Ingeniería en Desarrollo de Software

Actividad 3 Cross Site Scripting (XSS).

Auditoria Informática.

Ingeniería en Desarrollo de Software.

TUTOR: Jessica Hernández Romero.

ALUMNO: Homero Ramirez Hurtado.

FECHA: 22 de Febrero del2025.

Índice.

Introducción.

El Cross-Site Scripting (XSS) es una vulnerabilidad de seguridad en aplicaciones web que permite a los atacantes inyectar scripts maliciosos en páginas vistas por otros usuarios. Estos scripts pueden ejecutarse en el navegador del usuario, permitiendo a los atacantes robar información confidencial, como cookies de sesión, credenciales de inicio de sesión y otros datos sensibles. XSS se considera una de las principales amenazas de seguridad web y se encuentra en el ranking OWASP Top Ten, una lista de las diez vulnerabilidades más críticas en aplicaciones web.

Existen tres tipos principales de XSS: almacenado, reflejado y basado en DOM.

1. **XSS almacenado:** También conocido como persistente, ocurre cuando los scripts maliciosos se almacenan permanentemente en el servidor de destino, como en una base de datos, y se ejecutan cada vez que un usuario accede a la página afectada.

2. **XSS reflejado:** Este tipo de ataque ocurre cuando los scripts maliciosos se envían a un servidor web y se reflejan en la respuesta al usuario. A diferencia del XSS almacenado, los scripts no se almacenan en el servidor y solo se ejecutan cuando un usuario hace clic en un enlace malicioso.

3. **XSS basado en DOM:** Se produce cuando los scripts maliciosos manipulan el Document Object Model (DOM) en el navegador del usuario. Este tipo de ataque no requiere comunicación con el servidor y se basa en la forma en que el navegador procesa y muestra el contenido de la página.

Para prevenir XSS, es fundamental implementar prácticas de codificación segura, como validar y sanitizar la entrada de los usuarios, utilizar Content Security Policy (CSP) y evitar la inclusión de contenido no confiable en las páginas web. Adoptar estas medidas puede ayudar a proteger las aplicaciones web y la información de los usuarios contra este tipo de ataques.

Descripción.

Una auditoría informática para detectar y mitigar vulnerabilidades de Cross-Site Scripting (XSS) es un proceso exhaustivo que incluye diversas etapas y técnicas para evaluar la seguridad de una aplicación web. A continuación se describe cómo funcionaría este tipo de auditoría y las medidas para prevenir XSS:

1. **Planeación y Recolección de Información:** El auditor recopila información sobre la aplicación, incluyendo su arquitectura, tecnologías utilizadas y posibles puntos de entrada de datos. Esta etapa también implica revisar la documentación y entrevistar a los desarrolladores para entender mejor el funcionamiento interno de la aplicación.

2. **Análisis Estático del Código:** Se revisa el código fuente de la aplicación para identificar posibles vulnerabilidades de XSS. Se utilizan herramientas automatizadas de análisis de código estático que buscan patrones comunes de vulnerabilidades, como la falta de sanitización y validación de datos de entrada.

3. **Análisis Dinámico:** El auditor simula ataques XSS en la aplicación en un entorno controlado. Utiliza herramientas de pruebas de penetración para inyectar scripts maliciosos en diferentes puntos de entrada y observa cómo se comporta la aplicación. Esta etapa ayuda a identificar vulnerabilidades que no se detectaron en el análisis estático.

4. **Revisión del DOM y del lado del Cliente:** Se examina cómo la aplicación maneja y manipula el Document Object Model (DOM) en el navegador del usuario. El auditor busca

posibles manipulaciones de scripts maliciosos que puedan ejecutarse en el navegador sin interacción con el servidor.

5. **Informe de Resultados y Recomendaciones:** El auditor compila un informe detallado con los hallazgos, destacando las vulnerabilidades de XSS encontradas y proporcionando recomendaciones específicas para corregirlas.

Para evitar la vulnerabilidad de XSS, se deben seguir las siguientes prácticas:

- **Validación y Sanitización de Datos:** Validar y sanitizar todos los datos de entrada del usuario para eliminar o neutralizar los scripts maliciosos.
- **Uso de Content Security Policy (CSP):** Implementar una política de seguridad de contenido que restrinja la ejecución de scripts no confiables.
- **Codificación de Salida:** Codificar adecuadamente todos los datos antes de mostrarlos en la página web para evitar que los scripts maliciosos se ejecuten.
- **Pruebas de Seguridad Continuas:** Realizar pruebas de seguridad periódicas para identificar y corregir nuevas vulnerabilidades a medida que evolucionan las aplicaciones.

Implementar estas medidas puede ayudar a proteger las aplicaciones web contra ataques de XSS y garantizar la seguridad de la información de los usuarios.

Justificación.

Burp Suite Community Edition es una herramienta gratuita y poderosa para realizar pruebas de seguridad en aplicaciones web, y es especialmente útil para identificar y explotar vulnerabilidades de Cross-Site Scripting (XSS). Su utilización en actividades de auditoría y pruebas de penetración se justifica por diversas razones clave que refuerzan su eficacia y accesibilidad.

Primero, Burp Suite Community Edition ofrece una interfaz intuitiva y fácil de usar, lo que facilita la curva de aprendizaje para nuevos usuarios y permite a los auditores y desarrolladores centrarse en la identificación de vulnerabilidades sin necesidad de dominar herramientas complicadas. La capacidad de interceptar y modificar solicitudes y respuestas HTTP permite a los usuarios explorar cómo se gestionan los datos en la aplicación y detectar posibles puntos de inyección de scripts maliciosos.

Segundo, la herramienta incluye funciones de escaneo manual y automatizado que pueden identificar vulnerabilidades de XSS al analizar el contenido dinámico de la aplicación web. Los usuarios pueden inyectar diferentes scripts de prueba en los campos de entrada y observar cómo la aplicación maneja y refleja estos datos, lo que es fundamental para detectar XSS reflejado y basado en DOM.

Además, Burp Suite Community Edition es altamente configurable, lo que permite a los usuarios ajustar las configuraciones de prueba según las necesidades específicas de la aplicación. Esto incluye la capacidad de crear y ejecutar extensiones personalizadas que amplían las funcionalidades de la herramienta y mejoran la eficiencia de las pruebas de seguridad.

Por último, al ser una herramienta ampliamente reconocida y utilizada en la comunidad de seguridad informática, Burp Suite Community Edition cuenta con una gran cantidad de recursos, tutoriales y documentación en línea que pueden ayudar a los usuarios a maximizar su potencial en la detección y prevención de vulnerabilidades de XSS.

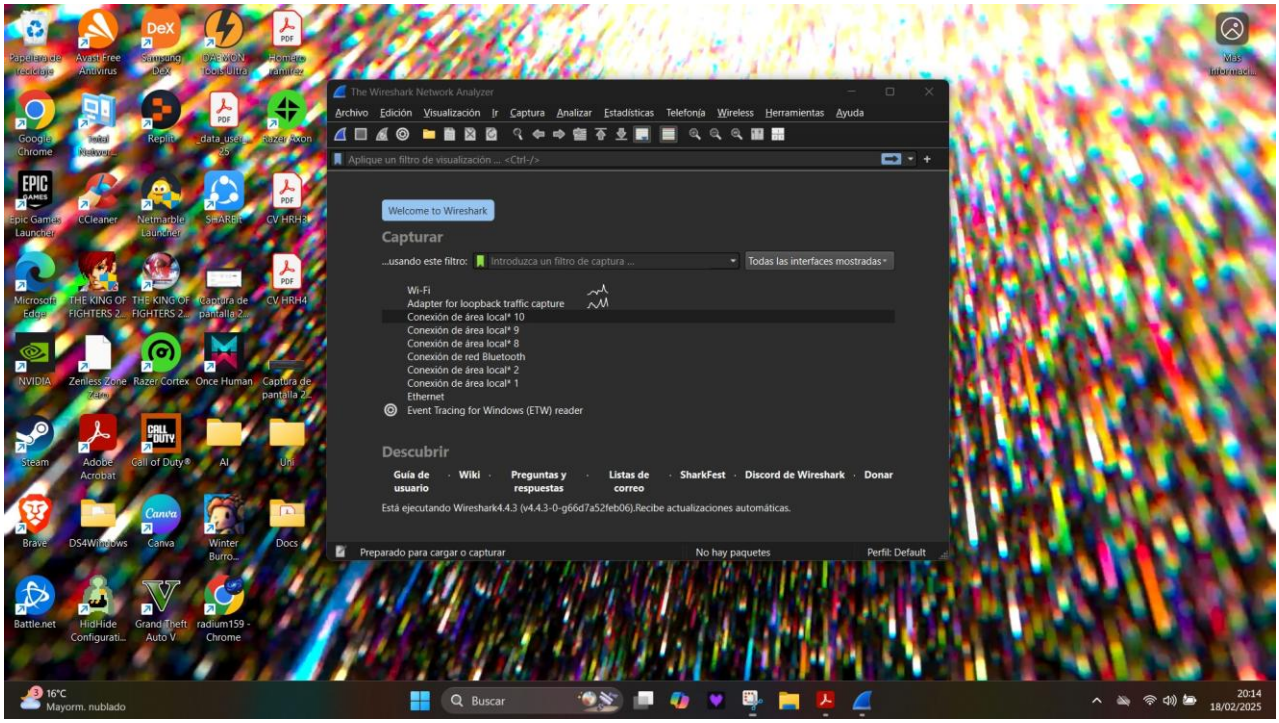
En resumen, la utilización de Burp Suite Community Edition para realizar pruebas de XSS está justificada por su accesibilidad, funcionalidades robustas, y amplio respaldo de la comunidad, lo que la convierte en una herramienta valiosa para garantizar la seguridad de las aplicaciones web.

Etapa 1.

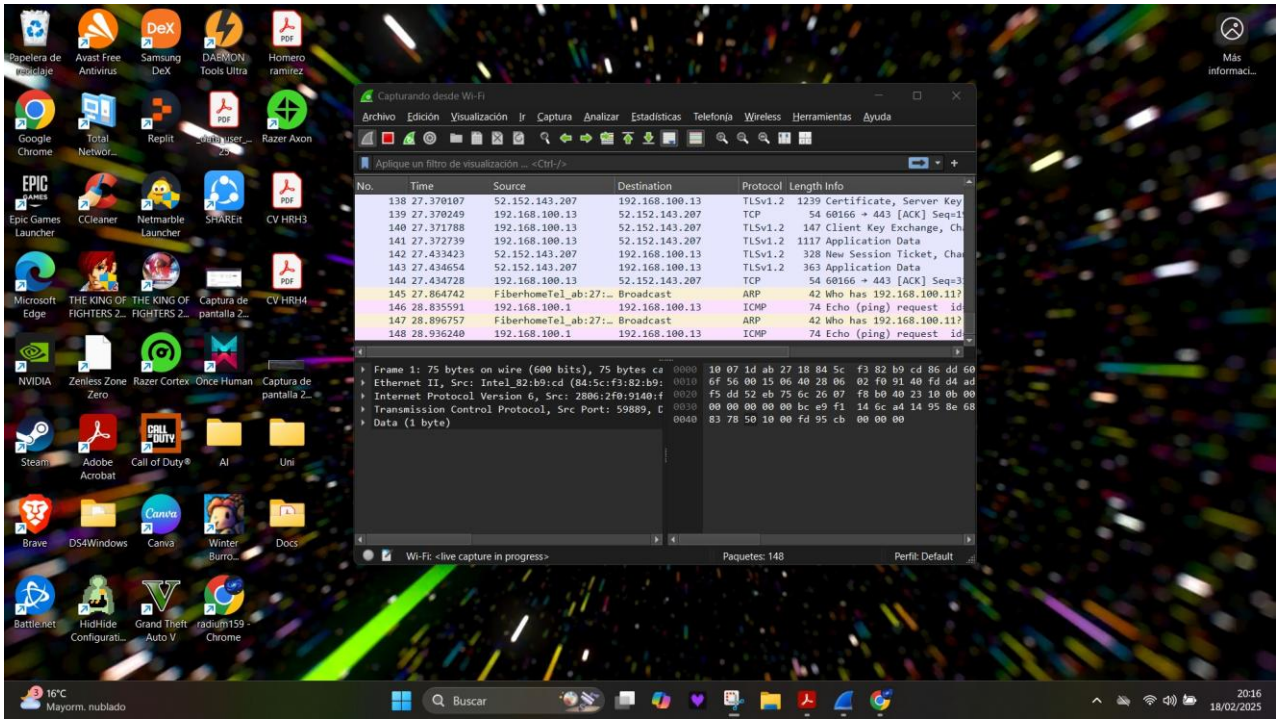
Descripción del Sitio Web.

Hostinger es una plataforma en línea que ofrece servicios de alojamiento web y dominios a precios asequibles. Fundada en 2004, Hostinger se ha convertido en uno de los proveedores líderes en el mercado, con más de 3 millones de usuarios en más de 150 países. Su creador de sitios web permite a los usuarios construir sitios web sin necesidad de conocimientos técnicos avanzados. Ofrece más de 150 plantillas personalizables, un editor de arrastrar y soltar, y funciones eCommerce integradas. Además, cuenta con herramientas de marketing e inteligencia artificial nativas, configuración de posicionamiento SEO, certificado SSL gratuito y guardado automático de versiones web.

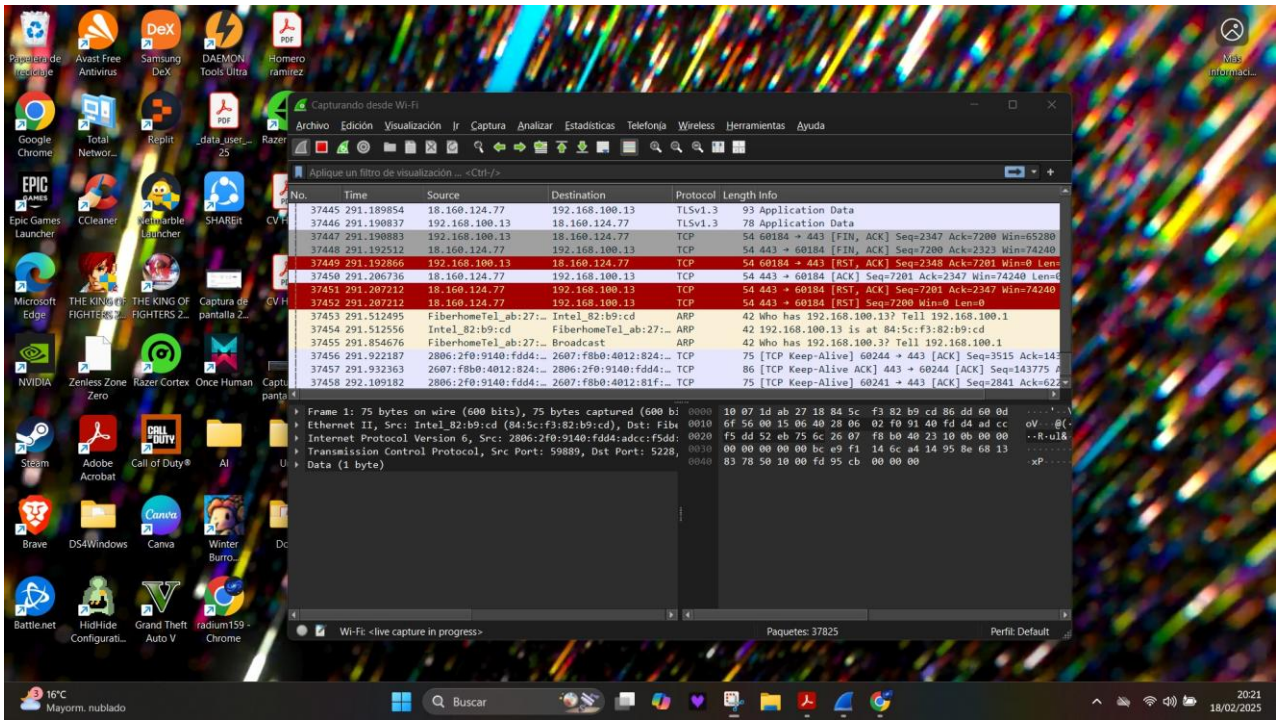
Ataque al Sitio.



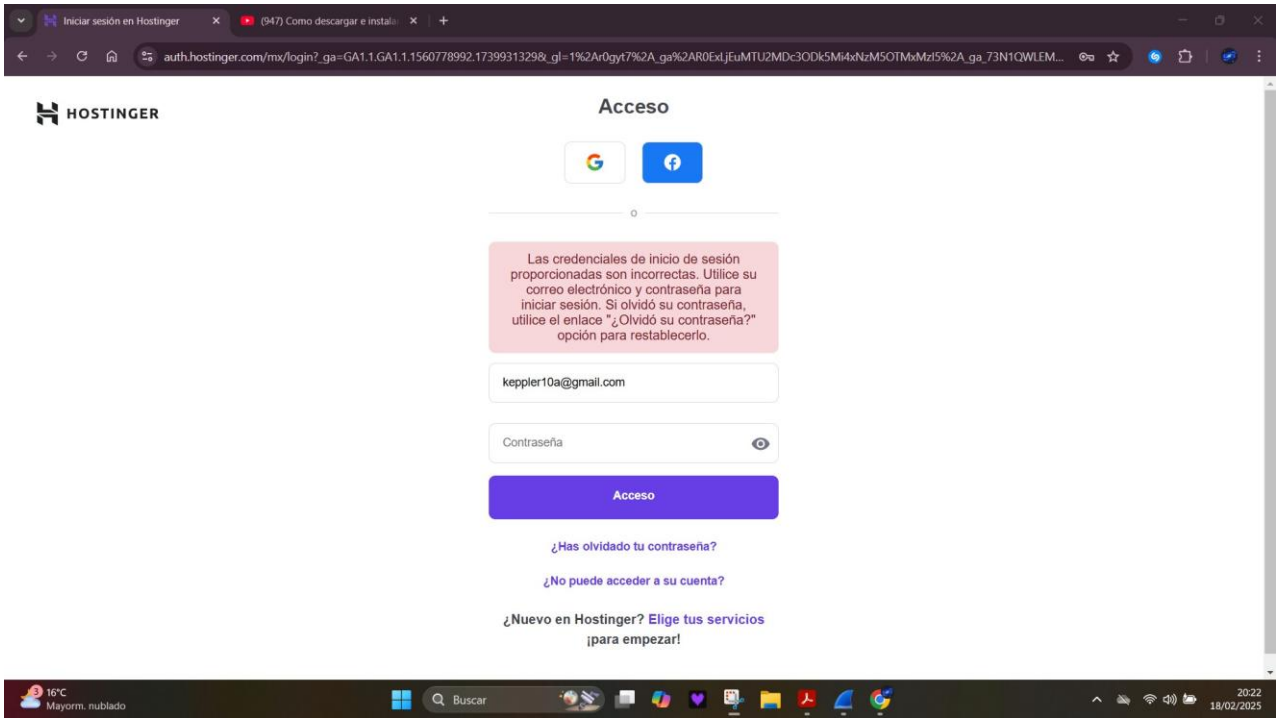
Se inicia WireShark.



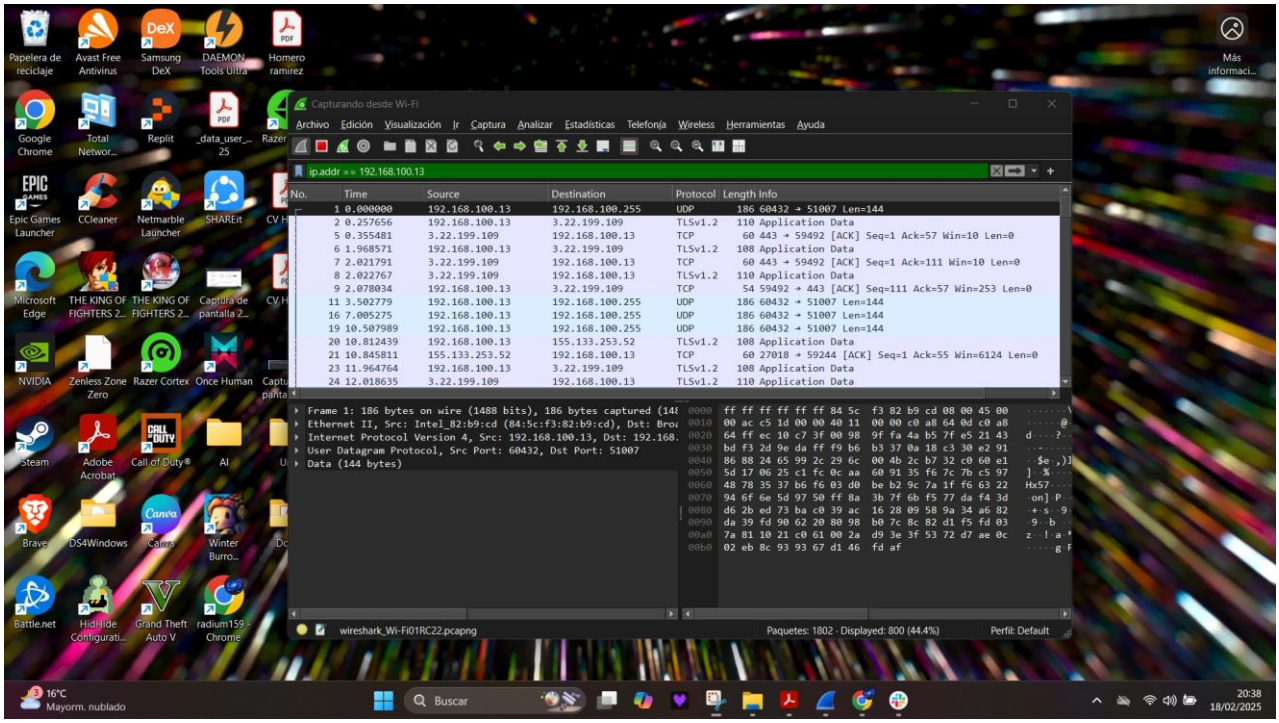
Se ingresa en la opción de WiFi.



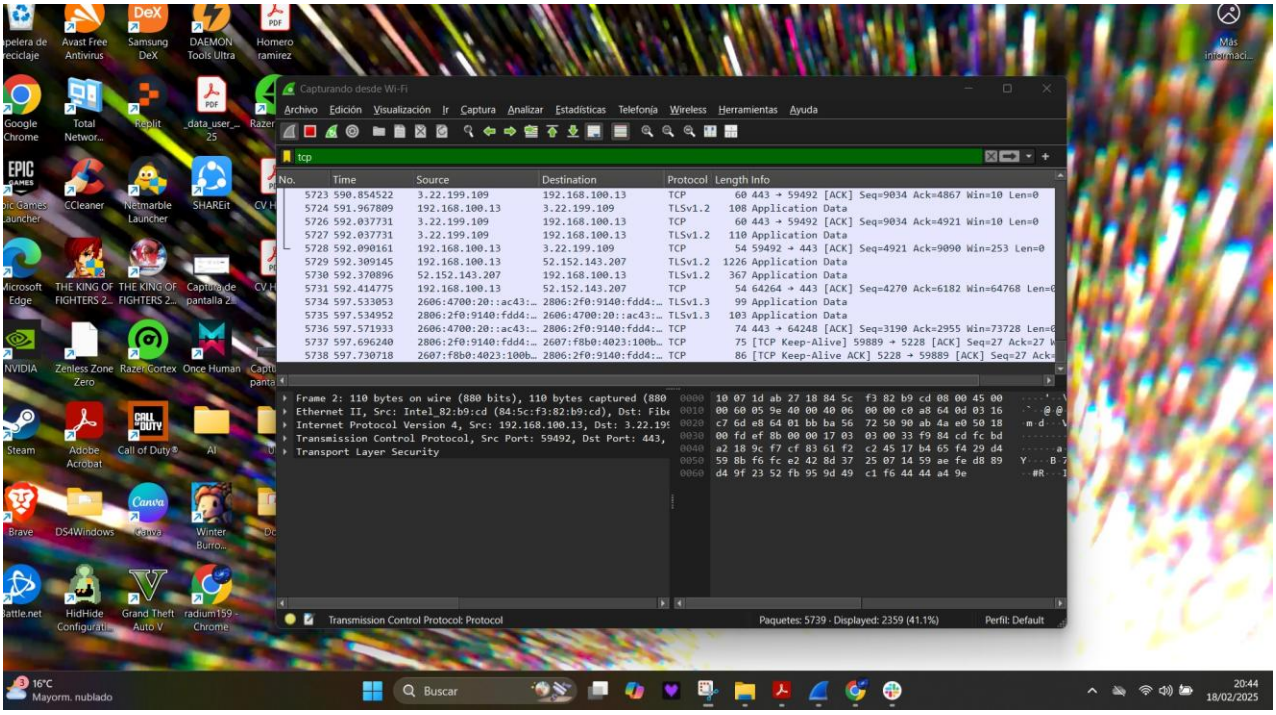
Se observan los detalles.



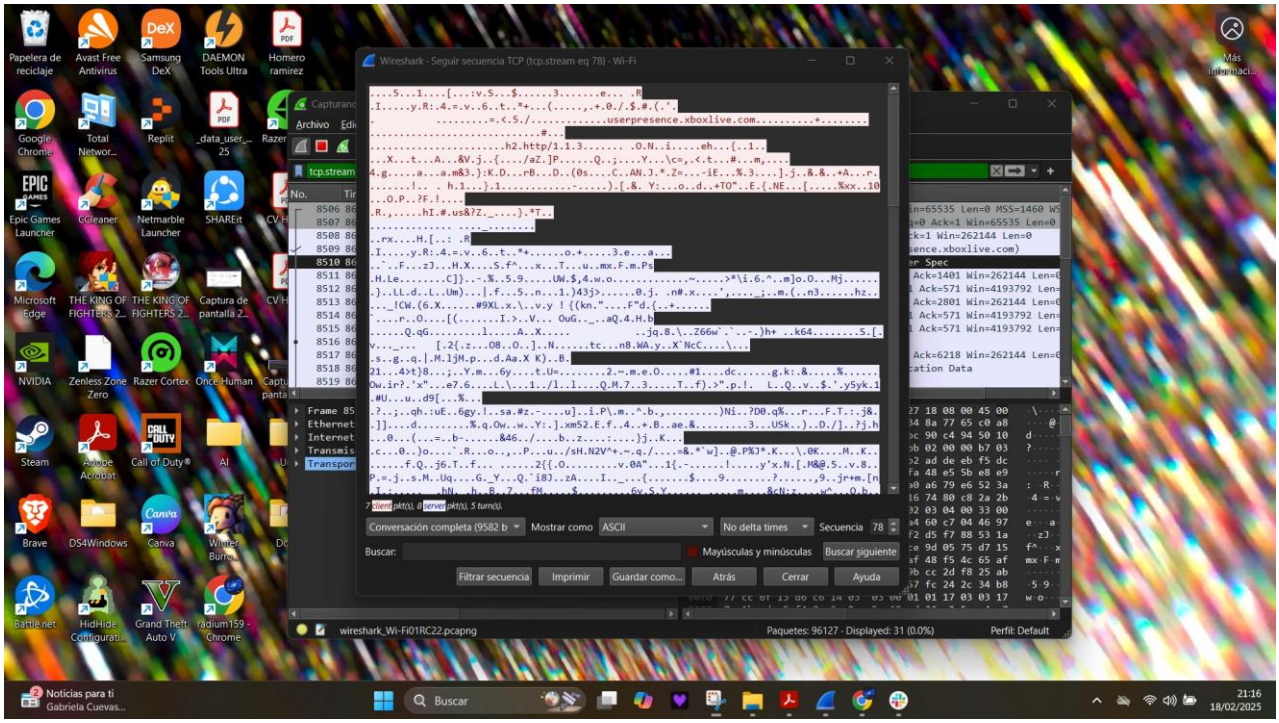
Se elige sitio web.



Se ingresa direccion ip.



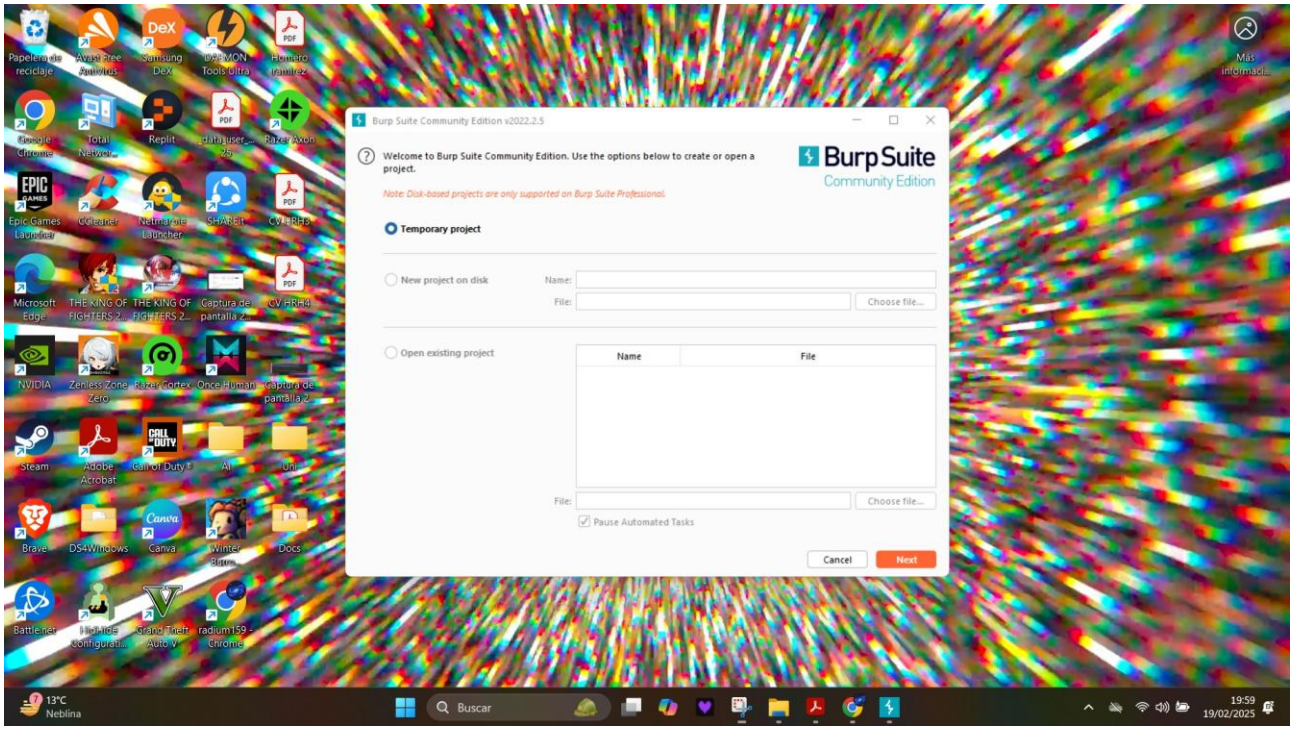
Se procede al ataque.



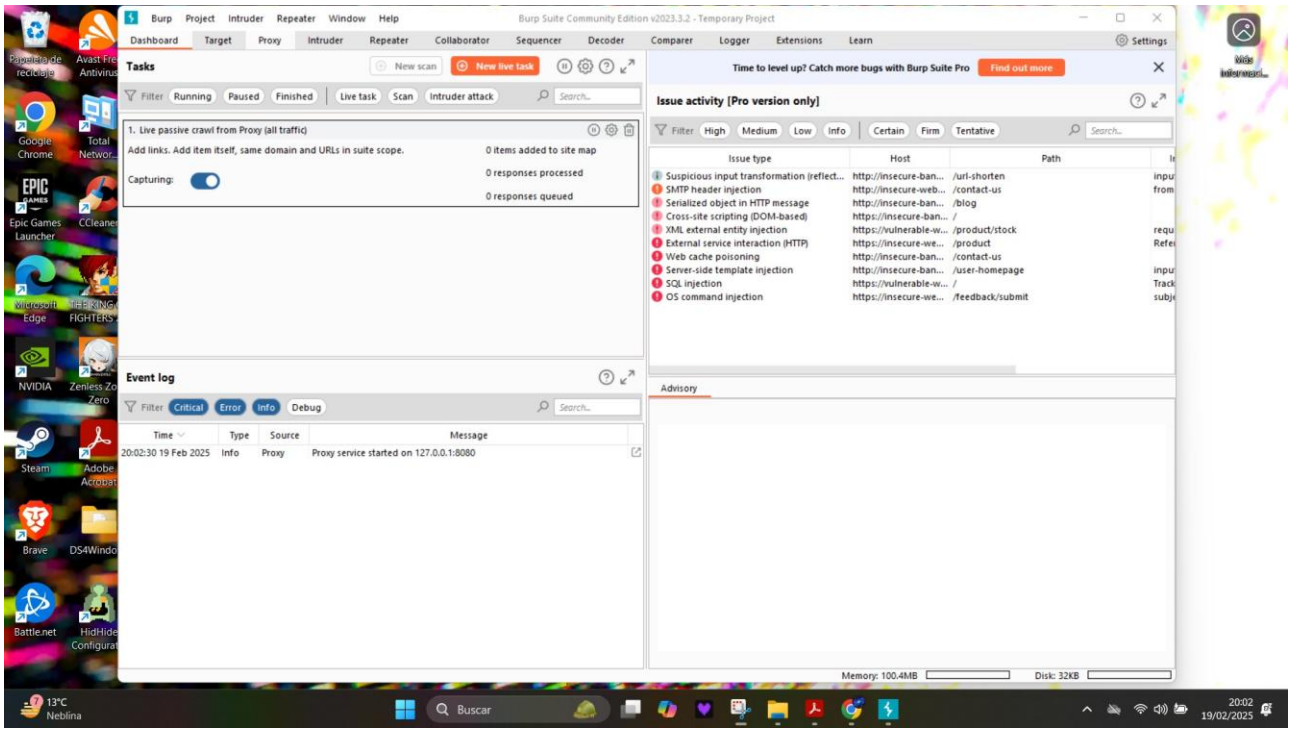
Resultado final del ataque.

Etapa 2.

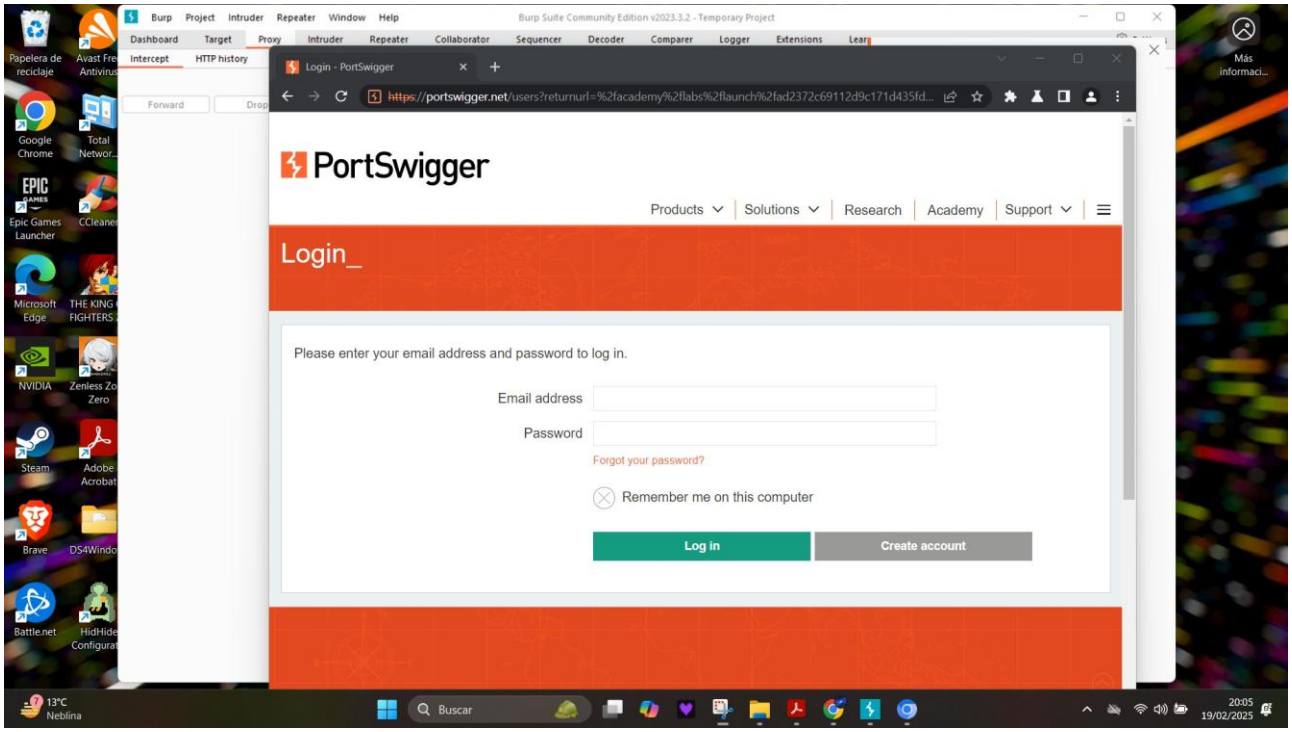
Ataque al Sitio.



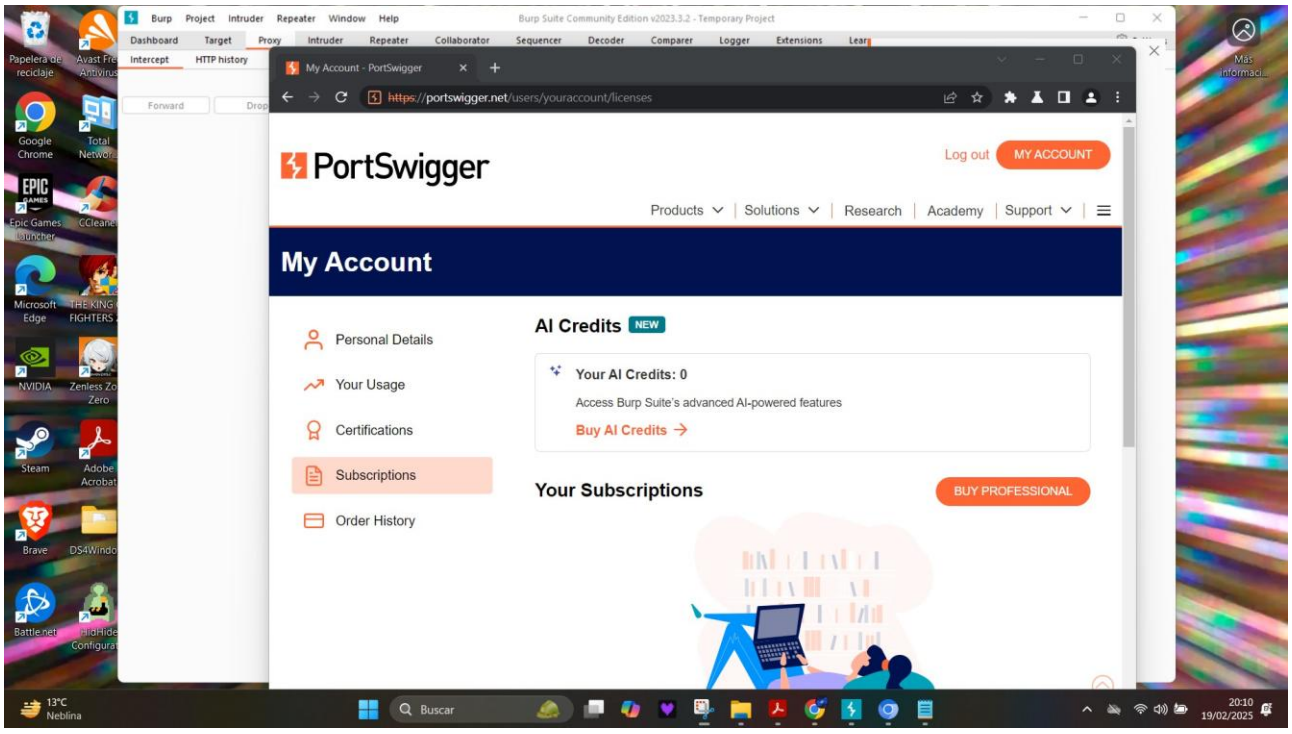
Inicio del software Burp Suite.



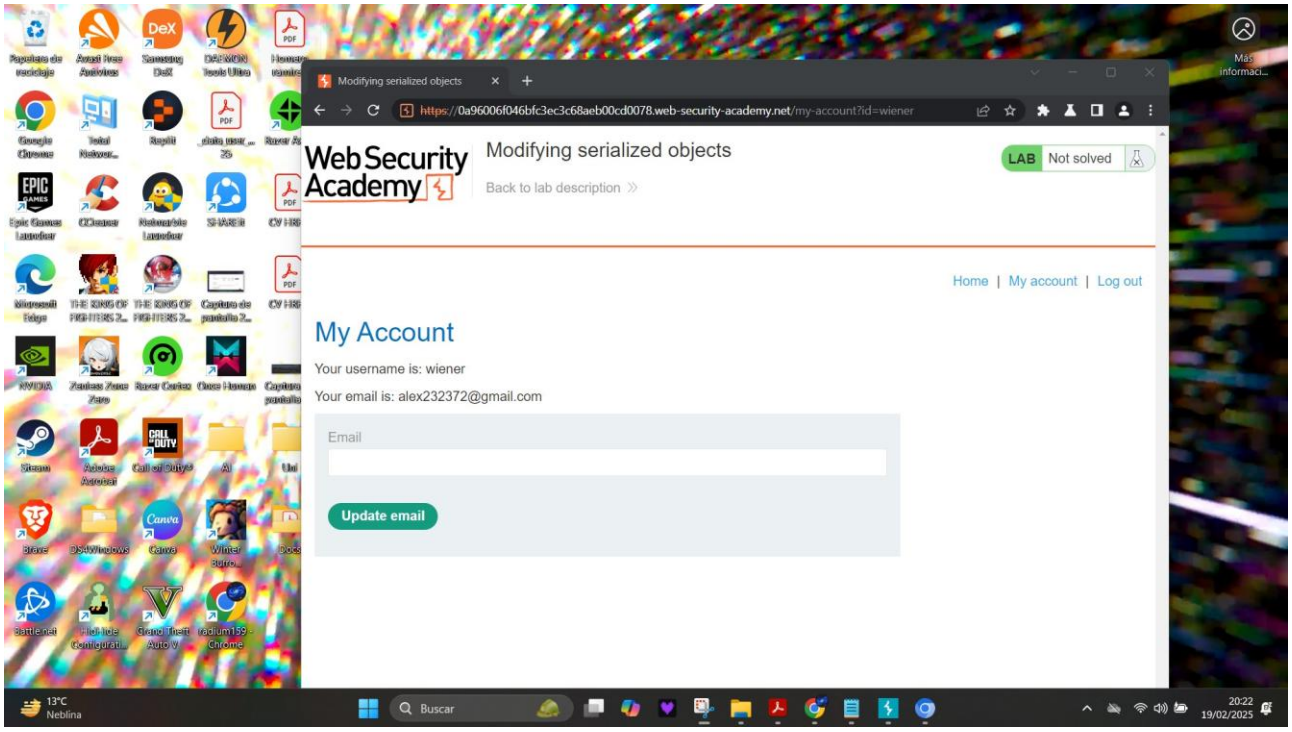
Pantalla de inicio del proyecto.



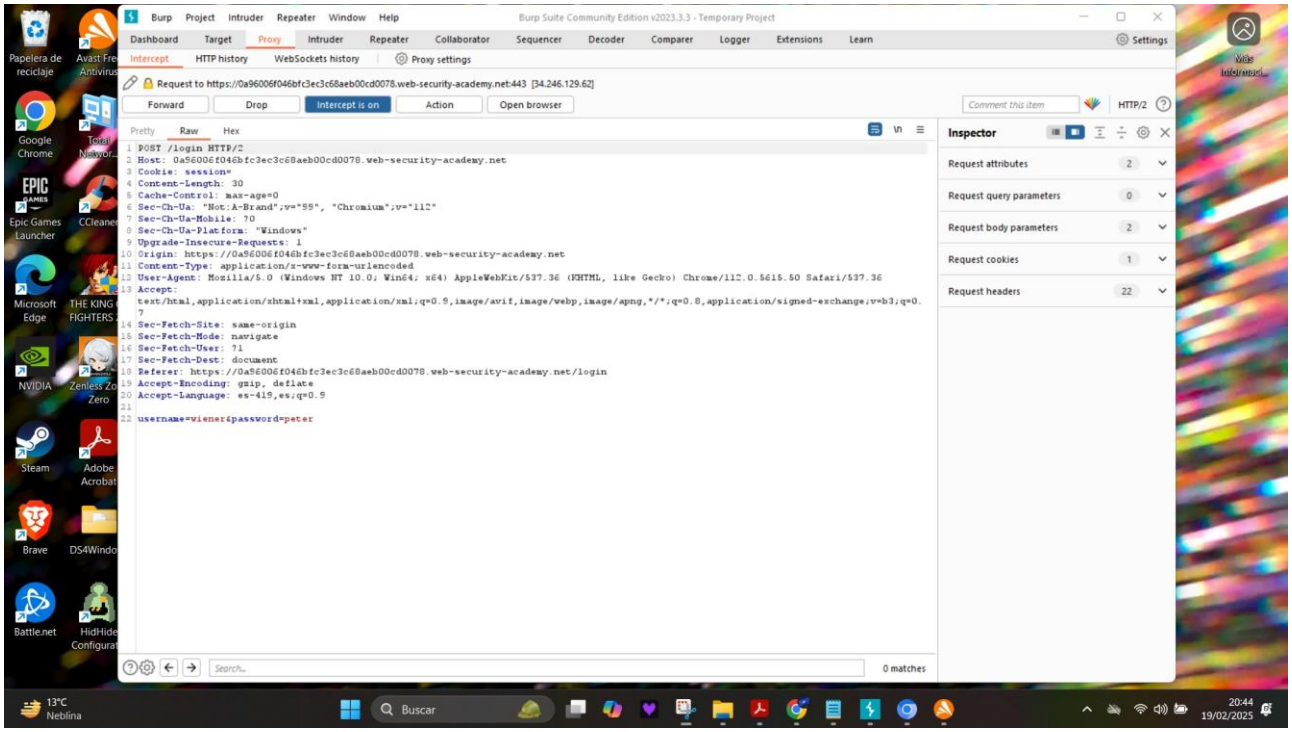
Se inicia el navegador con el enlace del laboratorio.



Se crea cuenta.

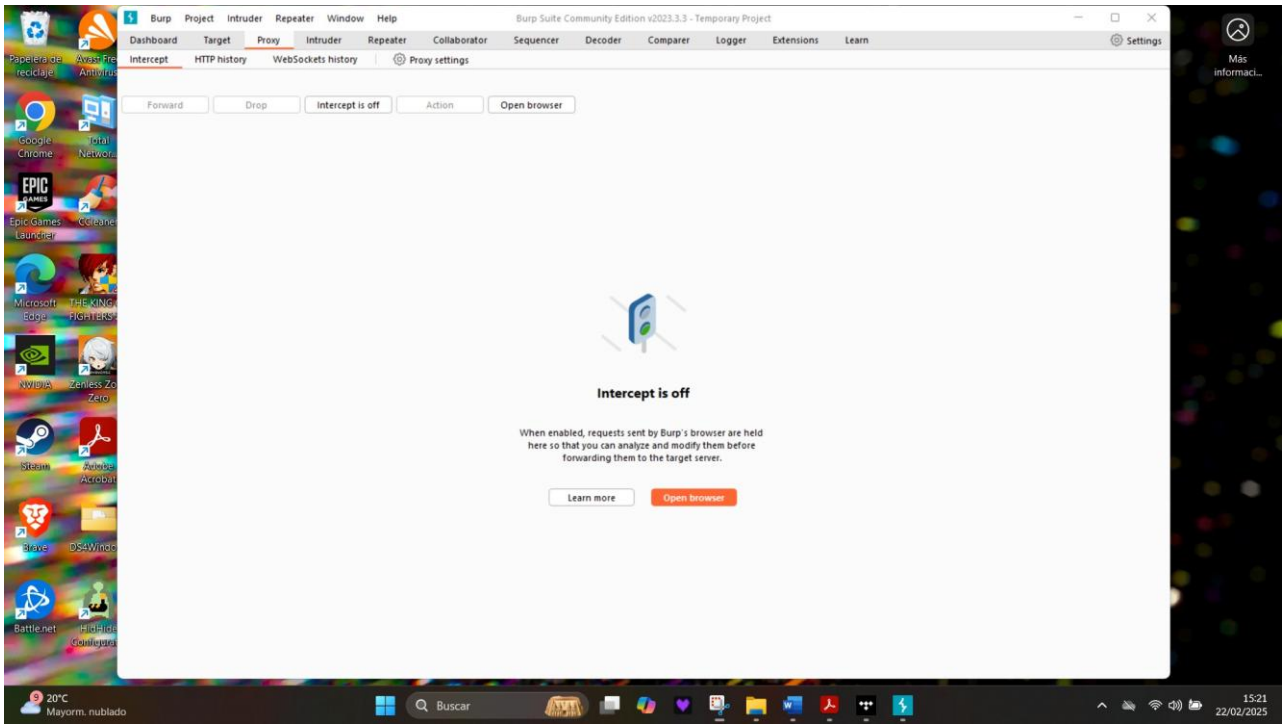


Se inicia la prueba de laboratorio.

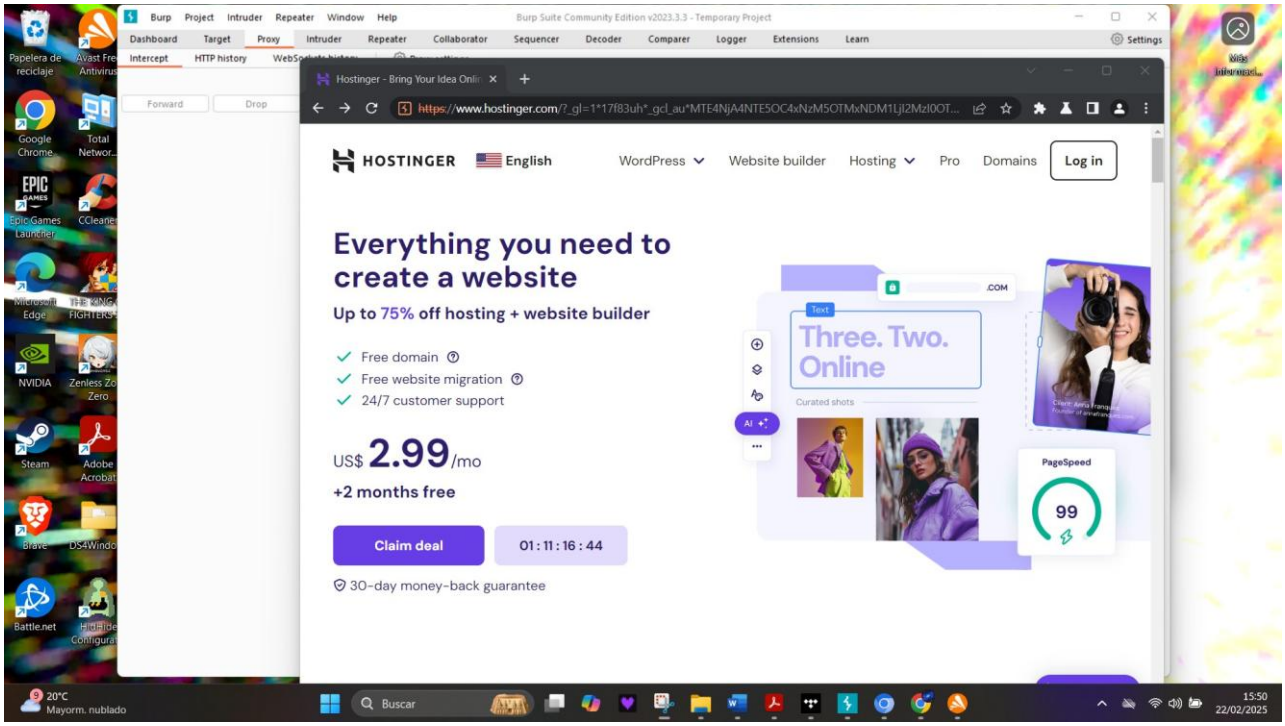


Se muestra informacion en el apartado de Proxy.

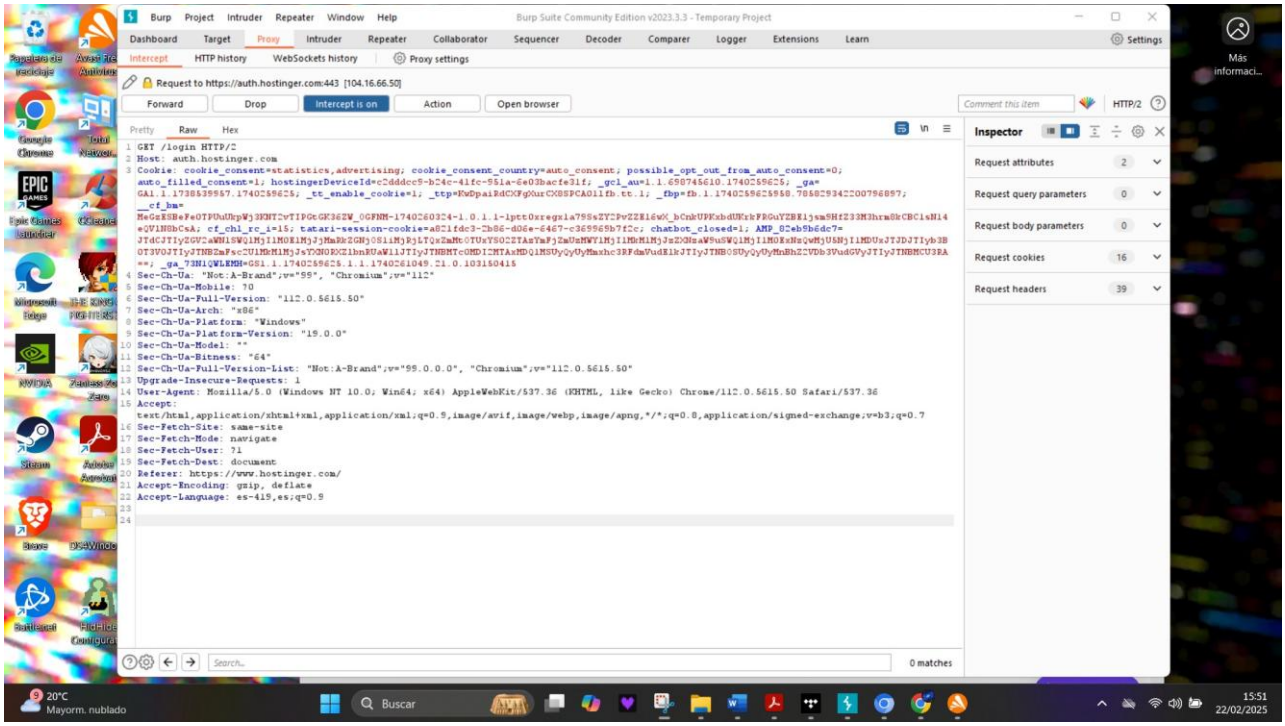
Etapa 3.



Se inicia el software con el nuevo proyecto.



Se ingresa al sitio web.



Se realiza la intrusión y se observa el resultado en Forward.

Conclusión.

Realizar una intrusión utilizando Burp Suite para llevar a cabo un ataque de Cross-Site Scripting (XSS) en un sitio web ha sido una experiencia reveladora y educativa. A través de este ejercicio, he desarrollado una comprensión más profunda de las vulnerabilidades de seguridad en aplicaciones web y la importancia crítica de implementar medidas de protección efectivas.

En primer lugar, aprendí a identificar y explotar las vulnerabilidades XSS, que permiten la inserción de scripts maliciosos en páginas web vistas por otros usuarios. Este proceso me ayudó a reconocer cómo los atacantes pueden manipular entradas no validadas para ejecutar scripts en el navegador de la víctima, lo que puede resultar en el robo de datos sensibles, el secuestro de sesiones o la distribución de malware.

Utilizando Burp Suite, adquirí habilidades prácticas en la interceptación y modificación de tráfico HTTP, lo que es fundamental para identificar puntos de inyección XSS. La herramienta me permitió automatizar el descubrimiento de estas vulnerabilidades y llevar a cabo ataques de manera controlada, reforzando así mi conocimiento sobre la mecánica detrás de los ataques XSS y las técnicas de evasión empleadas por los atacantes.

Además, el ejercicio subrayó la importancia de las prácticas de codificación segura, como la validación y escape de entradas, para mitigar los riesgos asociados con XSS. Comprendí cómo la implementación de Content Security Policy (CSP) y otras medidas de seguridad puede prevenir la ejecución de scripts no autorizados, mejorando significativamente la seguridad de las aplicaciones web.

En resumen, esta experiencia no solo me ha brindado habilidades técnicas esenciales en la utilización de Burp Suite y la explotación de XSS, sino que también ha resaltado la importancia de la seguridad proactiva en el desarrollo de software. Es una lección valiosa que fortalece mi compromiso con la creación de aplicaciones web seguras y robustas.

Referencias.

Video Tutoría 3.

https://academiaglobal-mx.zoom.us/rec/share/pT7vuEJy8CHhHep2BkRhq3iZ5NWIEQJlX4dO2dOPAhv5RyP_UPNj5ErDJRtonduE.4Yd61XR3lbRoUkOY