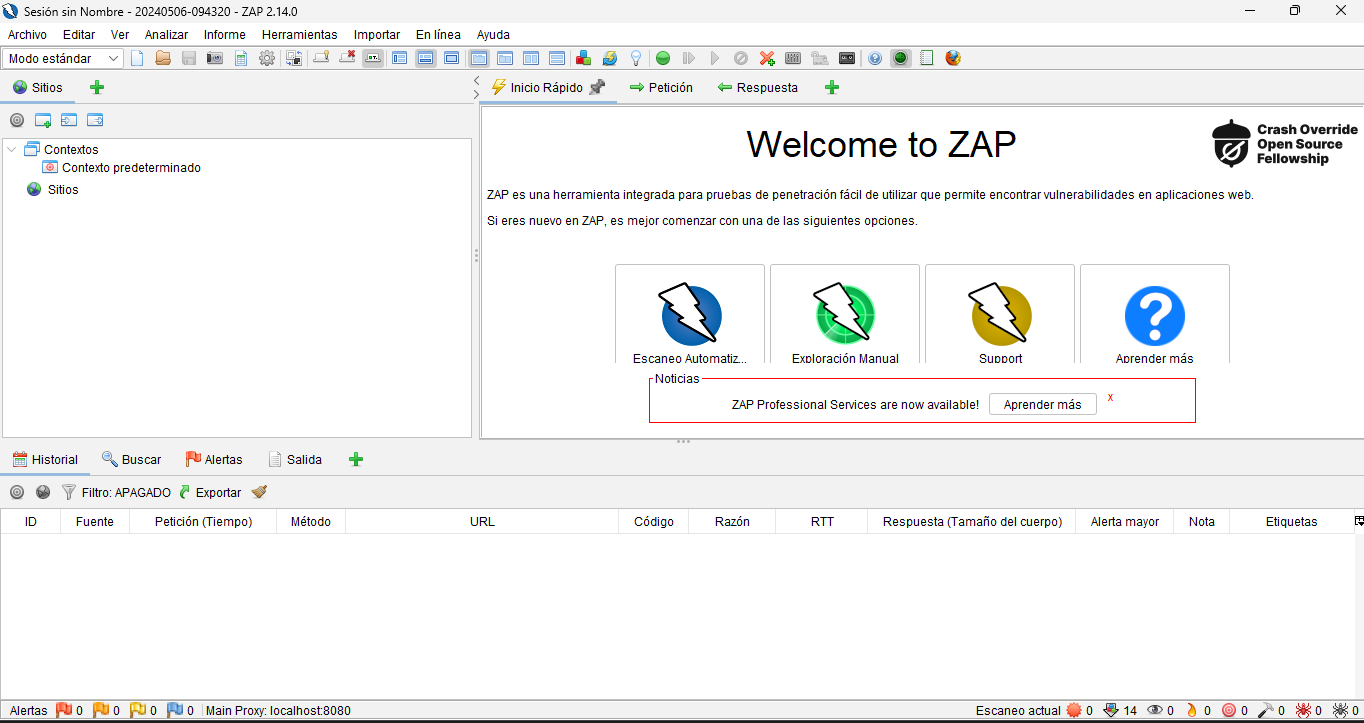
Modificación del archivo HOST donde se agregó la dirección IP 192.168.56.110 y www.badstore.net

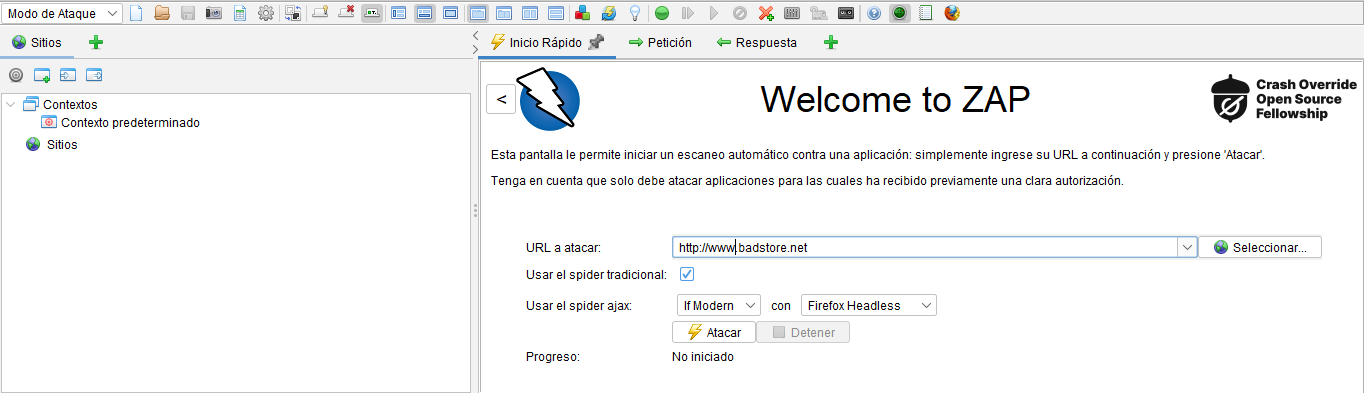
**Instalación de OWASP ZAP**



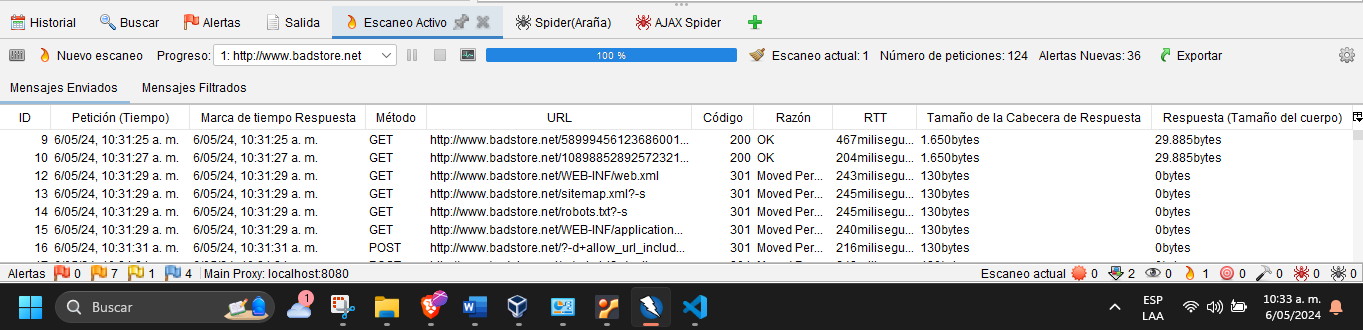




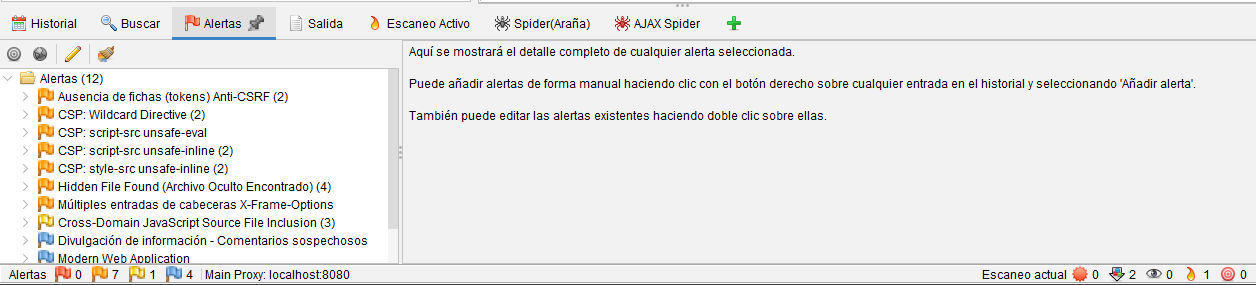




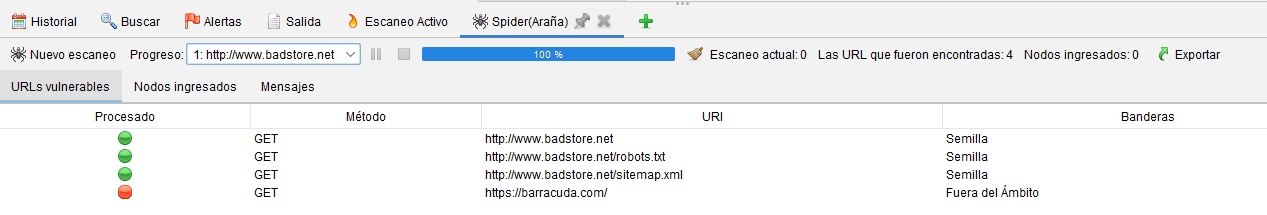
Se configura el ZAP en modo de ataque y se pega la URL [www.badstore.net](http://www.badstore.net) y le damos clic en atacar.



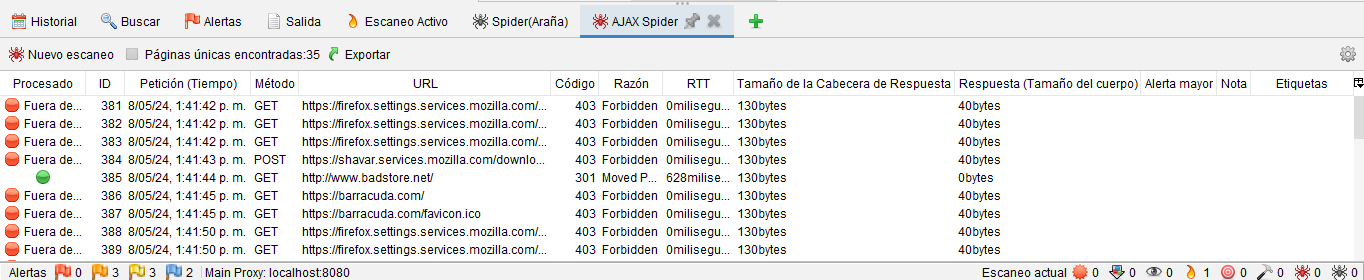
Aquí encuentra todas las vulnerabilidades de la pagina



Spider(Araña)



AJAX Spider



**Vulnerabilidades**

1. **CSP: Wildcard Directive**

* Riesgo: medio
* Confianza: alta
* Descripción: La Política de Seguridad de Contenidos (CSP) es una capa adicional de seguridad que ayuda a detectar y mitigar ciertos tipos de ataques. Entre ellos, los ataques de secuencias de comandos en sitios cruzados (XSS) y de inyección de datos. Estos ataques se utilizan para todo, desde el robo de datos hasta la desfiguración de sitios o la distribución de malware. CSP proporciona un conjunto de cabeceras HTTP estándar que permiten a los propietarios de sitios web declarar las fuentes de contenido aprobadas que los navegadores deberían poder cargar en esa página: los tipos cubiertos son JavaScript, CSS, marcos HTML, fuentes, imágenes y objetos incrustables como applets Java, ActiveX y archivos de audio y vídeo.
* Solución: Asegúrese de que su servidor web, servidor de aplicaciones, equilibrador de carga, etc. está correctamente configurado para establecer el encabezado Content-Security-Policy.
* Limitar el uso de comodines.
* Revisar y validar la configuración.
* Aplicar el principio de menor privilegio.
* Implementar controles de acceo adicionales.
* Monitorear y registrar actividades.
* Mantener actualizados los sistemas y servicios.
* Realizar pruebas de seguridad.

1. **Hidden File Found (Archivo Oculto Encontrado)**

* Riesgo: medio
* Confianza: bajo
* Descripción: Se identificó un archivo confidencial como accesible o disponible. Esto puede filtrar información administrativa, de configuración o de credenciales que puede ser aprovechada por un individuo malintencionado para atacar más adelante el sistema o mejorar la manera en que realiza ataques de ingeniería social.
* Solución: Considera si este componente es realmente necesario en producción; si no es así, desactívelo. Si es así, asegurar que el acceso requiera la autenticación y autorización adecuadas, o limita la exposición solo a sistemas internos o IPs de origen definidas, etc.
* Revisión y eliminación de archivos ocultos innecesarios.
* Políticas de visibilidad de archivos.
* Gestión de permisos de archivo.
* Monitoreo de cambios en archivos ocultos.
* Educación y concienciación del usuario.
* Implementar políticas de seguridad de acceso.
* Actualizaciones y parches de seguridad.

1. **Múltiples entradas de cabeceras X-Frame-Options**

* Riesgo: medio
* Confianza: medio
* Descripción: Se encontraron encabezados X-Frame-Options (XFO), una respuesta con múltiples entradas de cabeceras XFO puede no ser tratada de manera predecible por todos los user-agent.
* Solución: Asegúrese de que sólo haya una cabecera X-Frame-Options en la respuesta.
* Eliminar duplicados de cabecera.
* Configuración adecuada de la cabecera.
* Revisión de la configuración del servidor web.
* Pruebas de seguridad.
* Educación sobre seguridad web.

**Referencias Bibliográficas**

* Ramachandran, M. (2012). [Code Security: Best-practice guidelines and examples](https://bibliotecavirtual.unad.edu.co/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=602994&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_135). En Nova (Eds), Software Security Engineering: Design and Applications (pp. 135-148). Nova Science Publishers, Inc. <https://bibliotecavirtual.unad.edu.co/login?url=http://search.ebscohost.com/login.as> px?direct=true&db=nlebk&AN=602994&lang=es&site=eds[1]live&scope=site&ebv=EB&ppid=pp\_135
* OWASP (2020). Web Application Penetration Testing. <https://owasp.org/www-project-web-security-testing-guide/latest/3-The_OWASP_Testing_Framework/1-Penetration_Testing_Methodologies>
* OWASP (2021). TOP TEN. <https://owasp.org/Top10/es/>
* Ramachandran, M. (2012). Software Security Testing. En Nova (Eds), Software Security Engineering : Design and Applications (pp. 151-164). Nova Science Publishers, Inc. https://bibliotecavirtual.unad.edu.co/login?url=http://search.ebscohost.com/login.as px?direct=true&db=nlebk&AN=602994&lang=es&site=eds[1]live&scope=site&ebv=EB&ppid=pp\_151
* Bermejo, J. R. (2014). Metodología de evaluación de herramientas de análisis automático de seguridad de aplicaciones web para su adaptación en el ciclo de vida de desarrolloOpen this document with ReadSpeaker docReader . Madrid: UNED. http://e-spacio.uned.es/fez/eserv/tesisuned:IngInd[1]Jrbermejo/BERMEJO\_HIGUERA\_Juan\_Ramon\_Tesis.pdf
* Ramachandran, M. (2012). Design for software security. En Nova (Eds), Software Security Engineering : Design and Applications (pp. 101-112). Nova Science Publishers, Inc. <https://bibliotecavirtual.unad.edu.co/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=602994&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_101>
* Marmolejo, P.A. (2021). Seguridad en las fases del S-SDLC. https://repository.unad.edu.co/handle/10596/41639