Universidad Corporativa | Programas Académicos

Ingeniería en Desarrollo de Software

Actividad 2 - Deserialización insegura

Auditoria Informática

Ingeniería en Desarrollo de Software

Tutor: Mtra. Jessica Hernández Romero

Alumno: Fernando Pedraza Garate

Fecha: 20 de octubre del 2024

Índice

Etapa 2 – Deserialización insegura.

- Introducción. Pág. 3
- Descripción Pág. 4
- Justificación Pág. 5
- Ataque al sitio Pág. 6 - 23
- Conclusión Pág. 24
- Referencias bibliográficas. Pág. 25

Introducción

Empezaremos por entender que **la serialización** se refiera al proceso de convertir un objeto o datos en una estructura que puede ser almacenada o transmitida (por ejemplo, convertir un objeto en un archivo o cadena de texto), **la deserialización** es el proceso inverso, donde esa estructura de datos se convierte nuevamente en un objeto usable por la aplicación, es decir, convertir datos que están en un formato específico (como JSON, XML o binarios) de vuelta a un objeto que pueda ser manipulado por una aplicación, y **la deserialización insegura** hace referencia a una vulnerabilidad de seguridad que ocurre cuando una aplicación deserializa datos sin validarlos adecuadamente, permitiendo a un atacante inyectar o manipular datos maliciosos en el proceso.

Si el proceso de deserialización no se maneja correctamente, un atacante puede aprovecharlo para inyectar objetos maliciosos que pueden hacer que el sistema ejecute comandos no deseados o acceda a datos confidenciales. impactando en una ejecución remota de código (RCE), manipulando los datos para ejecutar código malicioso en el servidor, lo que puede resultar en una toma de control total del sistema afectado donde el atacante puede deserializar objetos que contienen información confidencial, como credenciales o información personal para realizar acciones no autorizadas, como modificar permisos o el comportamiento de una aplicación, ocasionando ataques de negación de servicio (DoS), datos maliciosos o desproporcionadamente grandes, intentando sobrecargar el servidor para provocar fallos en el sistema.

Descripción.

Una empresa de software está solicitando realizar varias pruebas de seguridad en páginas web que no cuentan con los candados de seguridad, y para esta segunda etapa, solicita realizar una prueba de deserialización insegura en una página específica mediante las cookies, para lograrlo se debe utilizar el programa Burp Suite Community Edition, con el objetivo de iniciar sesión como un usuario normal y luego pasar a modo administrador a través de las cookies, y con la ayuda de la plataforma PortSwigger, se realizará el ataque a la página proporcionada por ellos, en la que se iniciará sesión con las credenciales que se proporcionan, las cuales son para usuarios normales; no obstante, a través de las cookies, se entrara al modo administrador.

Este laboratorio utiliza un mecanismo de sesión basado en serialización y, por ende, es vulnerable a la escalada de privilegios, en el que hay que editar el objeto serializado en la cookie de sesión para aprovechar esta vulnerabilidad, obteniendo privilegios administrativos para finalmente, **eliminar la cuenta de Carlos como objetivo e iniciar sesión en la propia cuenta con las credenciales:**

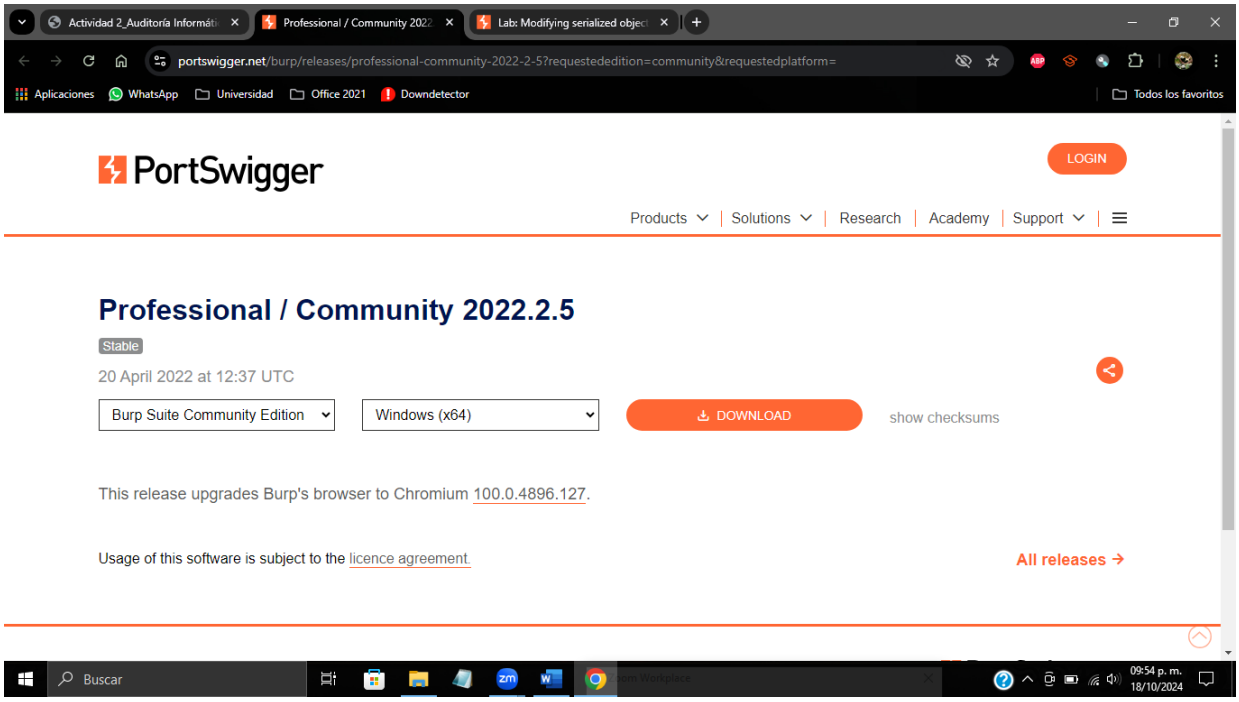
Usuario: **wiener** y contraseña: **Peter**

Justificación.

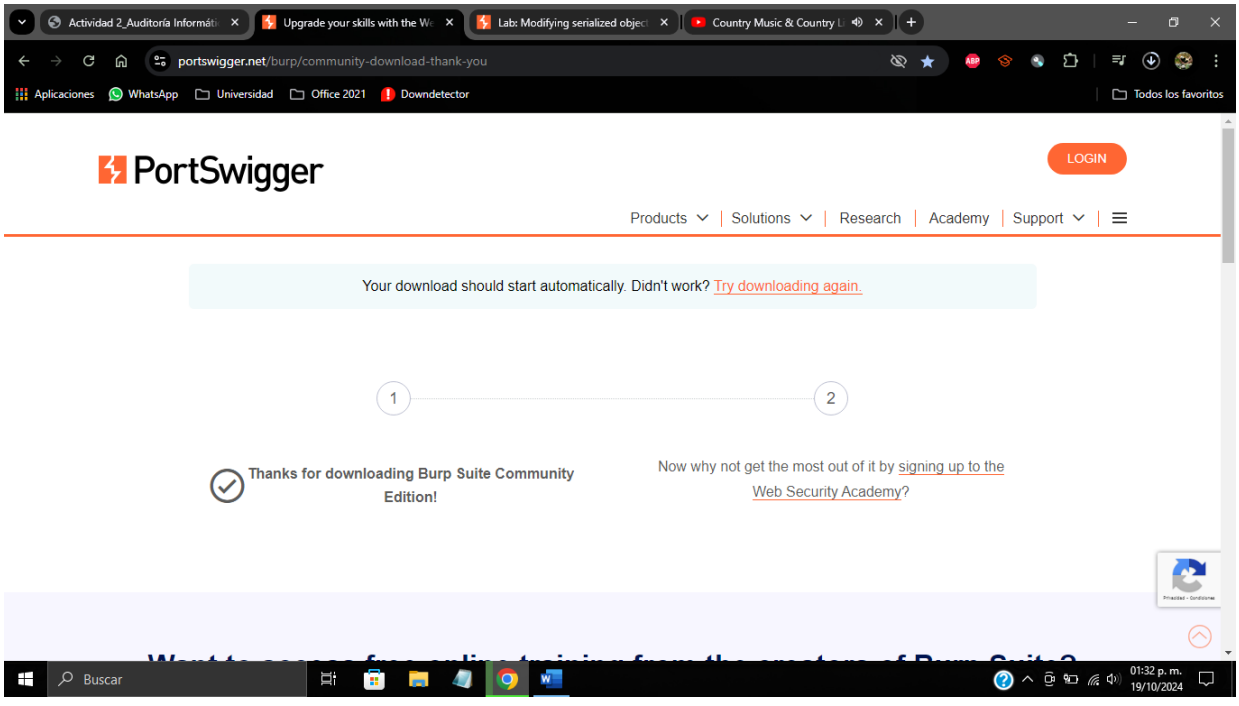
La deserialización insegura radica en que este tipo de vulnerabilidad permite a los atacantes explotar las funcionalidades internas de una aplicación para realizar acciones maliciosas a través de la inyección de datos manipulados en el proceso de deserialización, comprometiendo la integridad, confidencialidad y disponibilidad del sistema afectado, por esta razón, **es crucial implementar mecanismos robustos de validación y seguridad, así como evitar la deserialización de fuentes no confiables para mitigar estos riesgos.**

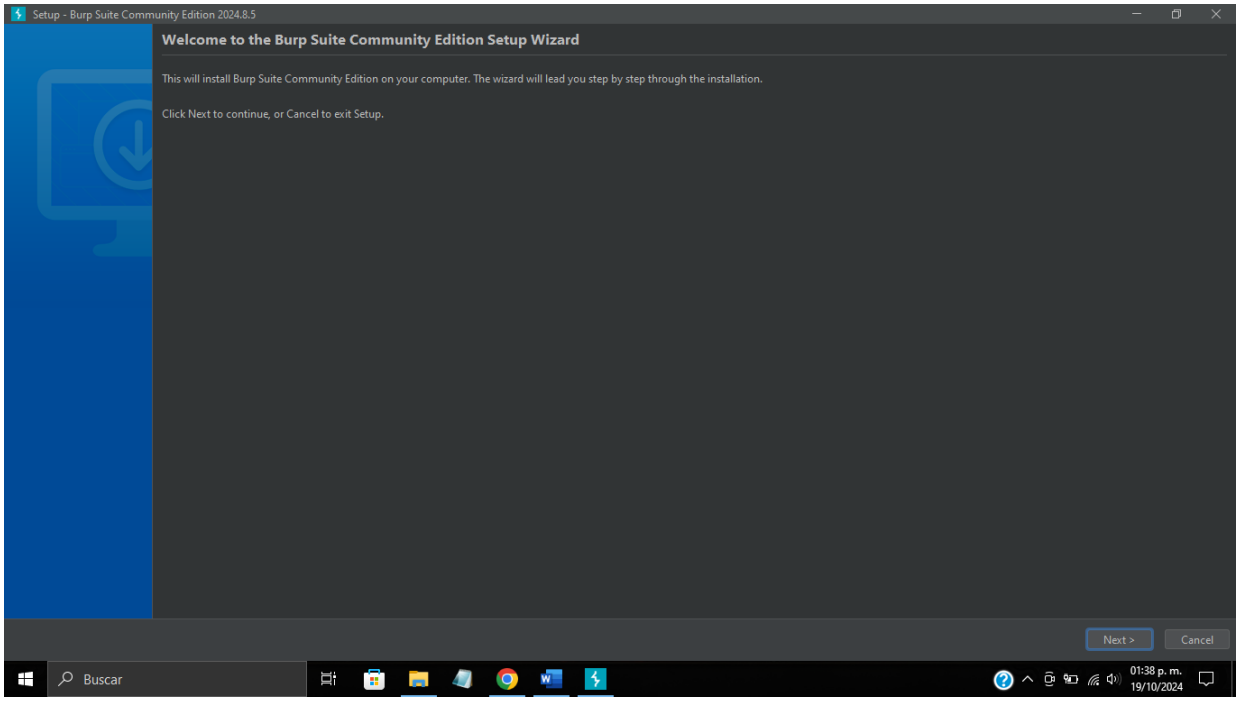
Existen casos documentados donde las empresas han sufrido graves consecuencias debido a la deserialización insegura, impactando en los servidores, comprometiendo por completo la seguridad del sistema, llevando al acceso no autorizado a datos confidenciales, con implicaciones legales y de privacidad, ocasionando que estos ataques pueden ser difíciles de detectar porque ocurren en la capa de deserialización, lo que justifica la necesidad de un manejo proactivo de esta vulnerabilidad, las organizaciones y entidades de seguridad, como OWASP (Open Web Application Security Project), reconocen la deserialización insegura como una de las principales amenazas en aplicaciones web, lo que justifica la necesidad de seguir prácticas seguras, como la de validar y filtrar los datos antes de deserializarlos, asegurando que provienen de fuentes confiables, o usar métodos más seguros para transferir y almacenar datos, implementando medidas de seguridad adicionales, como autenticación, autorización y uso de controles de acceso.

Ataque al sitio.

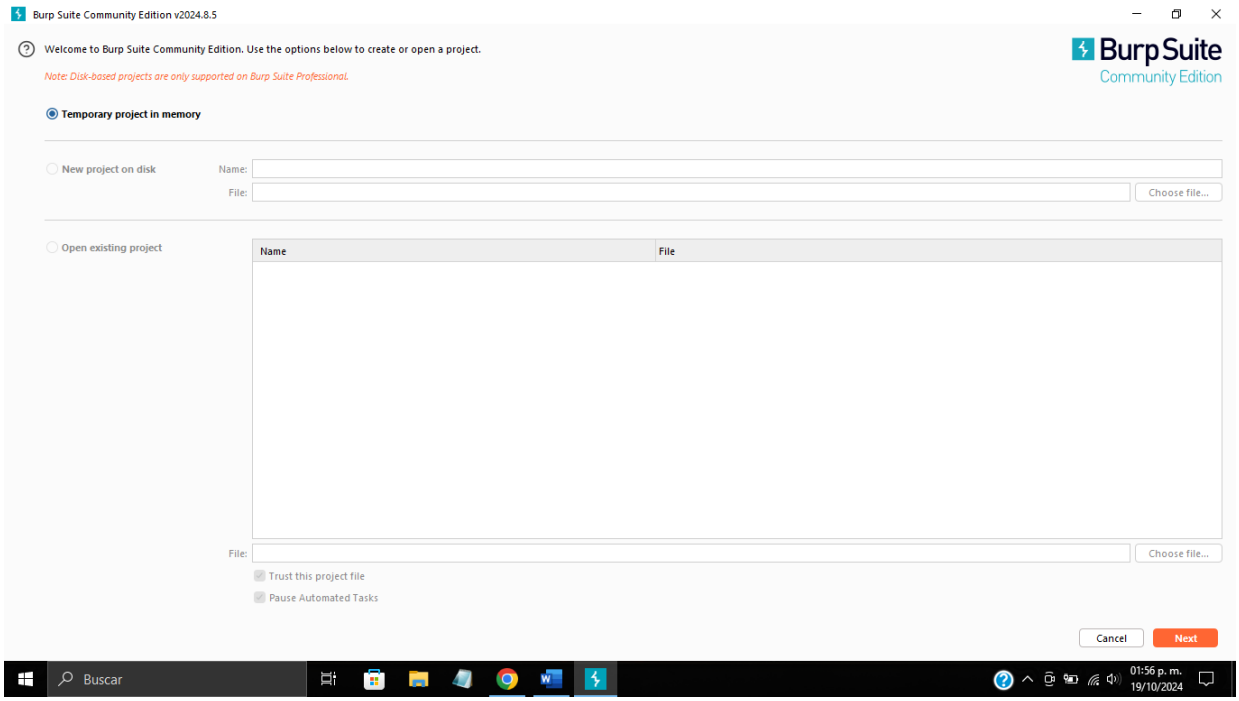


Se ingresa a PortSwigger y se descarga el programa Burp Suite Comunity Edition

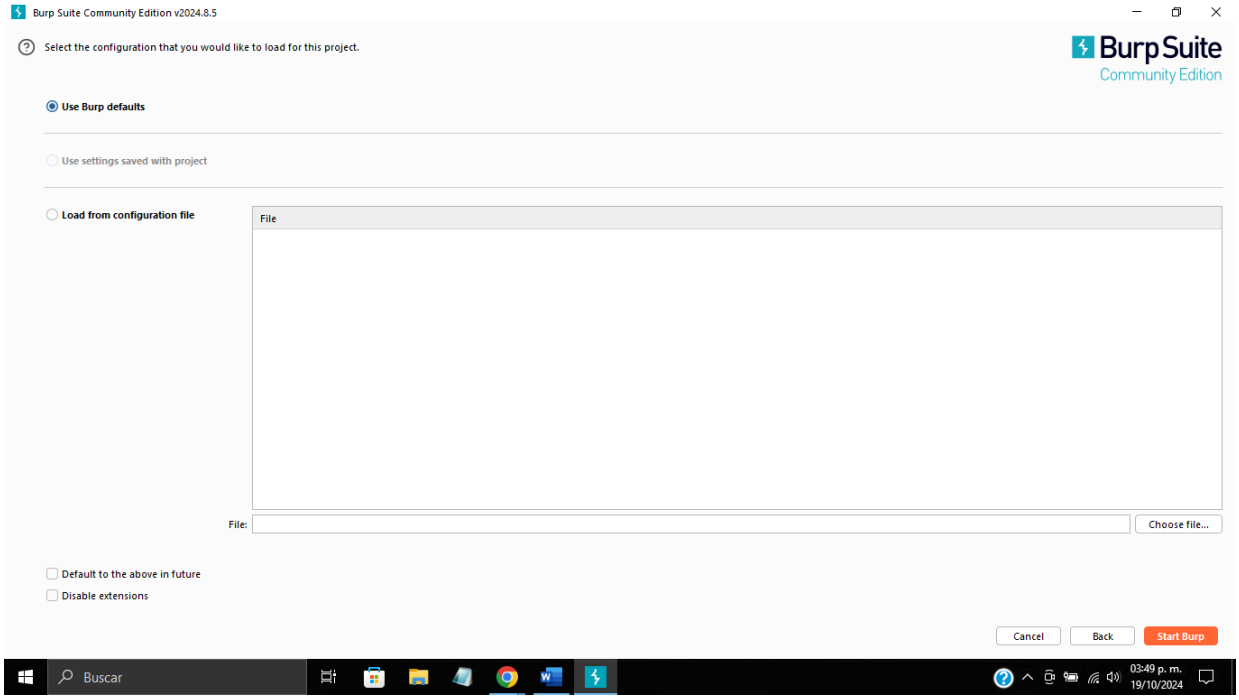




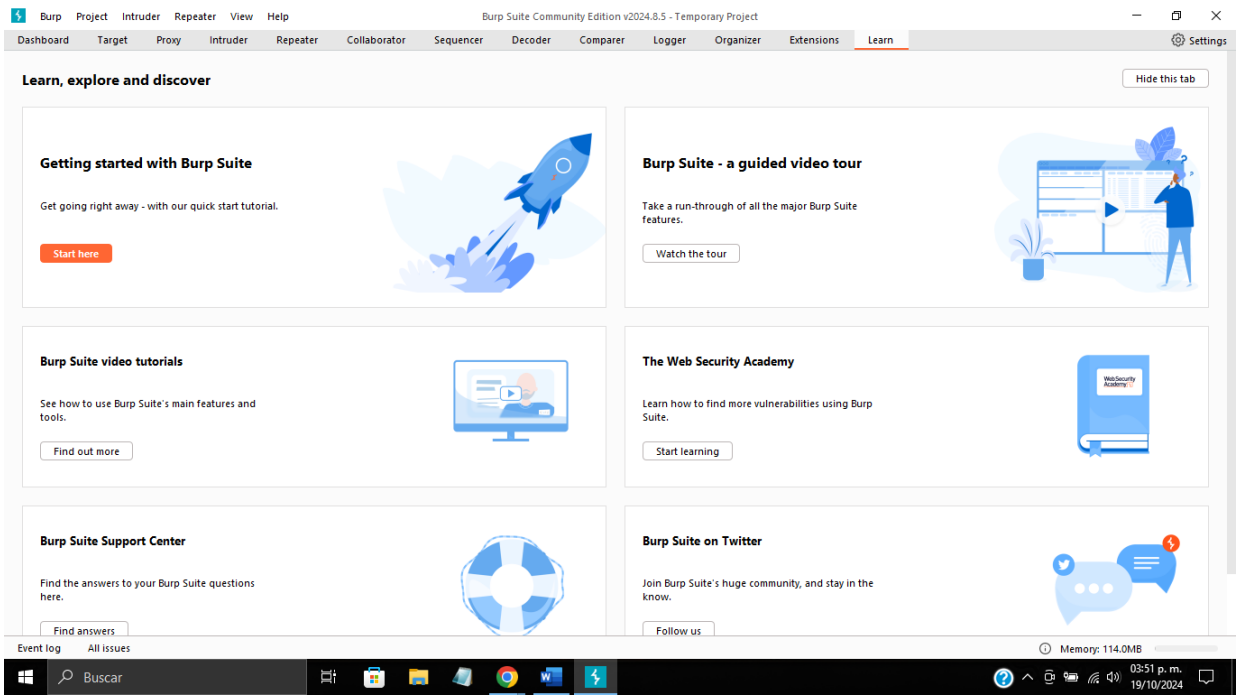
Para posteriormente instalarlo en el equipo y abrir el programa

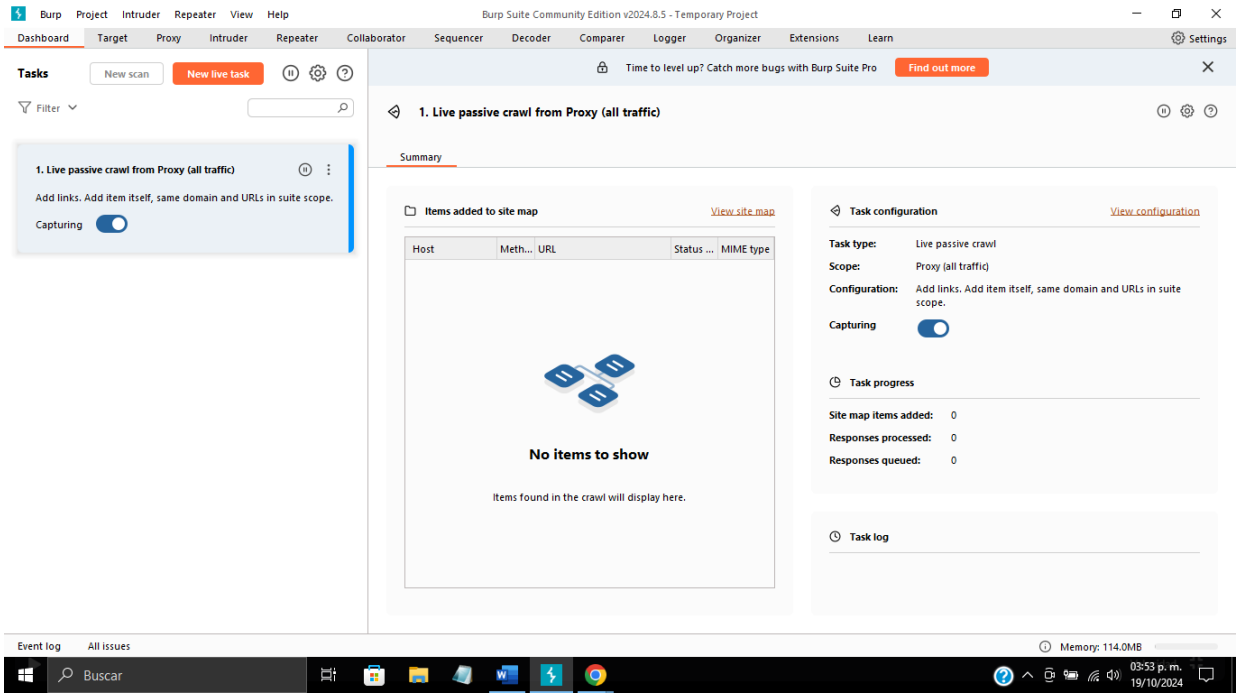


Se deja todo por default

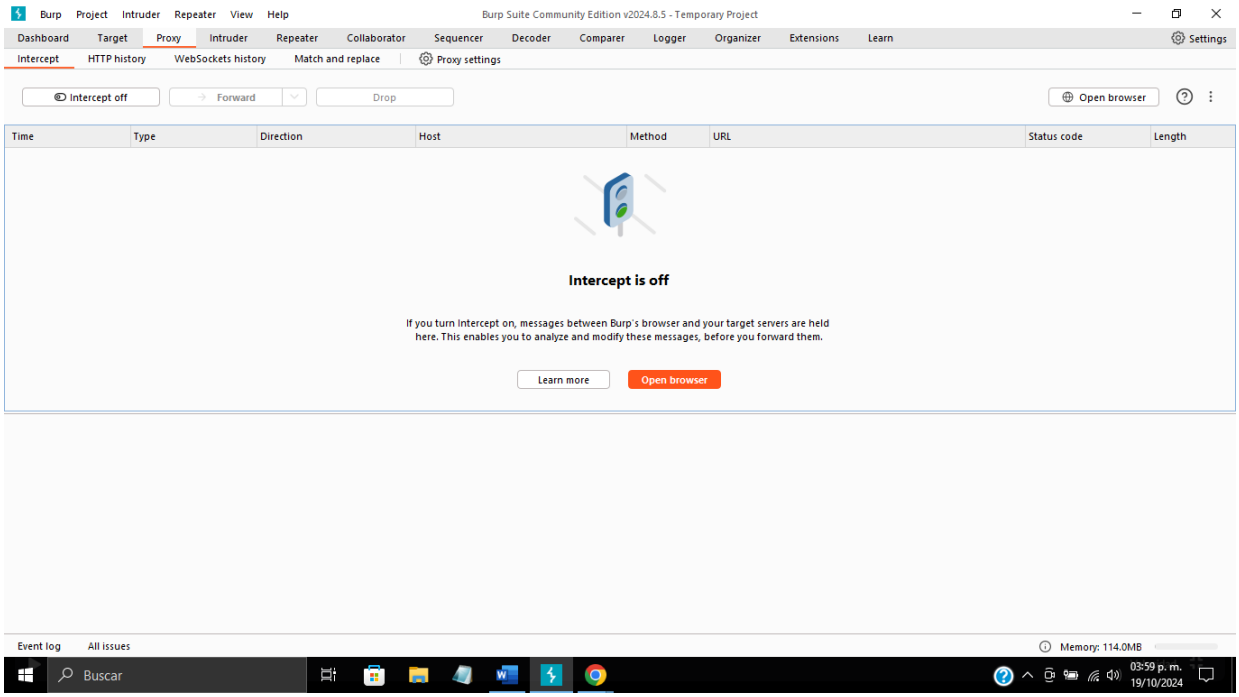


Se inicia el Burp

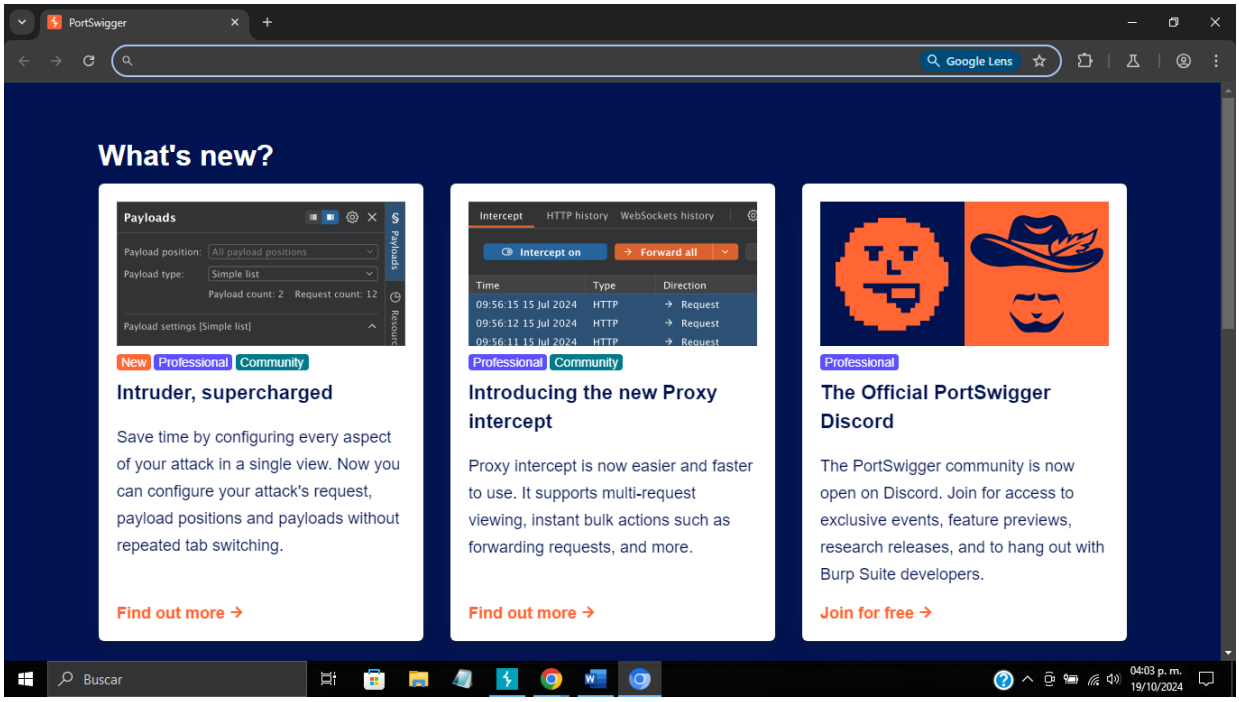


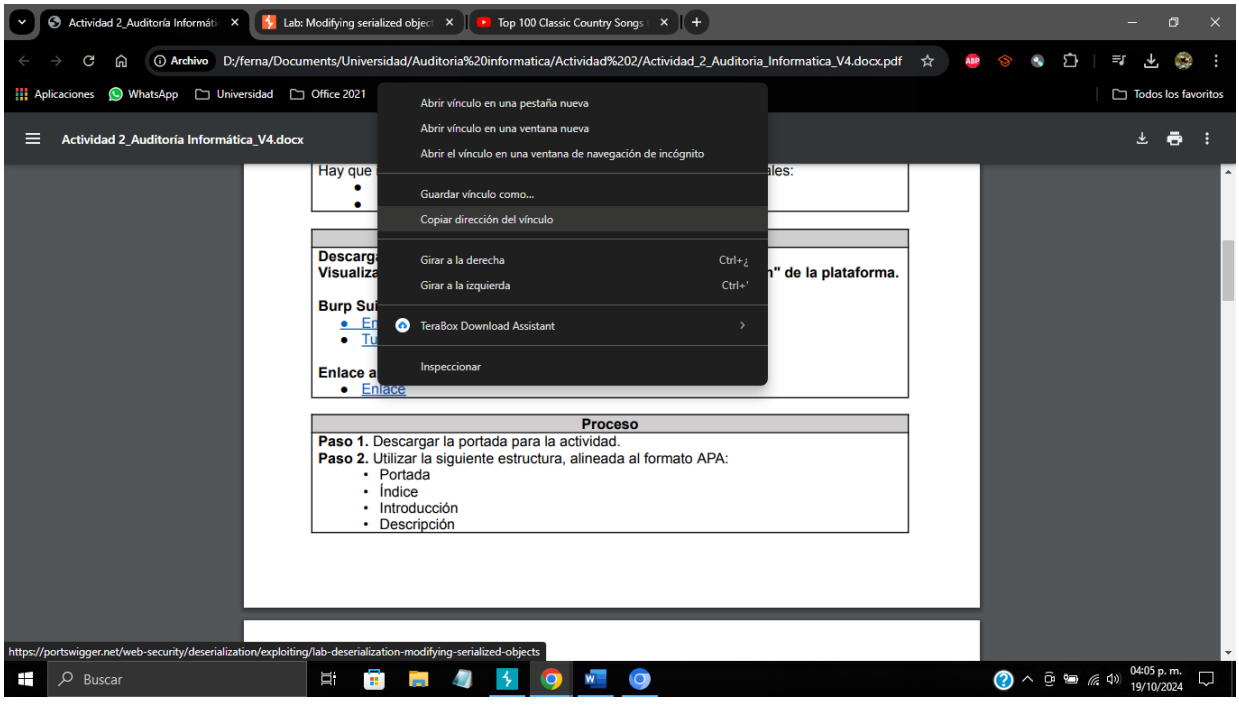


Se ingresa al Dashboard

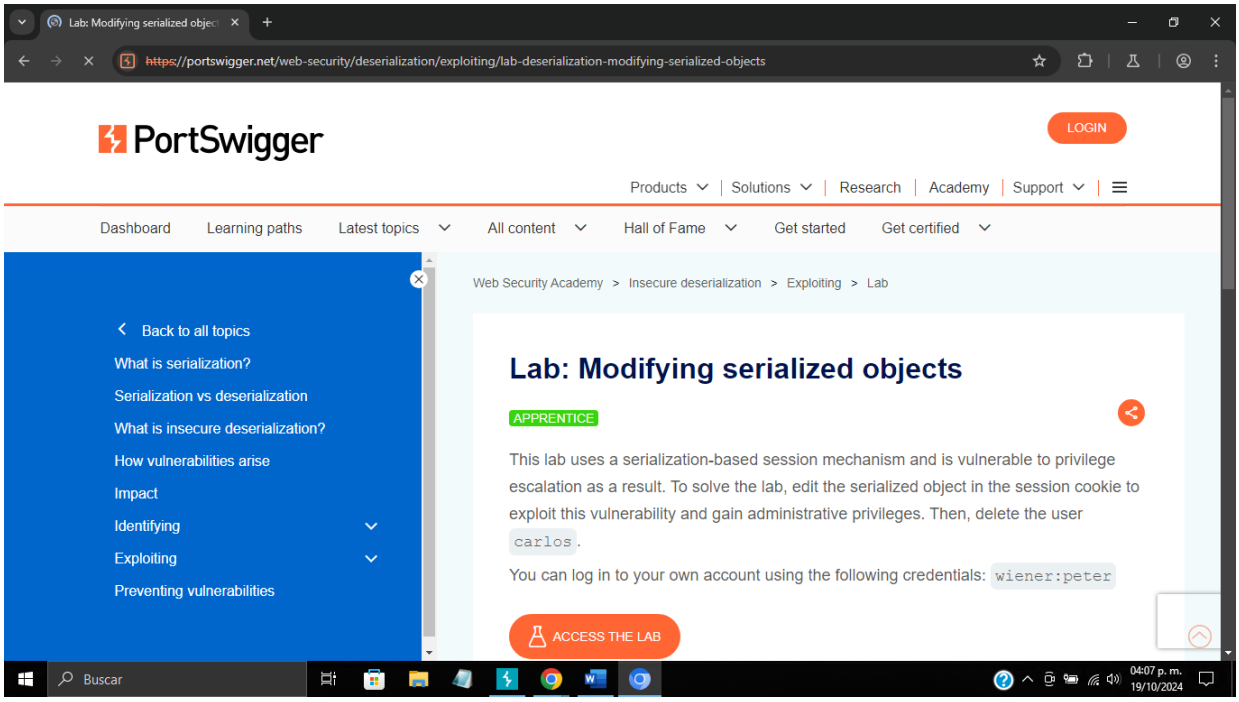


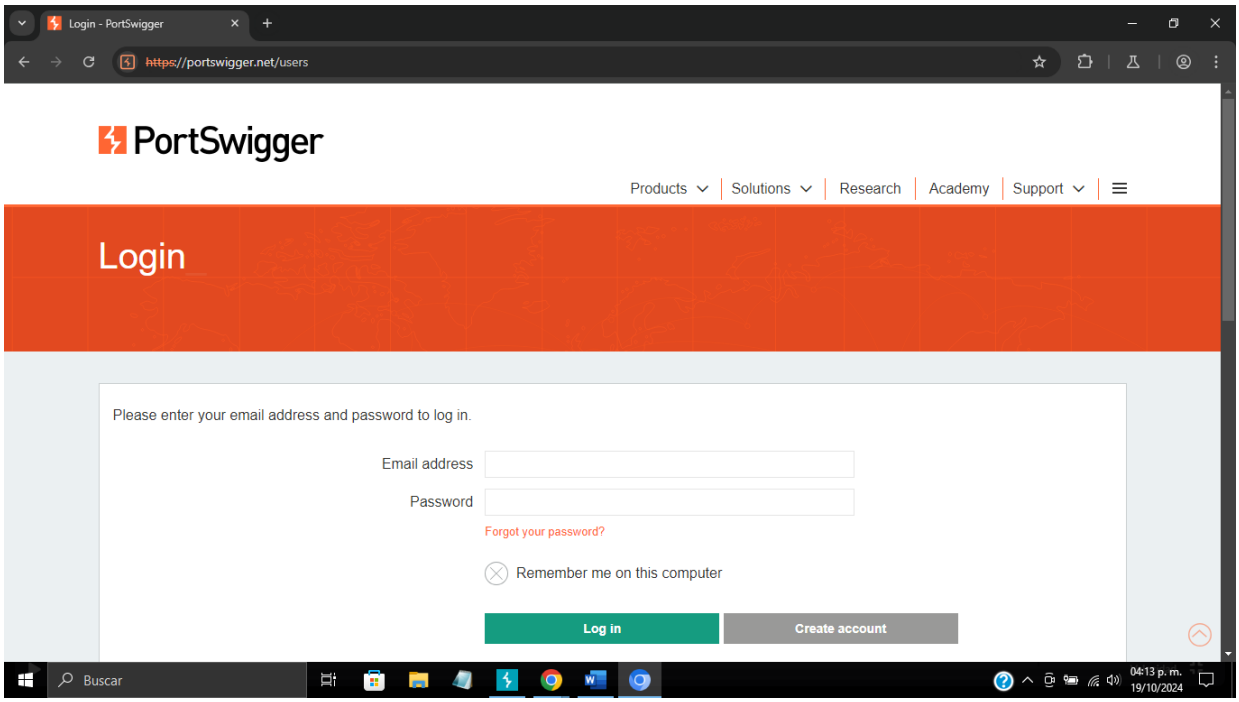
Se selecciona el apartado proxy para abrir el navegador



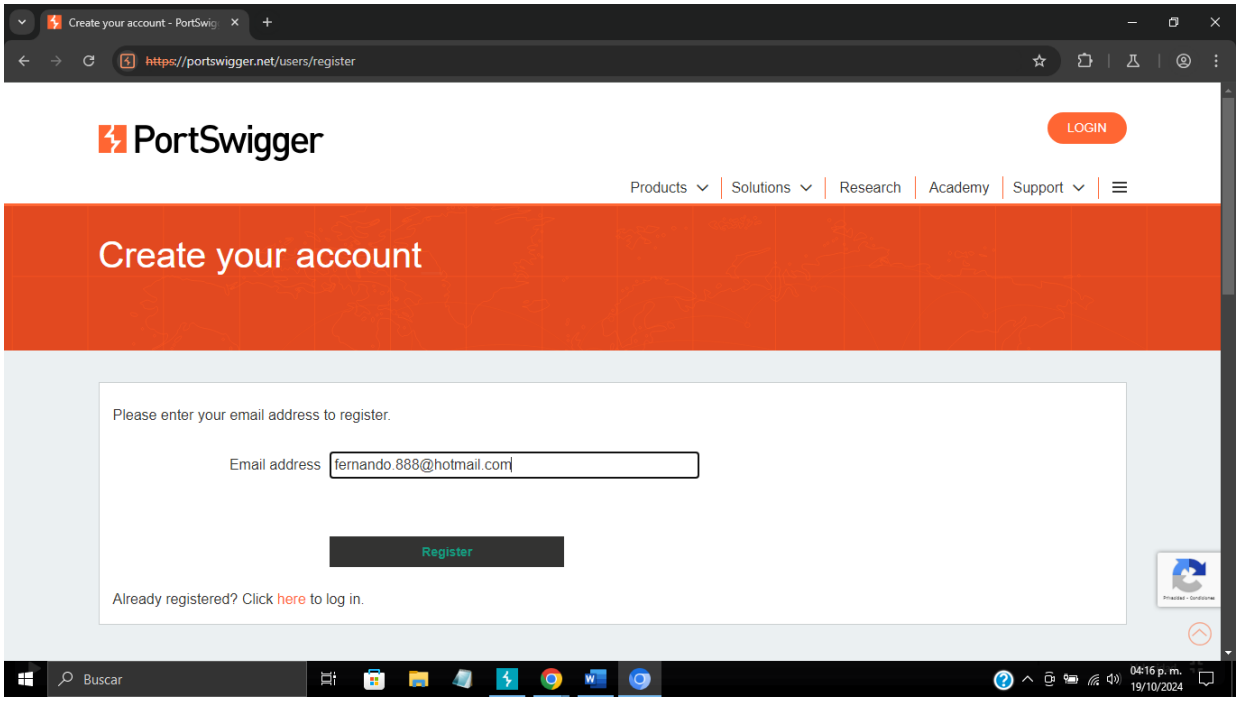


Se copia el enlace de la práctica y se pega el enlace en el navegador que se abrió de Burt para la practica

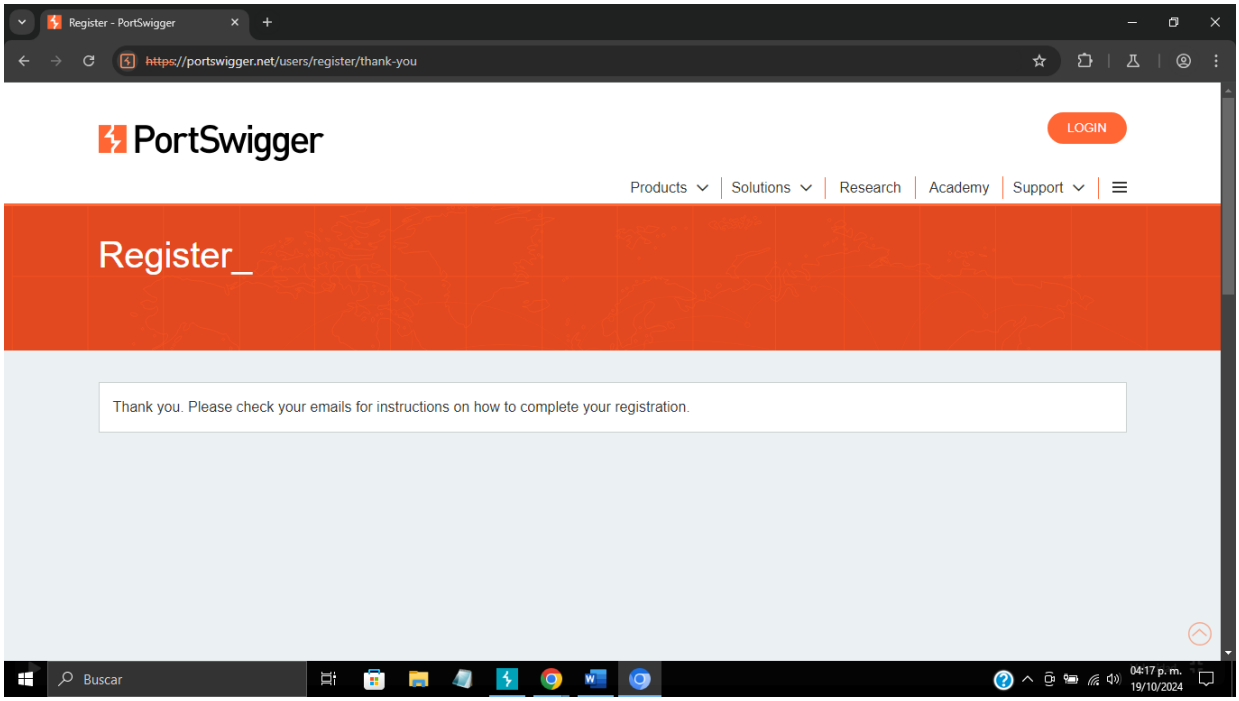


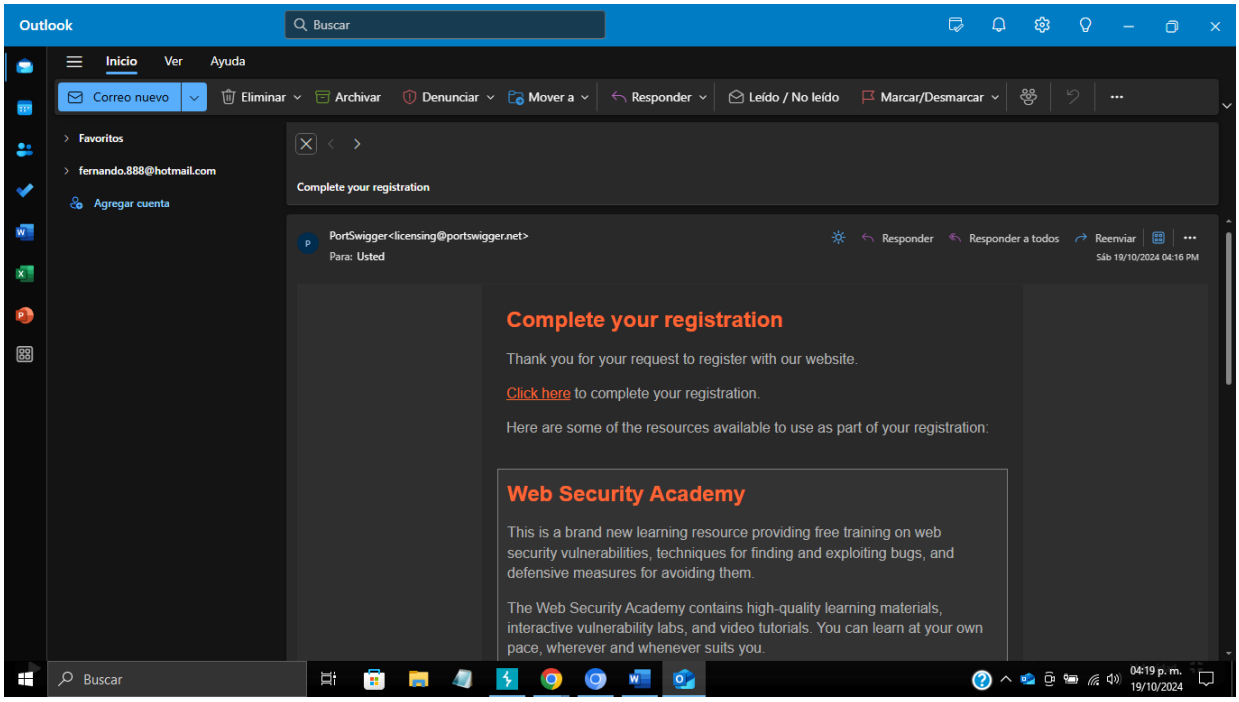


Se crea una nueva cuenta de usuario en PortSwigger para accesar al laboratorio de practica

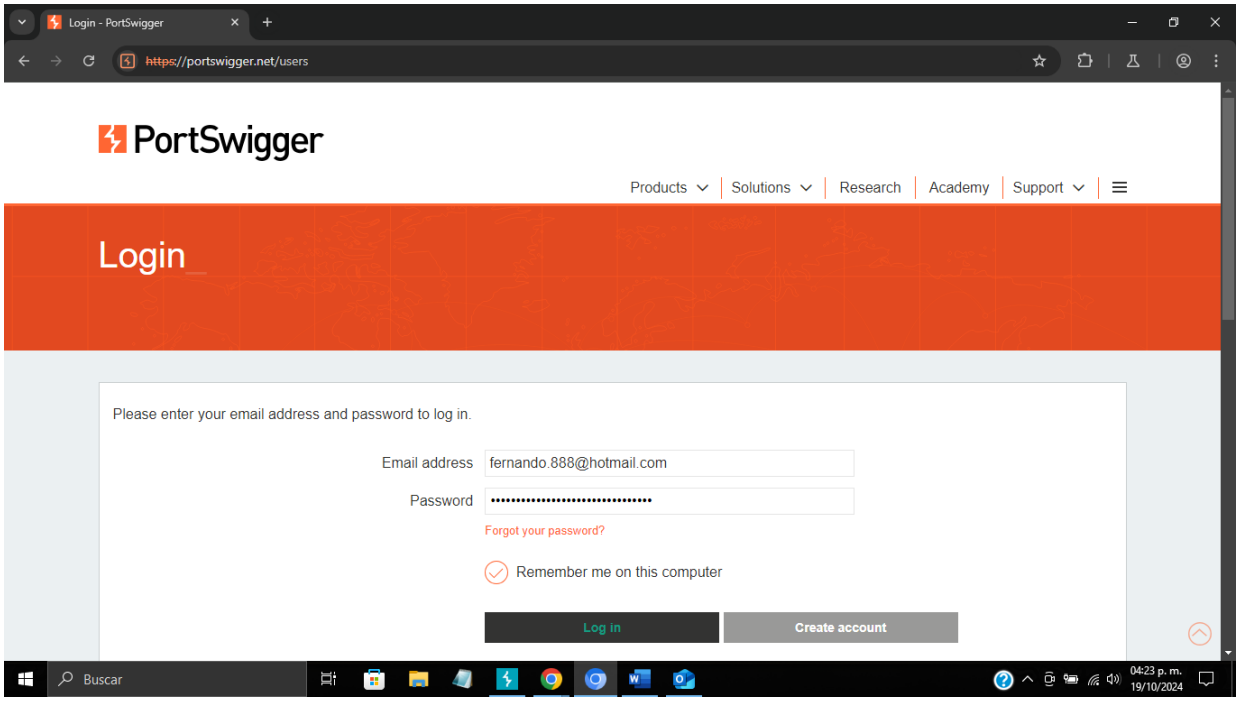


Y se siguen las instrucciones para completar el registro

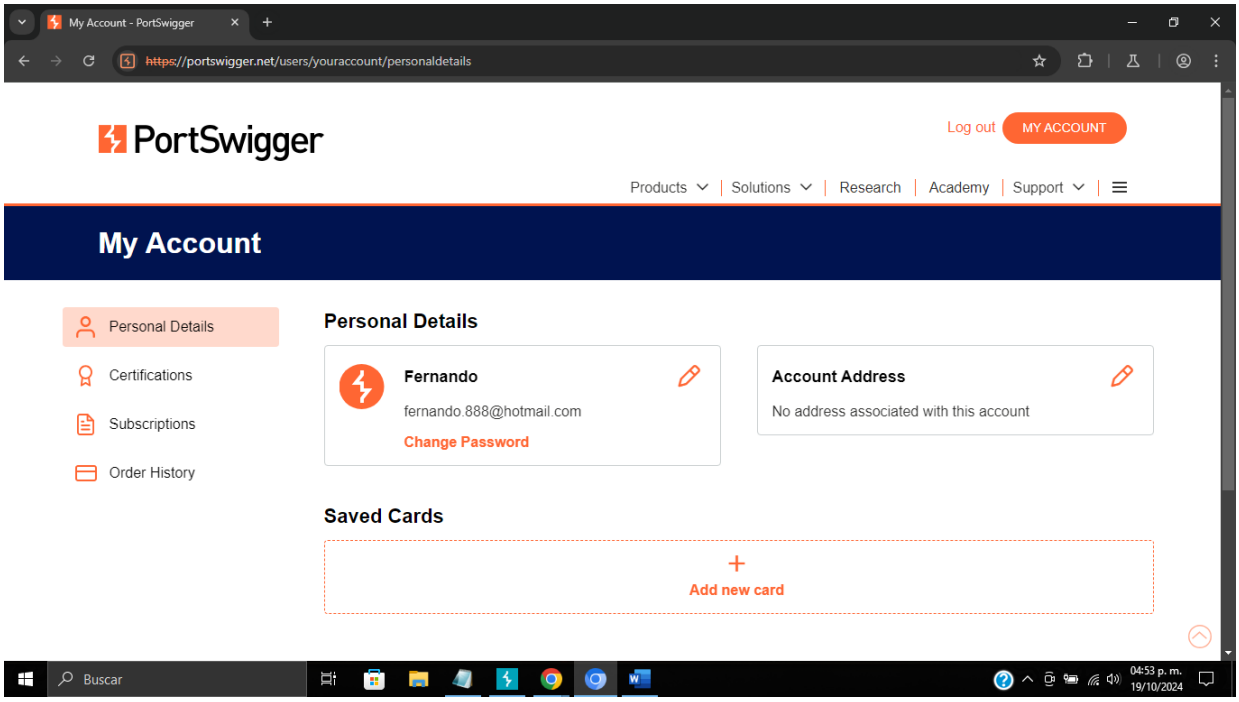




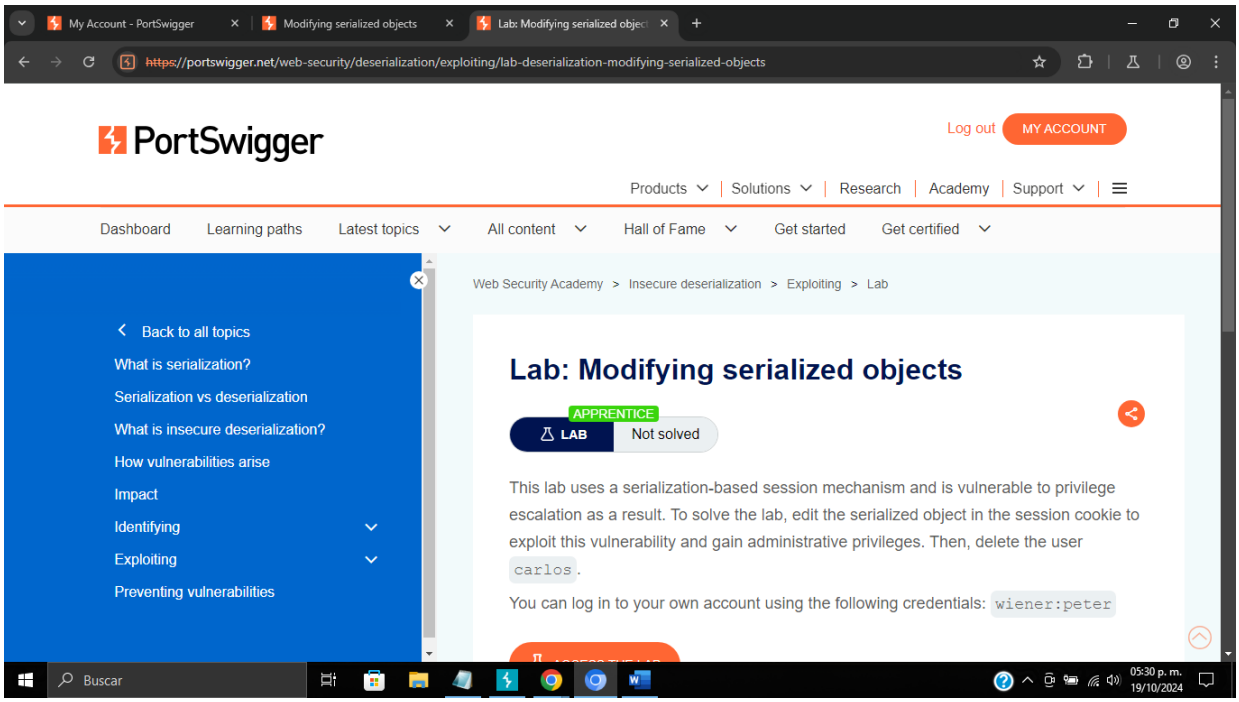
Confirmando la cuenta desde el correo



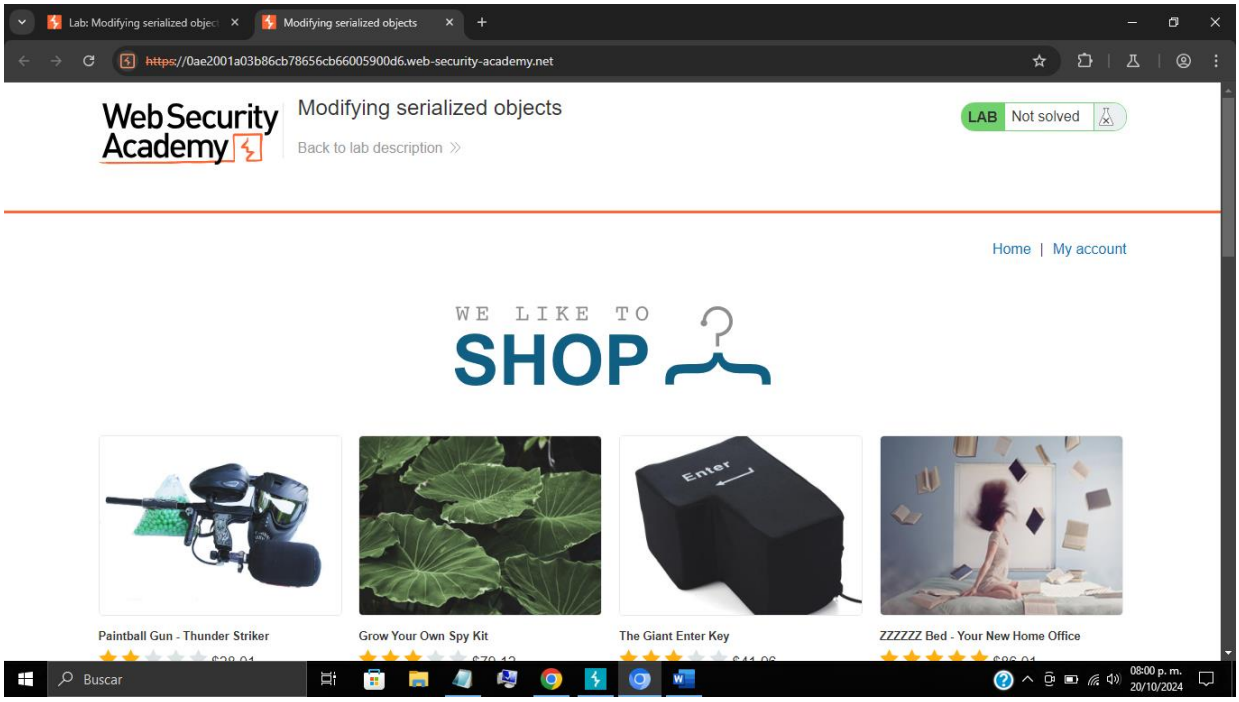
Se inicia sesión



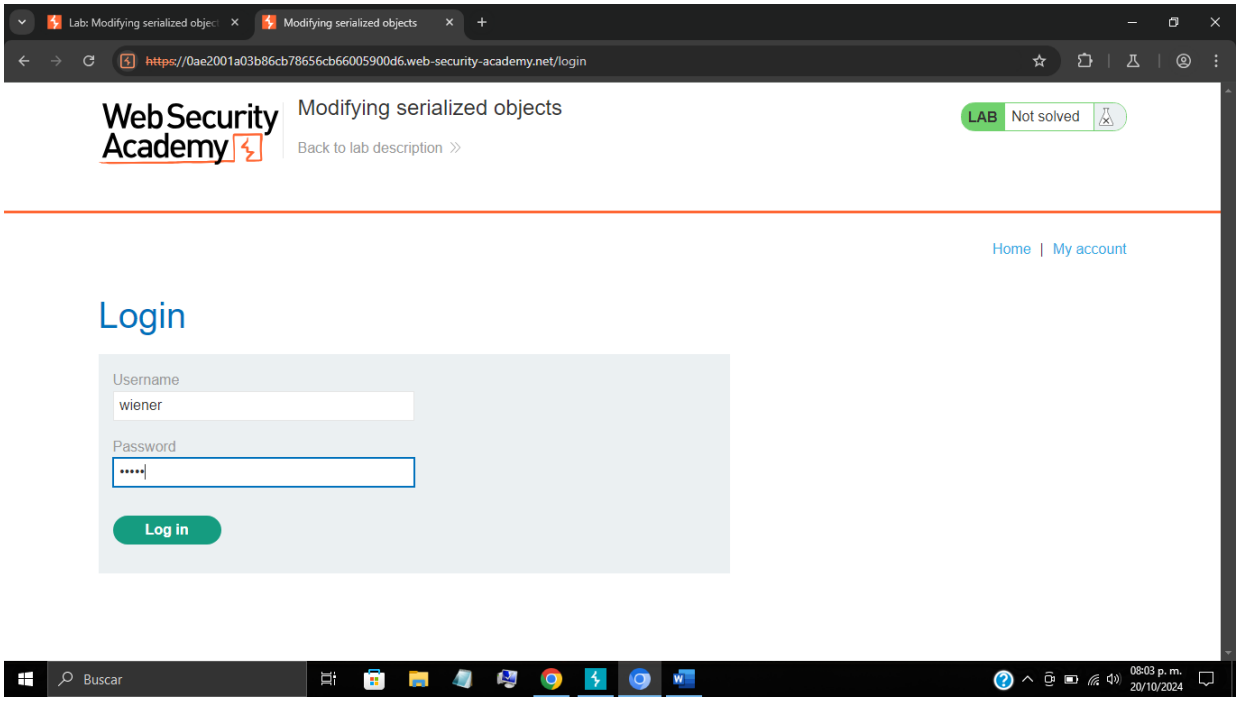
Una vez que se tiene el acceso se ingresa al enlace de la practica

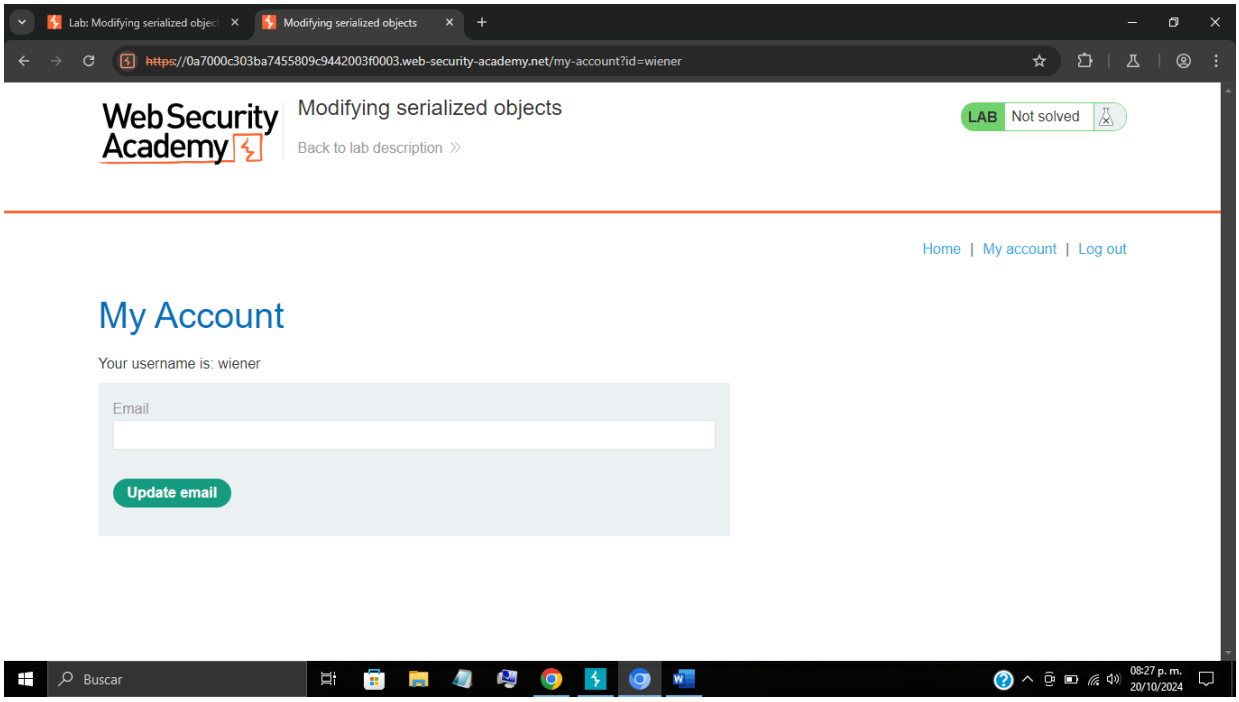


Y se ingresa al laboratorio de practica

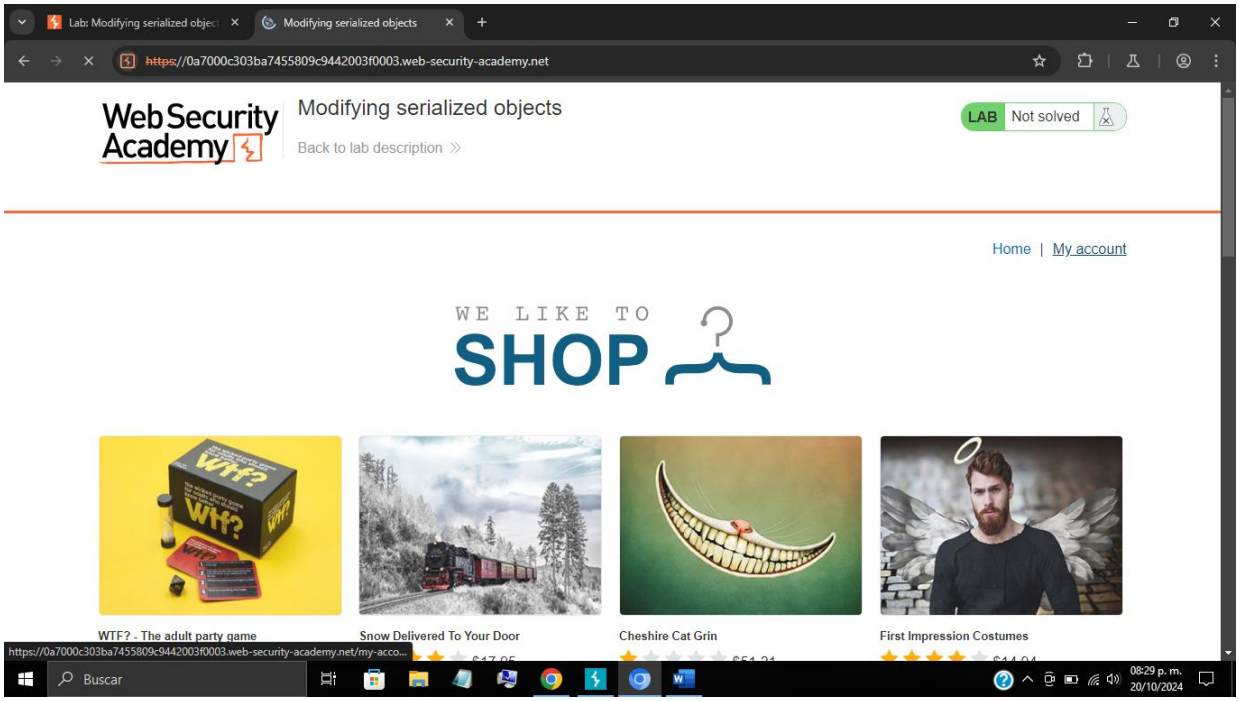


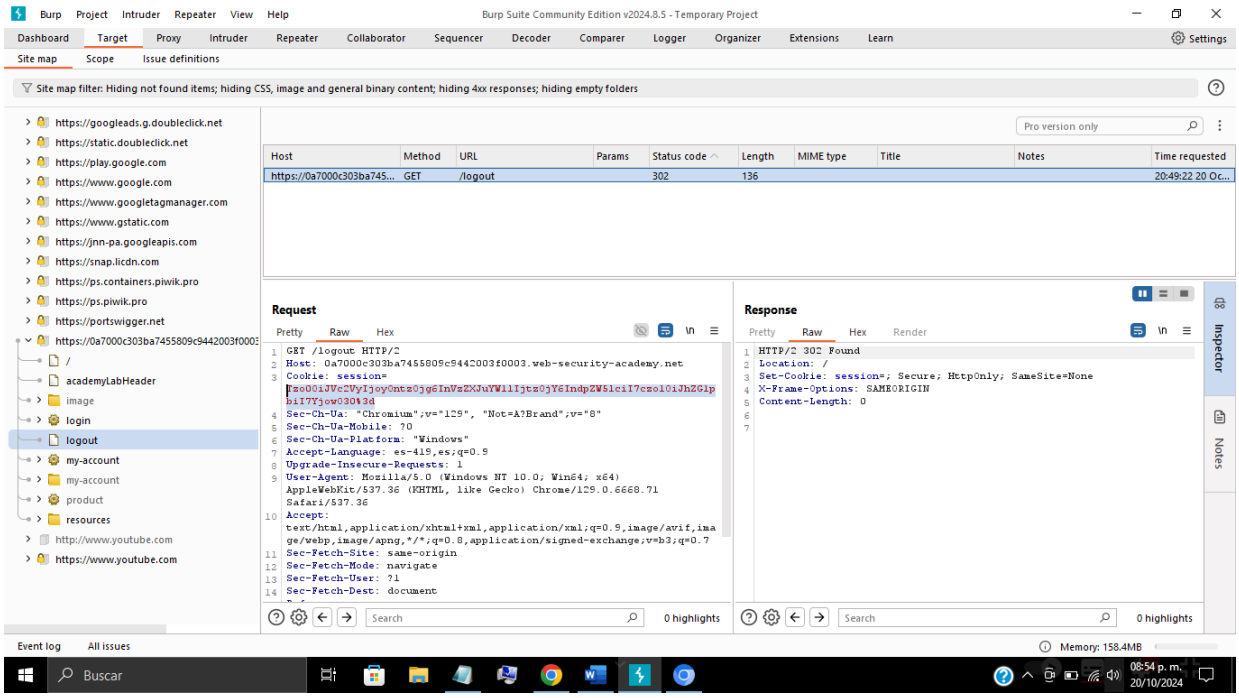
Para posteriormente ingresar el usuario y contraseña proporcionados



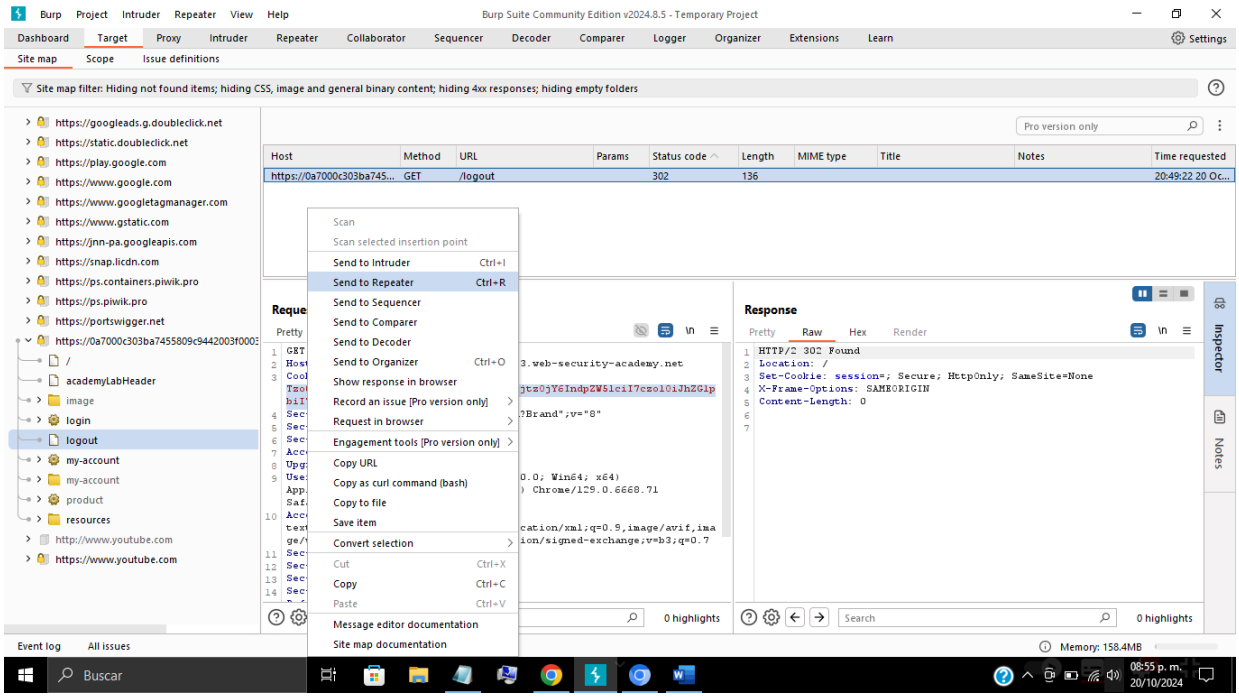


Una vez ingresado en el apartado log out cerramos sesión

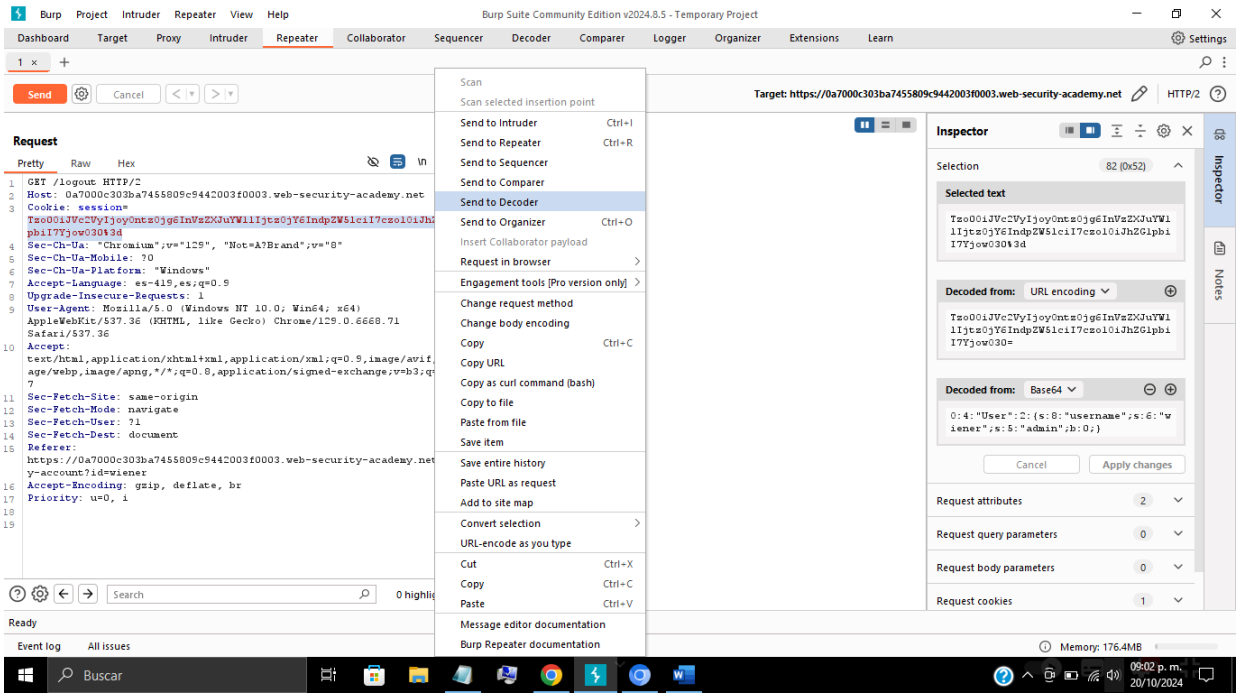




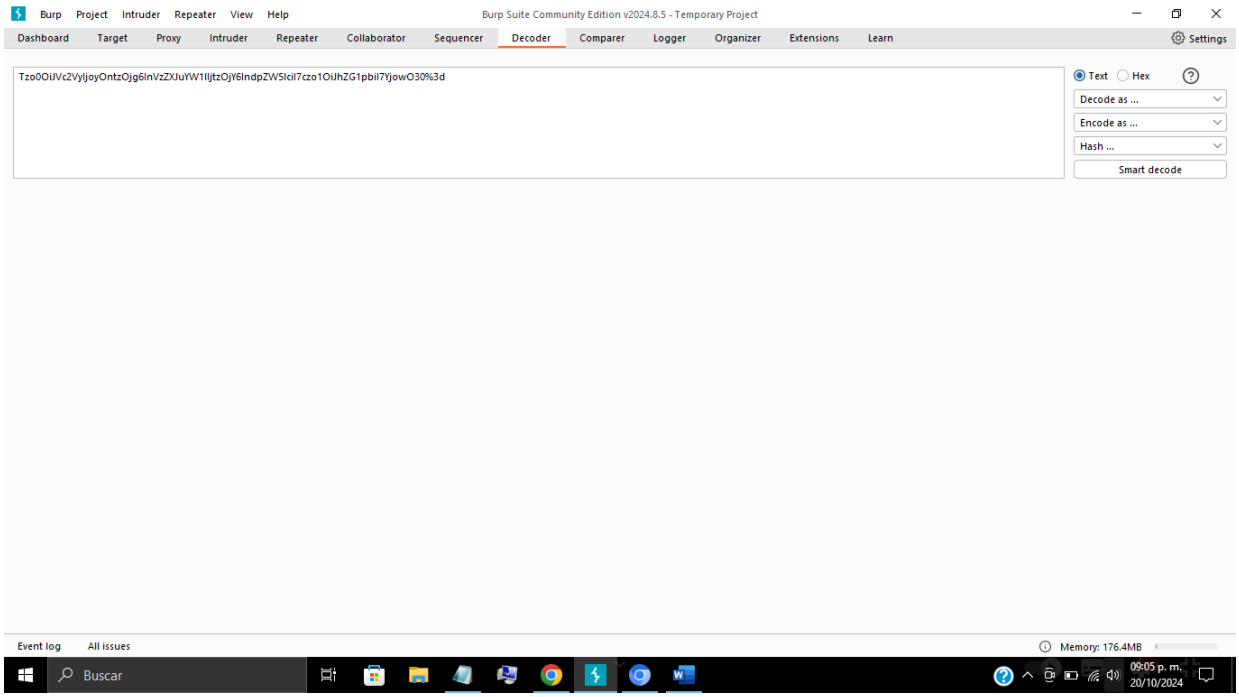
En Burt se puede observar en el target y sitio del mapa la alerta de vulnerabilidad



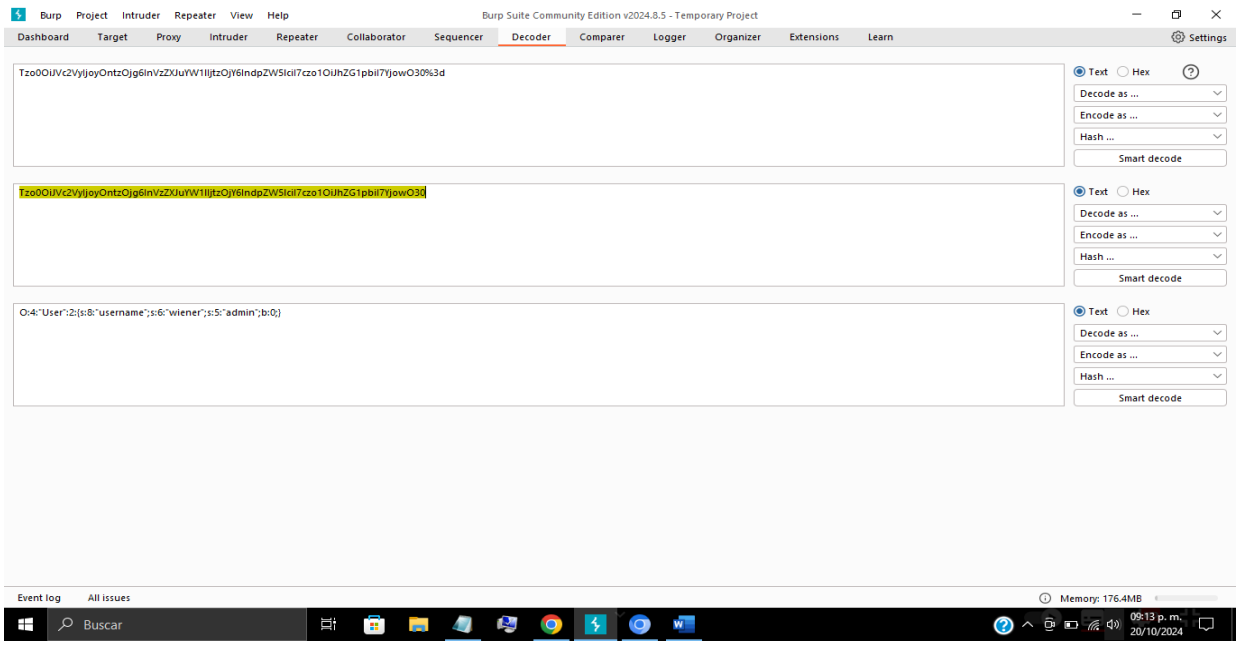
Se envía al repetidor



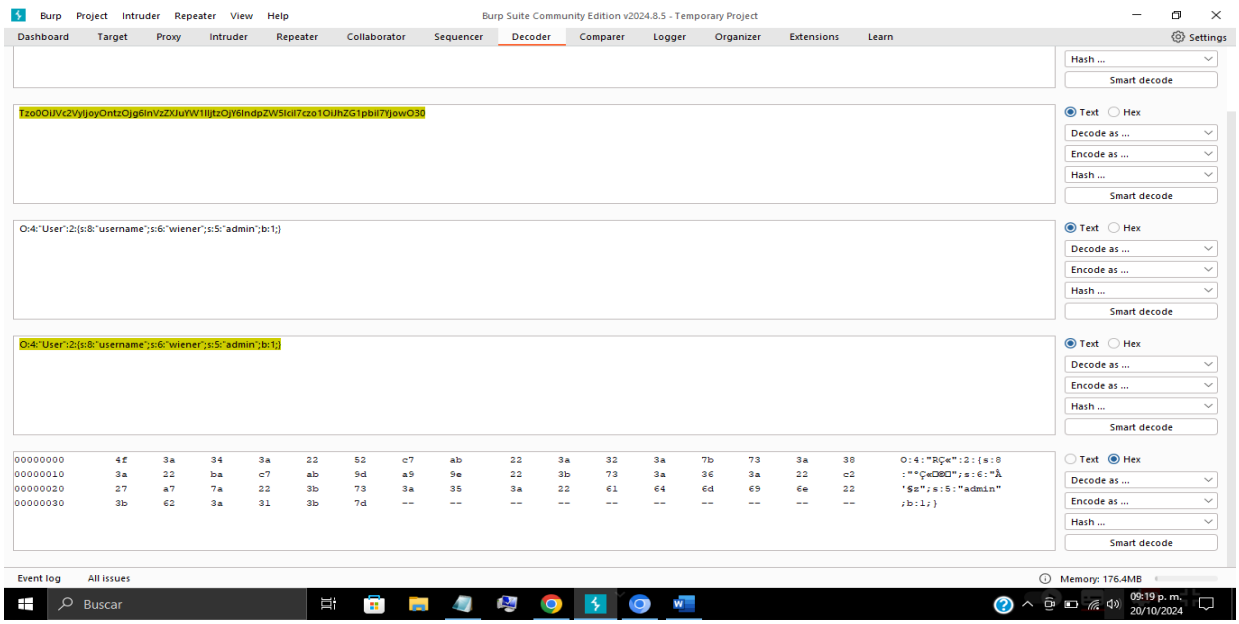
Se envía el enlace al decodificador



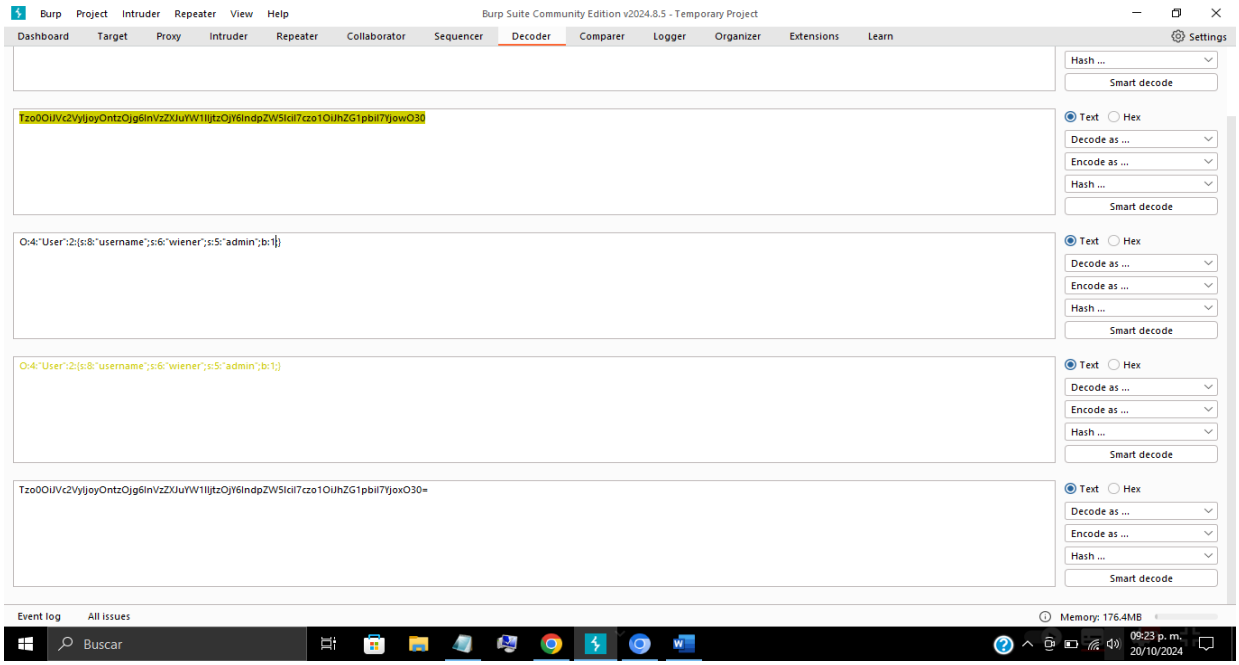
Se modifica la terminación del enlace eliminando %3d



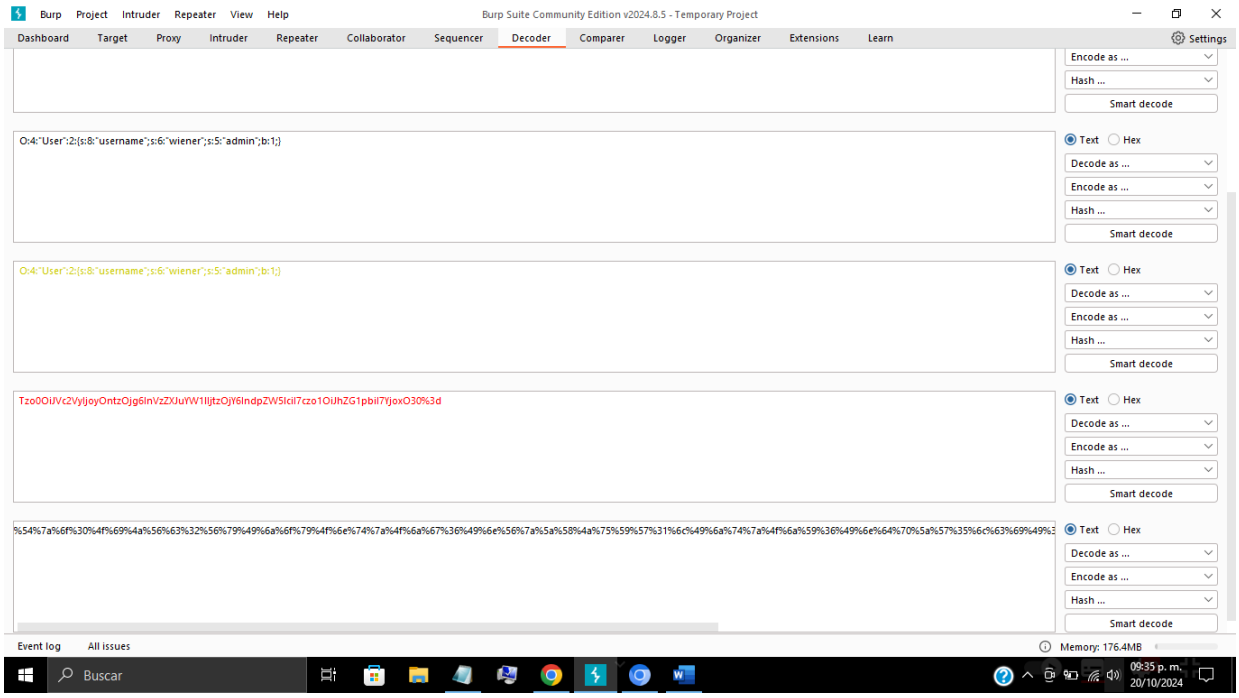
Para posteriormente decodificarlo a Base64



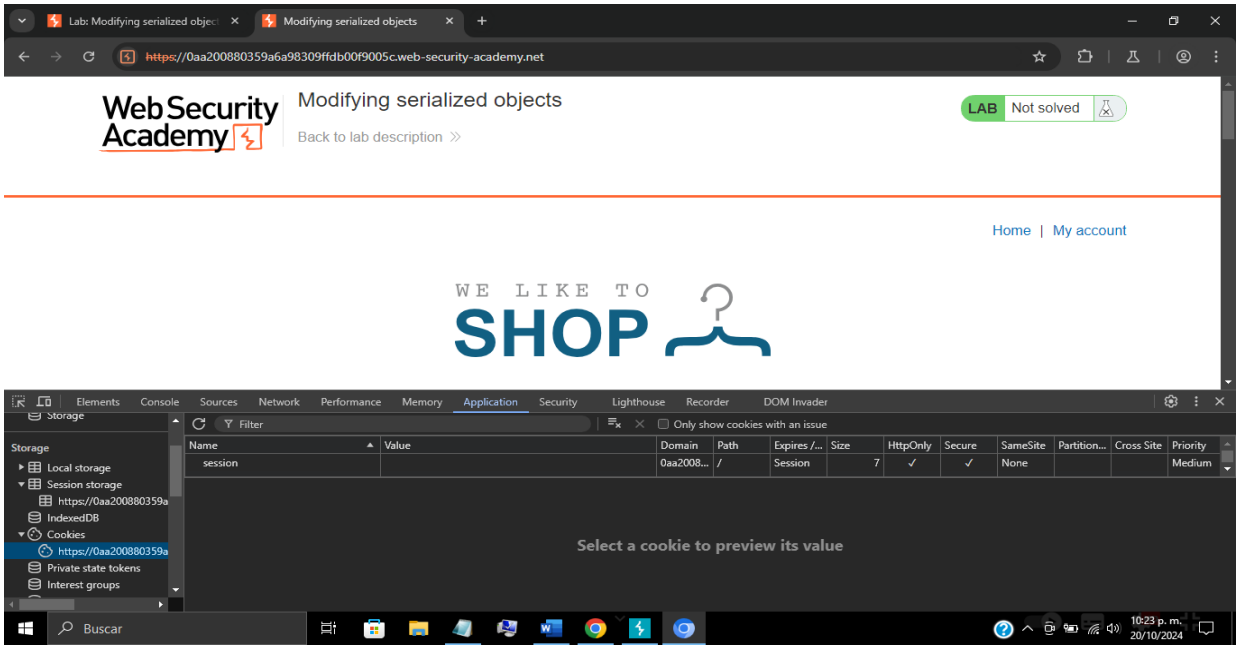
Se modifica el enlace decodificado cambiando el 0 por 1 para cambiar los privilegios del usuario (O:4:"User":2:{s:8:"username";s:6:"wiener";s:5:"admin";b:**0**;}), quedando: (O:4:"User":2:{s:8:"username";s:6:"wiener";s:5:"admin";b:**1**;}), igual a Base64



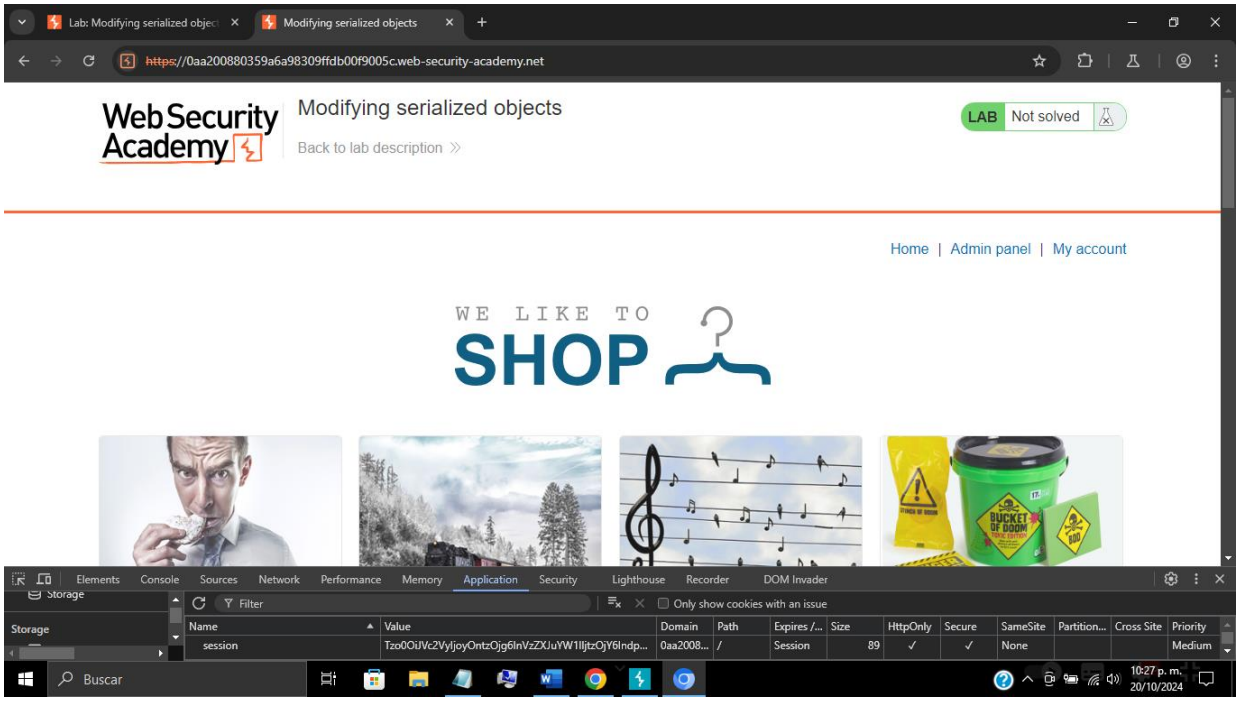
Posteriormente se codifica a Base64 para obtener el enlace modificado



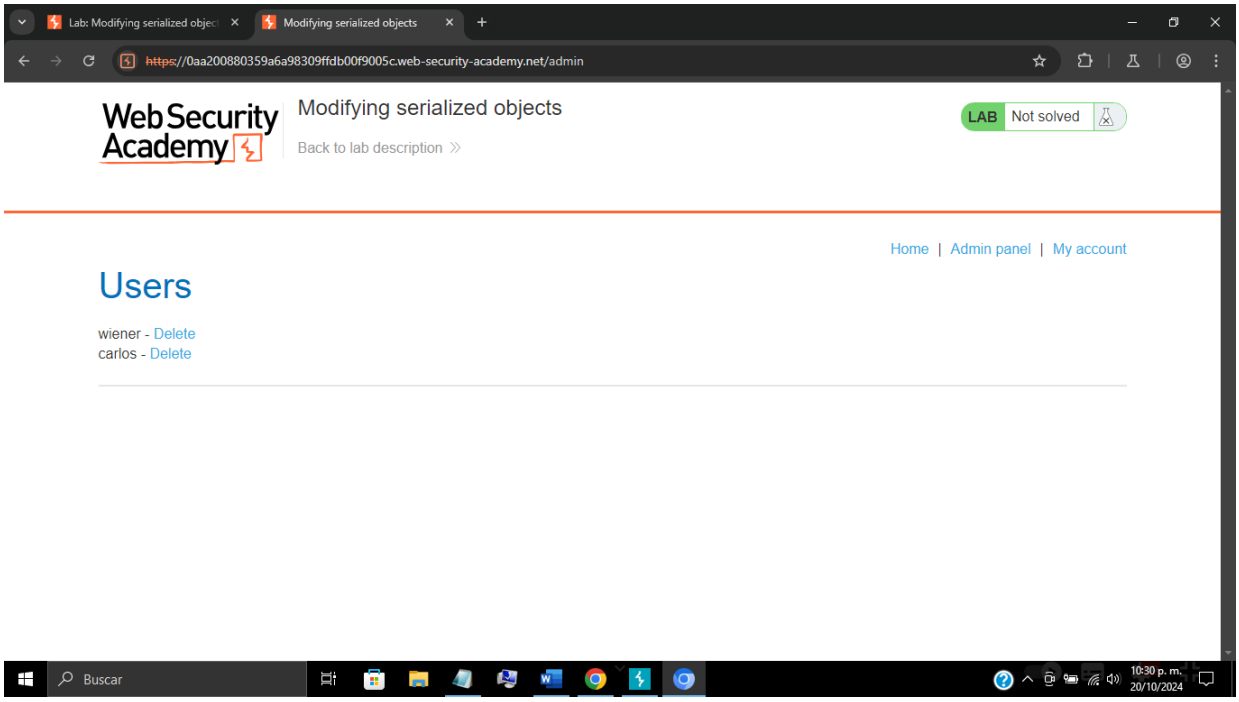
Y se modifica la terminación del enlace agregando los caracteres que se habían eliminado anteriormente %3d y se copia el enlace



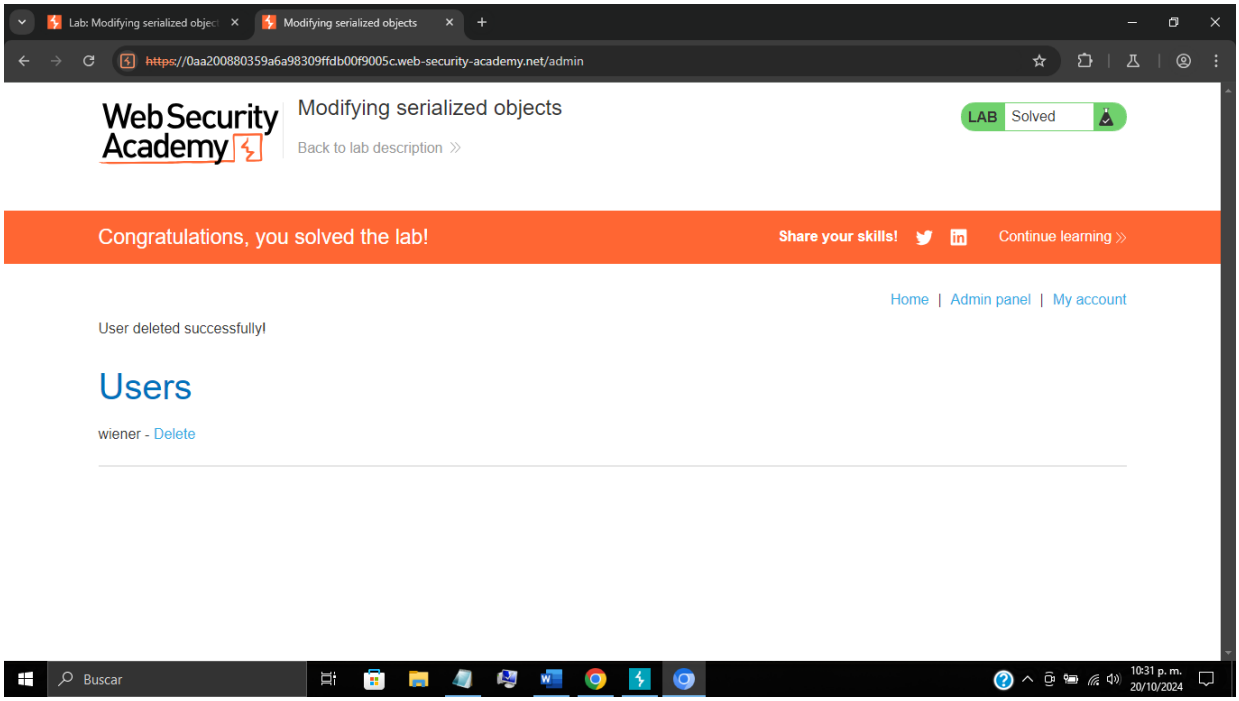
En el navegador se abren las herramientas para desarrolladores y se abren las propiedades de las cookies para pegar el enlace modificado en el campo value



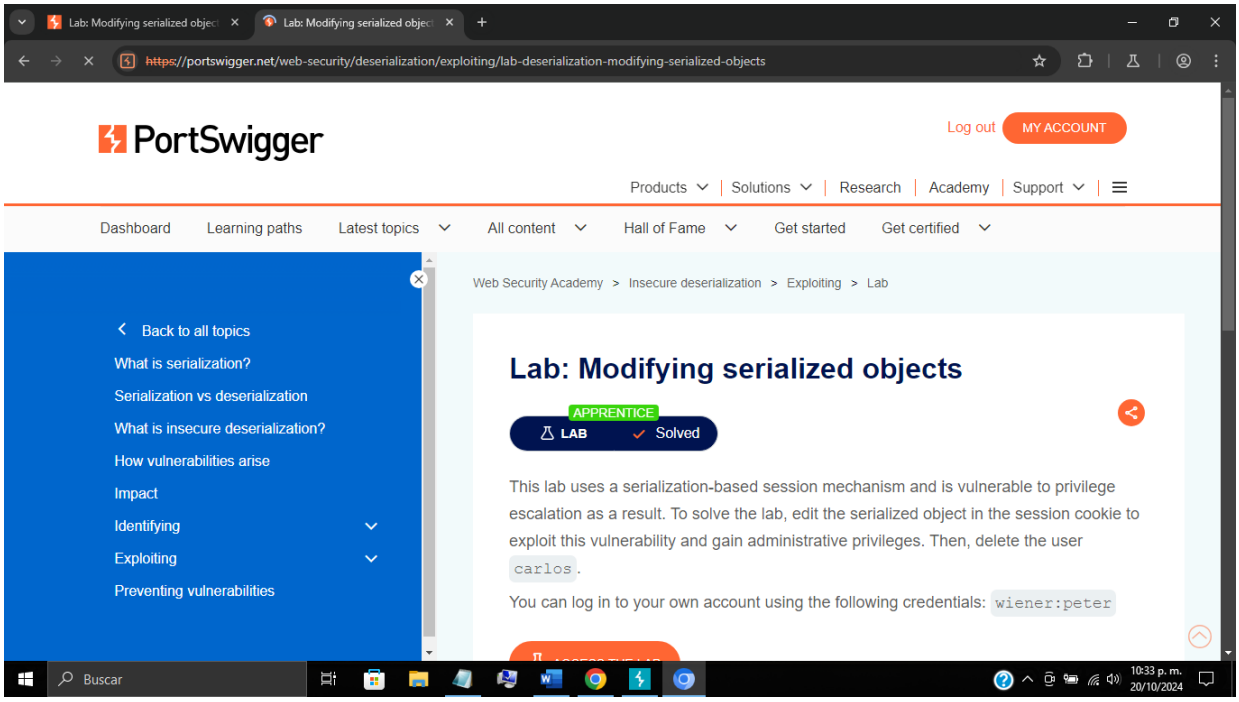
Se refresca la pagina para poder visualizar el panel de administrador



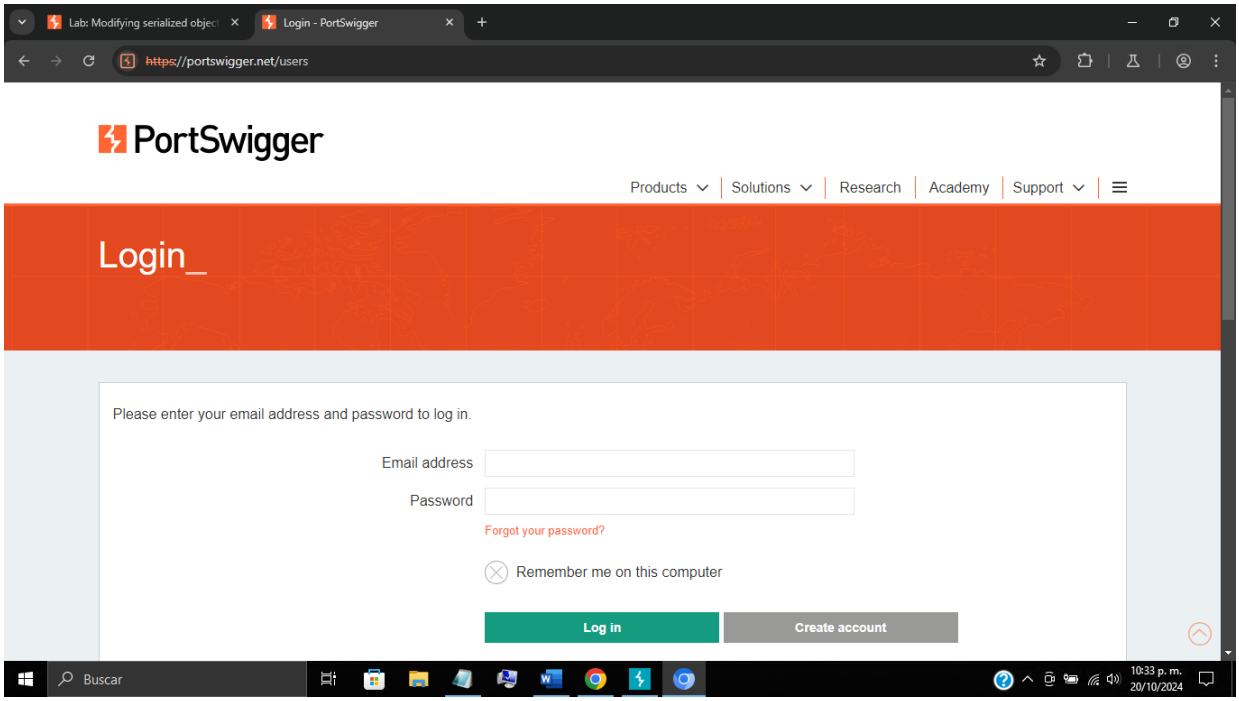
Se da click en el panel de administrador y nos muestra la lista de usuarios



Se elimina el usuario Carlos para concretar la prueba



Y se cierra sesión



Enlace de GitHub: https://github.com/Chifer888/Auditoria-informatica.git

Conclusión.

La deserialización insegura es una de las principales vulnerabilidades incluidas en las listas de amenazas de seguridad como OWASP Top 10, donde se supone que una aplicación web recibe datos de un cliente en formato JSON para deserializarlos y crear un objeto en el servidor y si la aplicación no valida adecuadamente ese JSON, el atacante puede modificar los datos para incluir un script malicioso y al deserializar los datos, el servidor ejecutaría ese script, causando daños significativos.

¿Qué aprendo?

Que es de suma importancia dar la prioridad adecuada que se merece la privacidad y seguridad al momento de diseñar páginas en la web, para que no sean vulnerables a intrusiones no deseadas, que en esta practica al ser la primera vez que utilizo Burp Suite Community Edition invertí demasiado tiempo para entender cómo funciona este proceso, y que una persona con los conocimientos necesarios le tomaría no más de cinco minutos para vulnerarla y modificar información privilegiada.

Referencias

ChatGPT. (n.d.). https://chatgpt.com/c/670f3505-b99c-8003-99e2-bdbfca4c3fcc

Caja de herramientas de Google Admin. (n.d.). https://toolbox.googleapps.com/apps/main/

Emanuele Picariello. (2022, June 29). *Insecure Deserialization vulnerabilities: Lab #1 by PortSwigger - Modifying Serialized Objects* [Video]. YouTube. https://www.youtube.com/watch?v=tm3u3Dw8I_4

Videoconferencias, conferencias web, seminario web, uso compartido de pantalla. (n.d.). Zoom. https://academiaglobal-mx.zoom.us/rec/play/My8s_p5AihIzs8Wmz9-adK9FlJnvJA9MLerRNwL-GGpk2ZzbFtR6o8iAhBb4YrkD4c5hgR43I_35RWqG.Vtbw-qG4_jDuIBec?canPlayFromShare=true&from=share_recording_detail&conti-nueMode=true&componentName=rec-play&originRequestUrl=https%3A%2F%2Facademia-global-mx.zoom.us%2Frec%2Fshare%2FKFTbRnIAl6ez2vvcpaLMiNdmhEwViz-BXYnxJlZ75njOevOCYS0xYB_dgHHFZ_L6y.dun0bhS2J89l5IYu