

Actividad | 2 | Deserialización insegura.

Auditoría Informática.

Ingeniería en Desarrollo de
Software.



academiaglobal

TUTOR: Jessica Hernández Romero

ALUMNO: Carlos Ariel Nicolini

FECHA: 15/10/2025

Índice

| | |
|------------------------------|-----------|
| Introducción | 3 |
| Descripción | 4 |
| Justificación | 5 |
| Ataque al sitio | 6 |
| Conclusión..... | 13 |
| Referencias..... | 14 |

Introducción

La deserialización es el proceso de convertir una estructura de datos o el estado de un objeto almacenado en un formato como Json, XML o binario en un objeto utilizable en memoria. Esta técnica se utiliza comúnmente en aplicaciones para la transmisión de datos o el almacenamiento de objetos en archivos. Es esencial en entornos de programación donde es necesario transferir, guardar o compartir objetos entre diferentes partes de una aplicación o entre diferentes aplicaciones. Sin embargo, puede generar vulnerabilidades de seguridad si no se maneja con cuidado.

Cuando la deserialización es insegura, utiliza datos no confiables para abusar de la lógica de una aplicación, lo que da lugar a diversos tipos de ataques, como ejecución remota de código, ataques de repetición, ataques de inyección y escalada de privilegios.

Para proteger las aplicaciones contra vulnerabilidades de deserialización es necesario tomar medidas importantes, algunas de las cuales pueden ser:

- Evitar la deserialización de datos de fuentes no confiables.
- Implementar la validación de entrada.
- Utilizar bibliotecas de serialización segura.
- Utilizar la autoprotección de aplicaciones en tiempo de ejecución (RASP).
- Utilizar un firewall de aplicaciones web (WAF).

Descripción

Contextualización:

Una empresa de software solicita realizar varias pruebas de seguridad en páginas web que no cuentan con los candados de seguridad.

Para esta segunda etapa, pide realizar una prueba de deserialización insegura en una página específica. Esta debe ser mediante las cookies. Para lograrlo, utilizar el programa Burp Suite Community Edition. El objetivo de esta prueba es que se inicie sesión como un usuario normal y luego pasar a modo administrador a través de las cookies.

Actividad:

Con la ayuda de la plataforma PortSwigger, realizar el ataque a una pagina proporcionada por ellos. En ella, iniciar sesión con las credenciales que se proporcionan, las cuales son para usuarios normales; no obstante, a través de las cookies, entrar al modo administrador.

Cabe mencionar que este laboratorio utiliza un mecanismo de sesión basado en serialización. Por ende, es vulnerable a la escalada de privilegios. En consecuencia, hay que editar el objeto serializado en la cookie de sesión para aprovechar esta vulnerabilidad y obtener privilegios administrativos. Finalmente, el objetivo es eliminar la cuenta de Carlos.

Hay que iniciar sesión en la propia cuenta con las siguientes credenciales:

- Usuario: wiener
- Contraseña: peter

Justificación

En esta segunda actividad de la materia, nos toca investigar y aprender sobre la deserialización insegura usando cookies, tenemos que entrar en una página con una cuenta normal y con ayuda del programa Burp Suite Community Edition usaremos la cookie de sesión, modificarla para pasar a modo administrador. El objetivo final es con esos derechos administrador eliminar la cuenta llamada Carlos para poder confirmar que la escalación de privilegios funcione de manera correcta.

Este ejercicio es una forma de enseñarnos como se deben diseñar los ambientes productivos y los mecanismos que deben tener para impedir esos riesgos de seguridad que pueden tener grandes implicaciones.

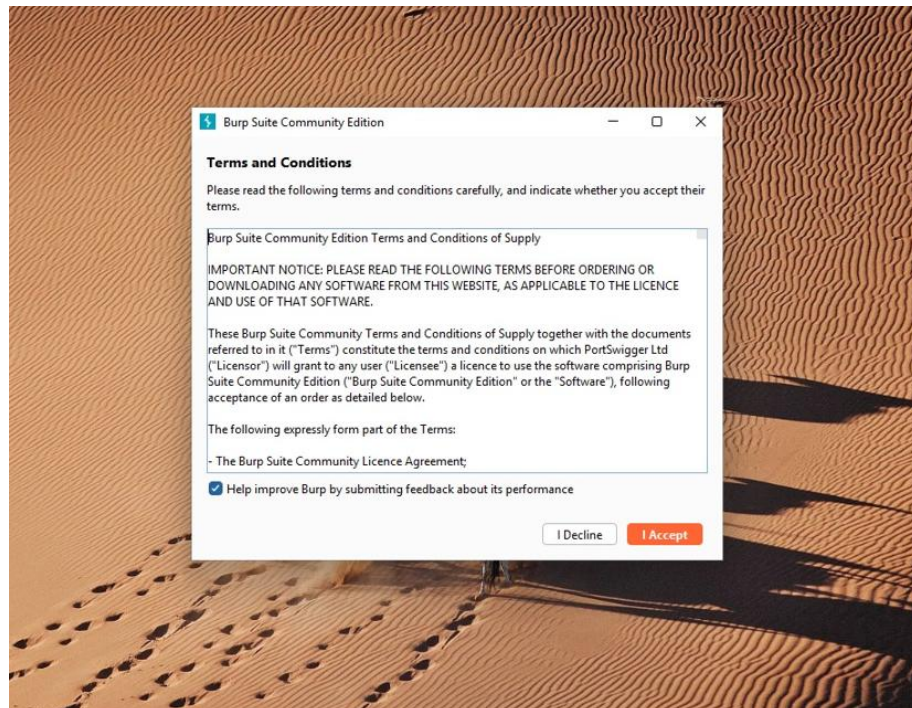
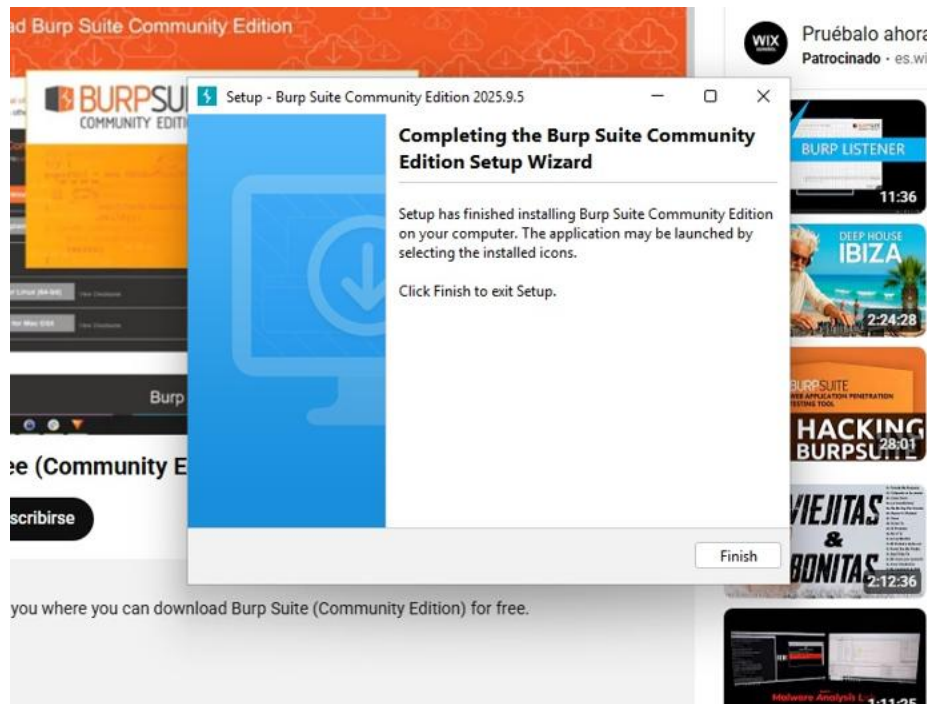
Cabe mencionar que es un ejercicio muy divertido, si me tomo un poco entender el proceso, pero una vez comprendido se pudo realizar sin mas problemas, lo cual me alegro y también me puso alerta de sobremanera, ya que, si las aplicaciones o sitios que utilizamos no están configurados de manera correcta, nuestros datos están comprometidos, nosotros sin darnos cuenta y de todas las implicaciones que eso puede generarnos.

Este trabajo fue subido al siguiente enlace de GitHub

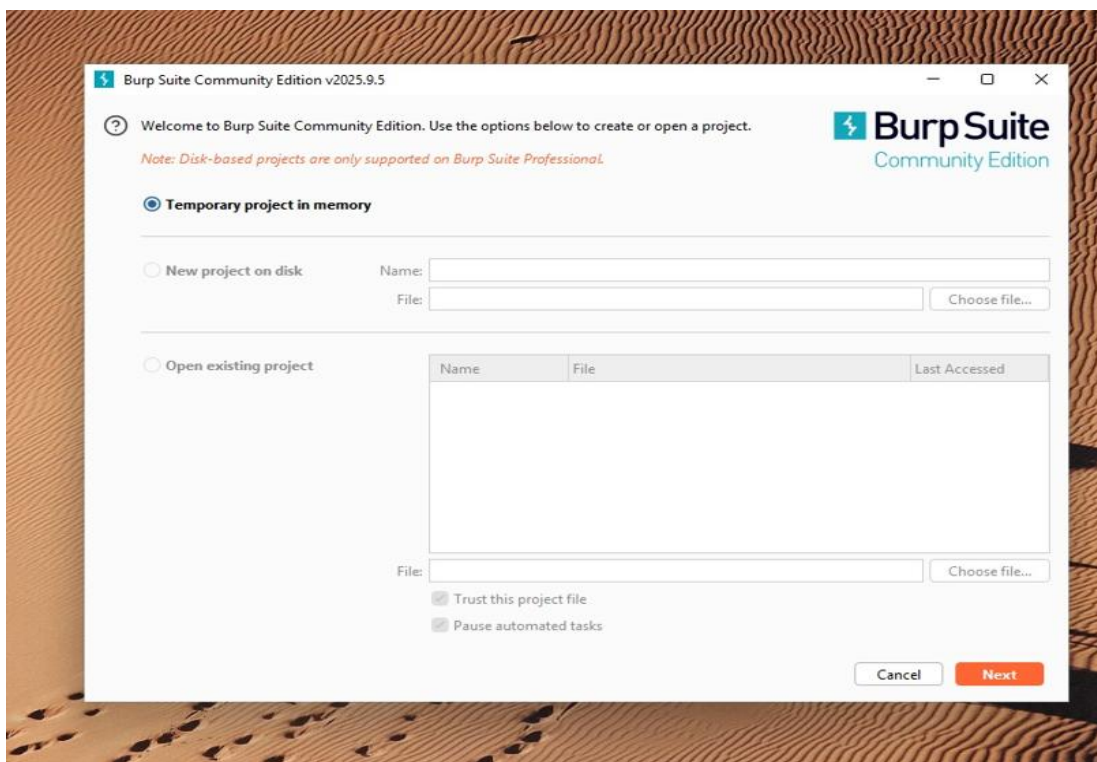
<https://github.com/CarlosNico/Auditor-a-Inform-tica>

Ataque al sitio

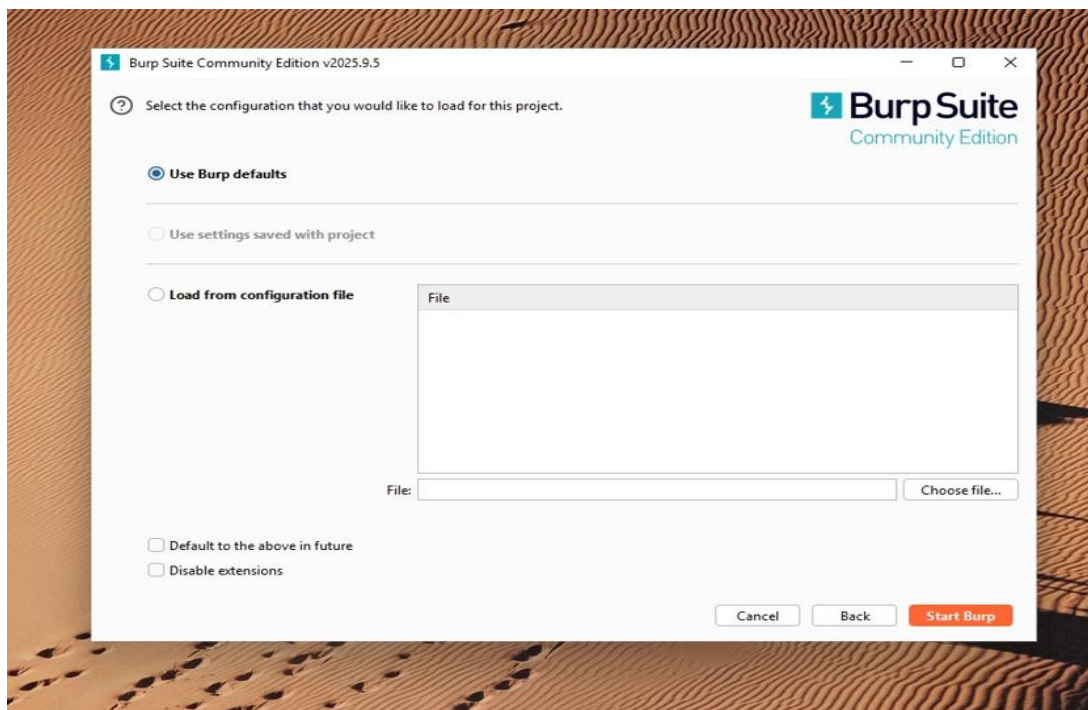
Para este ejercicio realizaremos un ataque a una página proporcionada en un laboratorio en la plataforma PortSwigger. Para tal motivo, ingresamos a la página donde descargaremos el software y lo instalaremos y lo ejecutamos como se muestra en las siguientes capturas.



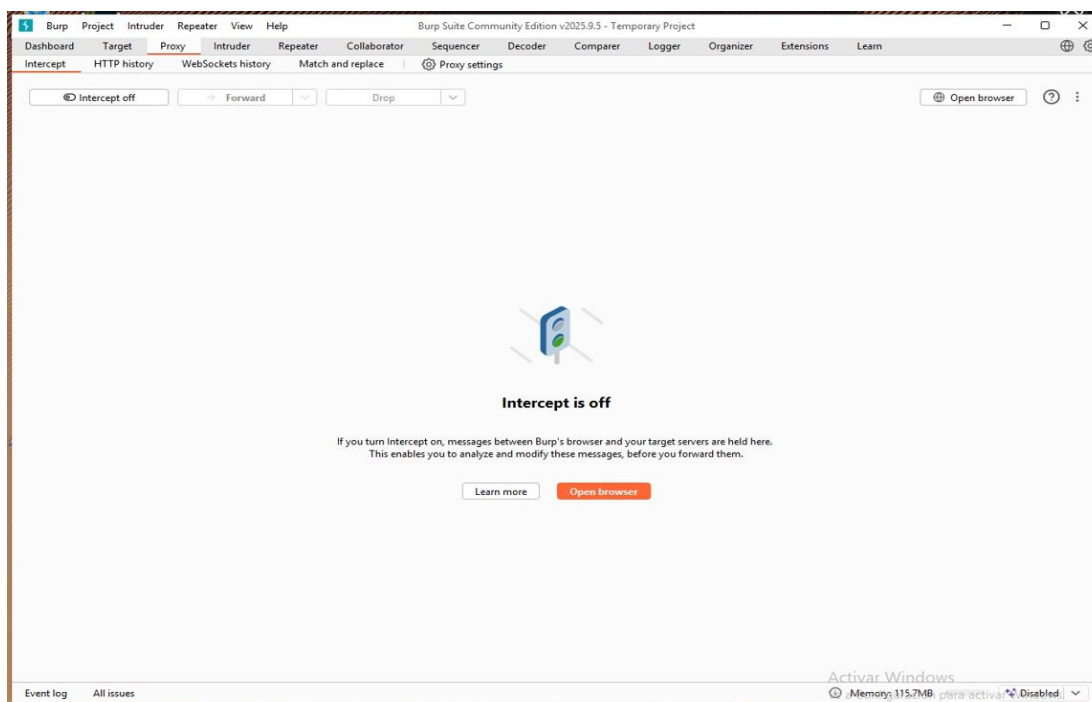
Cuando ingresamos al programa dejamos seleccionada la opción de default como se muestra en la imagen y presionamos next.



En la pantalla siguiente también dejamos seleccionada la opción default y seleccionamos Start Burp.



Con el programa ya abierto completamente, seleccionamos la pestaña de proxy, presionamos la opción Open browser, la cual nos abrirá un explorador donde pondremos el enlace del laboratorio compartido en el ejercicio (<https://portswigger.net/web-security/deserialization/exploiting/lab-deserialization-modifying-serialized-objects>).



Se nos abrirá la pagina del laboratorio y le damos Access the lab (debemos logearnos para poder realizarlo).

Lab: Modifying serialized objects

Web Security Academy > Insecure deserialization > Exploiting > Lab

Lab: Modifying serialized objects

APPRENTICE

LAB Not solved

This lab uses a serialization-based session mechanism and is vulnerable to privilege escalation as a result. To solve the lab, edit the serialized object in the session cookie to exploit this vulnerability and gain administrative privileges. Then, delete the user `carlos`.

You can log in to your own account using the following credentials:

```
wiener:peter
```

ACCESS THE LAB

Solution

1. Log in using your own credentials. Notice that the post-login `GET /my-account` request contains a session cookie that appears to be URL and Base64-encoded.
2. Use Burp's Inspector panel to study the request in its decoded form. Notice that the cookie is a fast-deserialized object. The object attribute

My Account - PortSwigger

Log out MY ACCOUNT

My Account

Personal Details

Carlos Nicolini
elfoludo@gmail.com
Change Password

Account Address

No address associated with this account

Saved Cards

+
Add new card

PortSwigger

Follow us

Burp Suite
 Web vulnerability scanner
 Burp Suite Editions
 Release Notes

Vulnerabilities
 Cross-site scripting (XSS)
 SQL injection
 Cross-site request forgery
 XML external entity injection


Customers
 Organizations
 Testers
 Developers


Company
 About
 Careers
 Contact
 Legal

Insights
 Web Security Academy
 Blog
 Research

Lab: Modifying serialized object x Modifying serialized objects x +


https://0a69007e03b943b1aaa77dc300740004.web-security-academy.net


WebSecurity Academy  Modifying serialized objects

LAB Not solved 


Back to lab description >>

Home | My account


WE LIKE TO SHOP 




Gym Suit
★★★★★ \$3.57
[View details](#)




Eggstastic, Fun, Food Eggcessories
★★★★★ \$55.49
[View details](#)





Picture Box
★★★★★ \$19.02
[View details](#)




What Do You Meme?
★★★★★ \$22.58
[View details](#)











Presionamos en My Account ingresaremos con los siguientes datos wiener:peter

Lab: Modifying serialized object x Modifying serialized objects x +

https://0a69007e03b943b1aaa77dc300740004.web-security-academy.net/login

WebSecurity Academy  Modifying serialized objects

LAB Not solved 

Back to lab description >>

Home | My account

Login

Username

Password

[Log in](#)

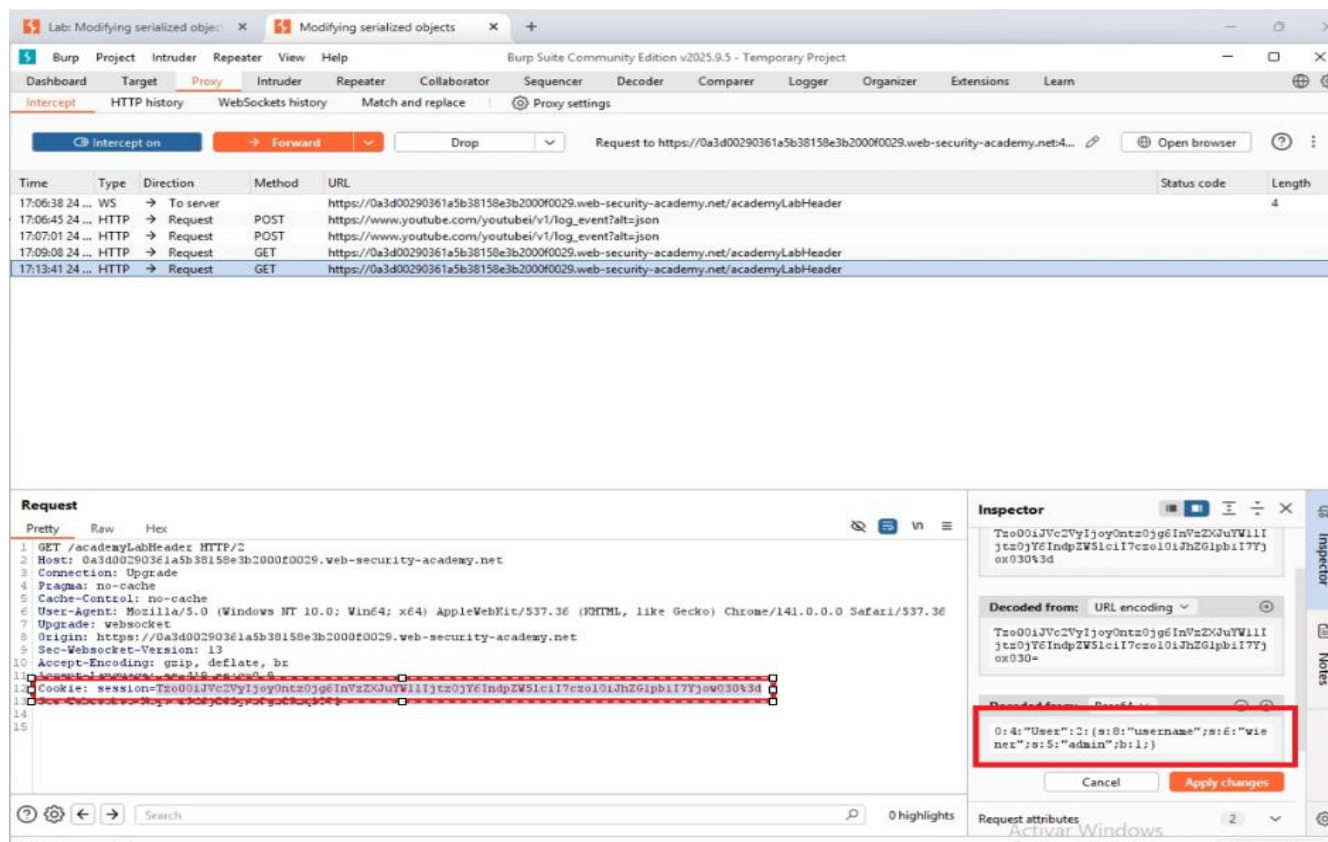
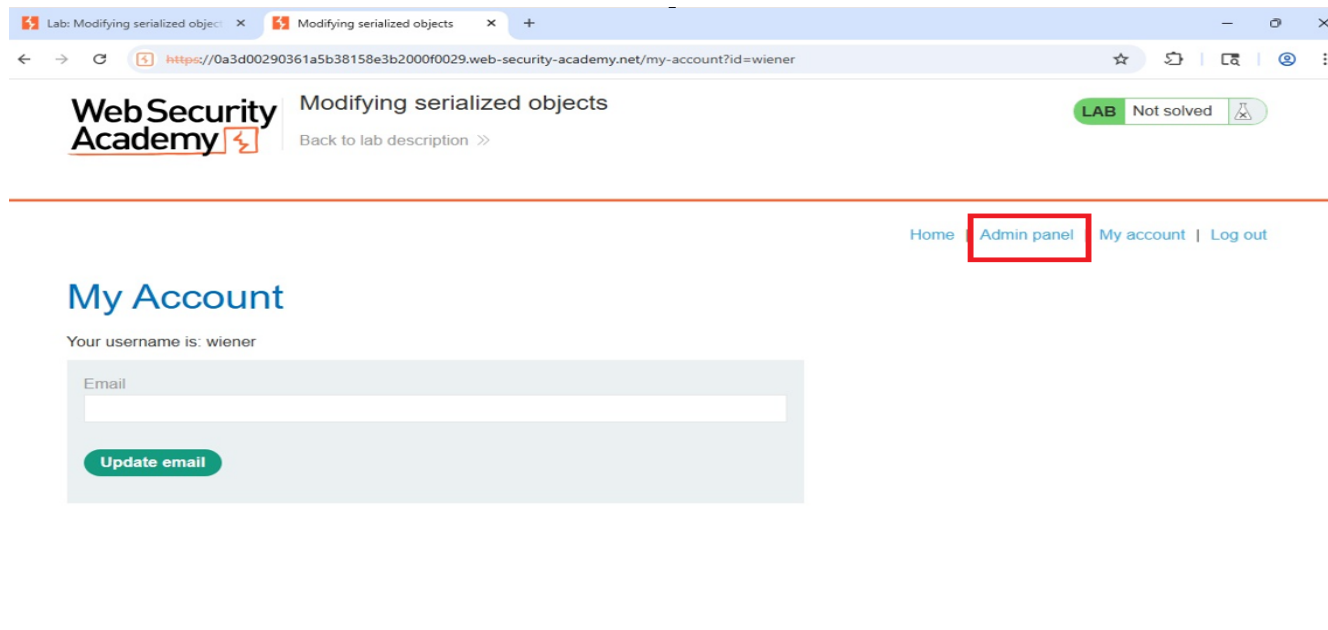
Una vez realizado el login encendemos la opción Intercept para empezar con la captura. Como se muestra en la siguiente imagen detectamos la cookie, el cual elegimos y se muestra en el decoded base64

The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. The 'Request' tab displays the raw HTTP request, and the 'Inspector' tab shows the decoded content. The cookie header is highlighted in the request, and its decoded value is shown in the inspector.

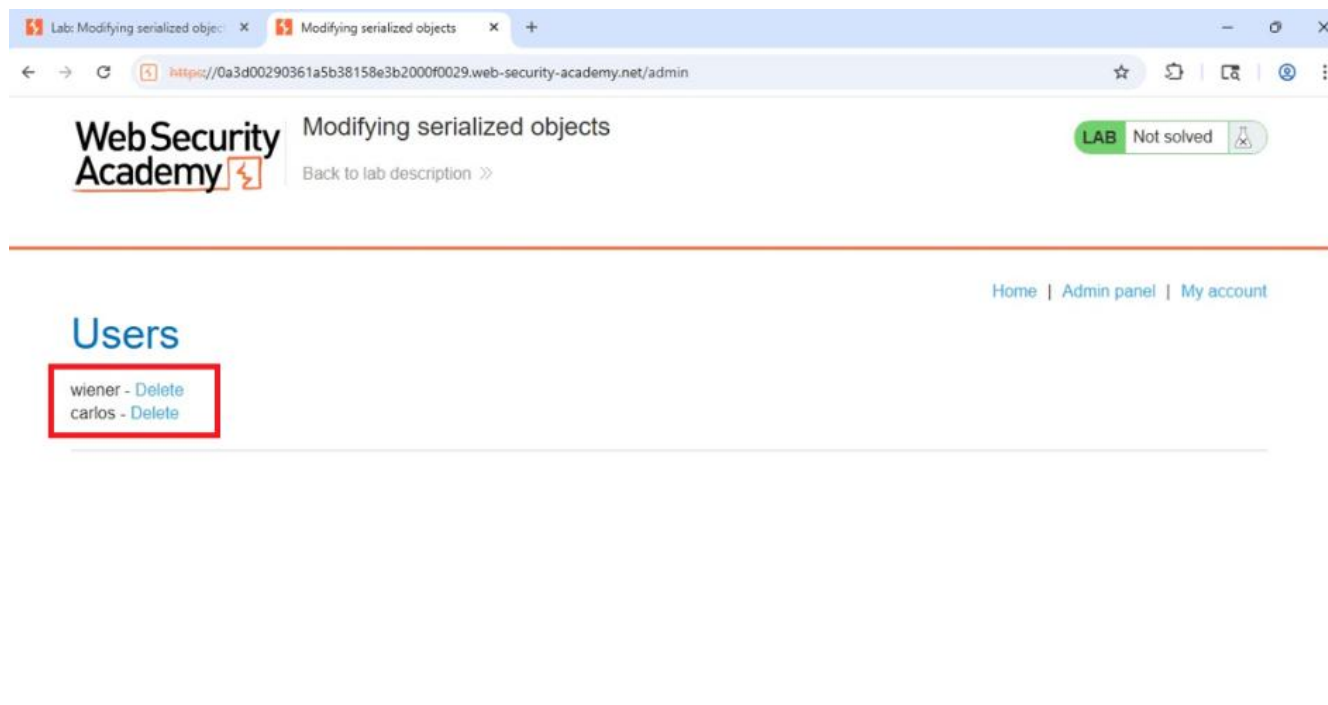
Esta decodificación `O:4:"User":2:{s:8:"username";s:6:"wiener";s:5:"admin";b:0;}` en palabras cortas es donde capturamos los datos del usuario Wiener (con el que ingresamos) y la parte final nos indica con un booleano 0, lo cual nos indica que no es administrador.

En este punto es donde realizaremos el ataque, al cambiar el 0 por el 1 (le diremos que somos administrador), aplicar el cambio y presionaremos en la opción de forward (esta al lado del botón Intercept).

Al realizar ese paso nos aparecerá en la pantalla la opción de Admin panel. A continuación, presionamos en ese botón y volvemos a la captura, donde buscaremos la cookie y realizaremos el mismo paso anterior, donde cambiaremos el booleano de 0 a 1, aplicaremos y presionaremos forward.



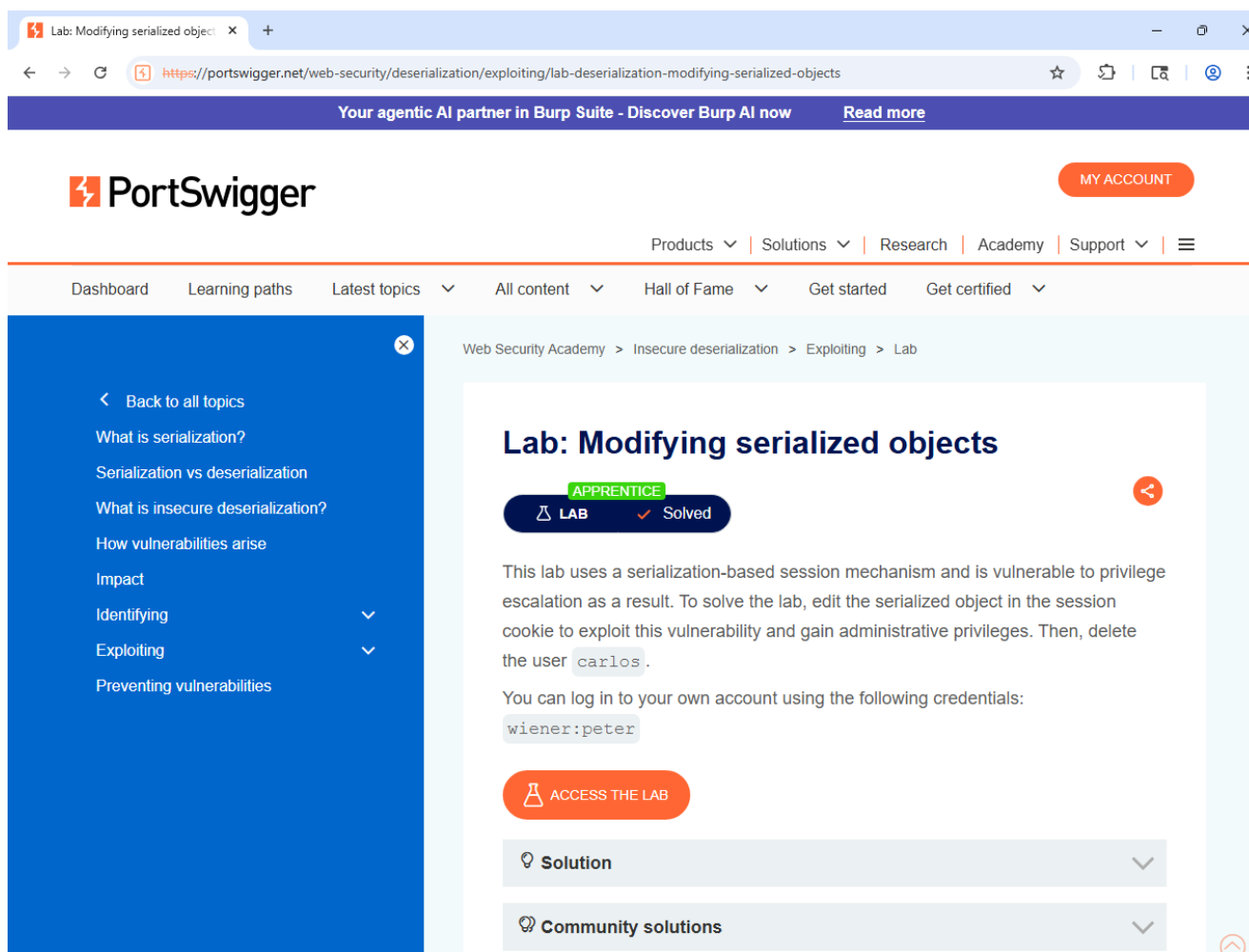
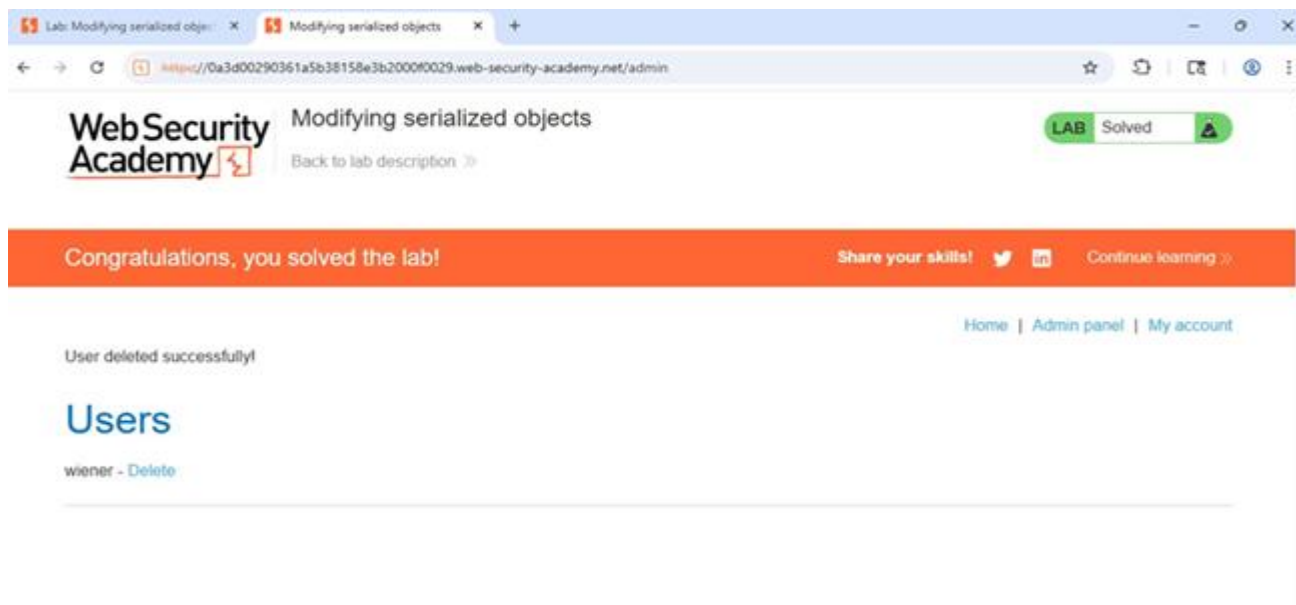
Al realizar ese cambio ingresaremos al panel de administración y podremos ver los usuarios en la pagina con nuestros derechos de administrador por el ataque a la cookie.



A continuación, realizaremos el mismo proceso, presionaremos en eliminar el usuario carlos, iremos a la captura, seleccionaremos la cookie, cambiaremos el valor del booleano a 1, aplicaremos el cambio y presionaremos forward (es posible que se deba realizar varias veces, hay que realizarlo más veces necesarias).

Al terminar el proceso veremos que el usuario carlos fue eliminado y nos da un mensaje de felicitaciones por haber terminado el ejercicio como fue solicitado.

Se adjuntan las imágenes donde señala que el ejercicio fue resuelto como fue solicitado por la profesora.



Conclusión

Realizar este ejercicio es fundamental para entender la necesidad de poner la atención que requiere la seguridad al desarrollar aplicación o sitios web, ya que como quedó demostrado en este trabajo, un diseño e implementación insegura puede poner en riesgo no solo la aplicación sino los datos personales de las personas que lo utilizan, lo que puede terminar en pérdida de información, pérdidas monetarias, impacto en la credibilidad de la marca, etc.

Entender y poder solucionar estos detalles de mal diseño a nivel empresarial son de un valor importante, ya que te da herramientas para poder justificar mejoras, lo cual aumenta tu credibilidad y puede abrir tus puertas en el mercado. A nivel personal, como usuario que utiliza sitios te vuelve más consciente de como debes utilizar tus contraseñas y cómo comportarte en los sitios.

Si después de realizar este ejercicio uno esta mas consciente de como debe cuidar sus datos en línea, ya que no son algo ahí que esta sin causar efectos, esos datos pueden costarle a uno mucho dinero y más, aunque también da un poco de miedo. Utilizar ese laboratorio fue muy divertido y entretenido, ojalá pueda tener mas casos donde pueda seguir utilizándolos y aprendiendo.

Este trabajo fue subido al siguiente enlace de GitHub

<https://github.com/CarlosNico/Auditor-a-Inform-tica>

Referencias

Insecure deserialization. (n.d.). Portswigger.net. Retrieved October 25, 2025, from <https://portswigger.net/web-security/deserialization>

Kumar, R., McKeever, G., Wright, M., Hasson, E., Cheng, L., Rohit Kumar, Guillotin, E., & Muly Levy. (n.d.). *Deserialization.* Learning Center; Imperva Inc. Retrieved October 25, 2025, from <https://www.imperva.com/learn/application-security/deserialization/>

Lab: Authentication bypass via flawed state machine. (n.d.). Portswigger.net. Retrieved October 25, 2025, from <https://portswigger.net/web-security/logic-flaws/examples/lab-logic-flaws-authentication-bypass-via-flawed-state-machine>