UNIVERSIDAD AUTONÓMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Desarrollo de Software Seguro

Actividad práctica (asíncrona): Cross-Site Request Forgery

Estudiante: Osmar Abelardo Bustos Vázquez

Carrera: Lic. en Seguridad de Tecnologías de Información

Matrícula: 1912361

Groupo: 064

Docente: M.C. Romeo Alfonso Sanchez Lopez

19 de octubre 2023

AGOSTO-DICIEMBRE-2023

Cross-Site Request Forgery

Enlace de página en Github: https://github.com/osm4r/DDSCross-Site-Request-Forgery

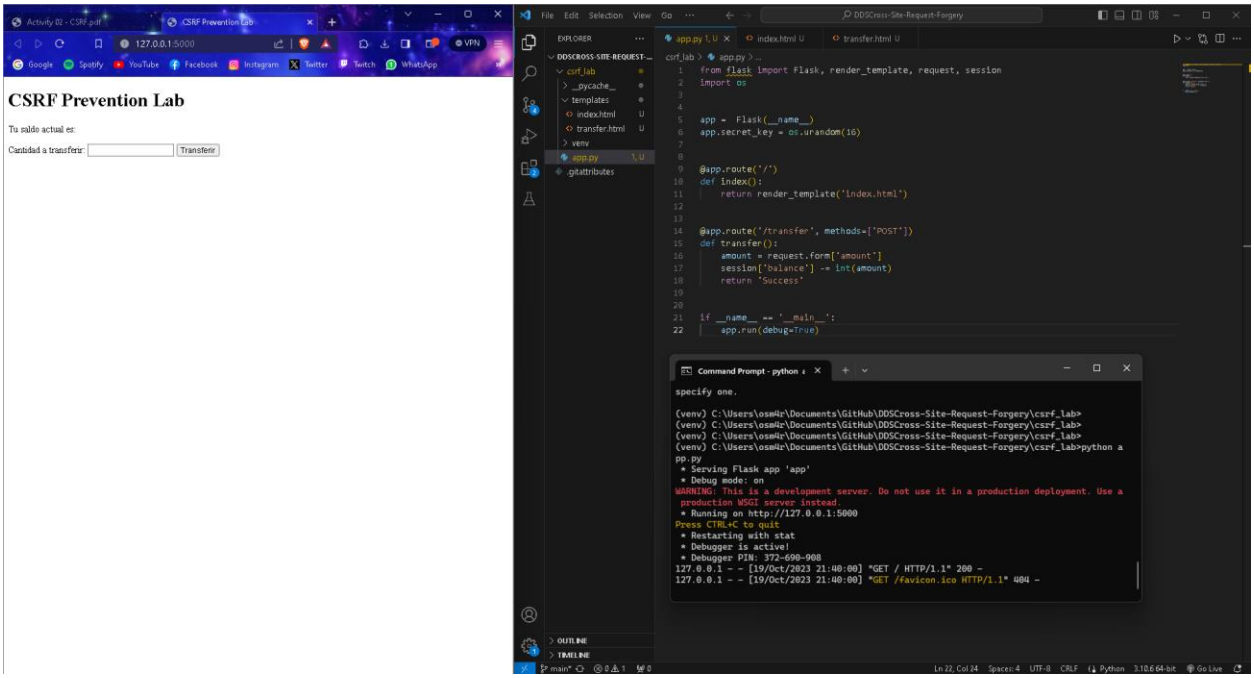
Aprendizajes:

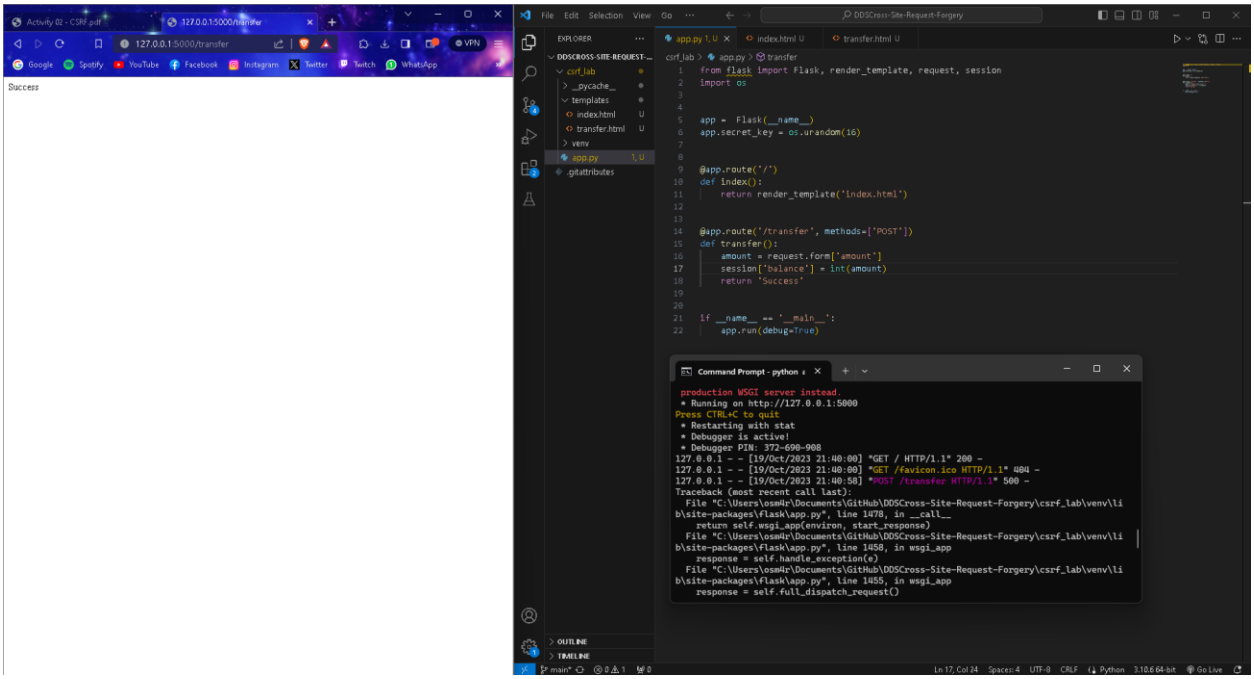
- Uso de Flask-WTF
- Se reforzó el conocimiento de Cross-Site Request Forgery (CSRF)
- Uso de tokens CSRF
- Prevenir ataques CSRF

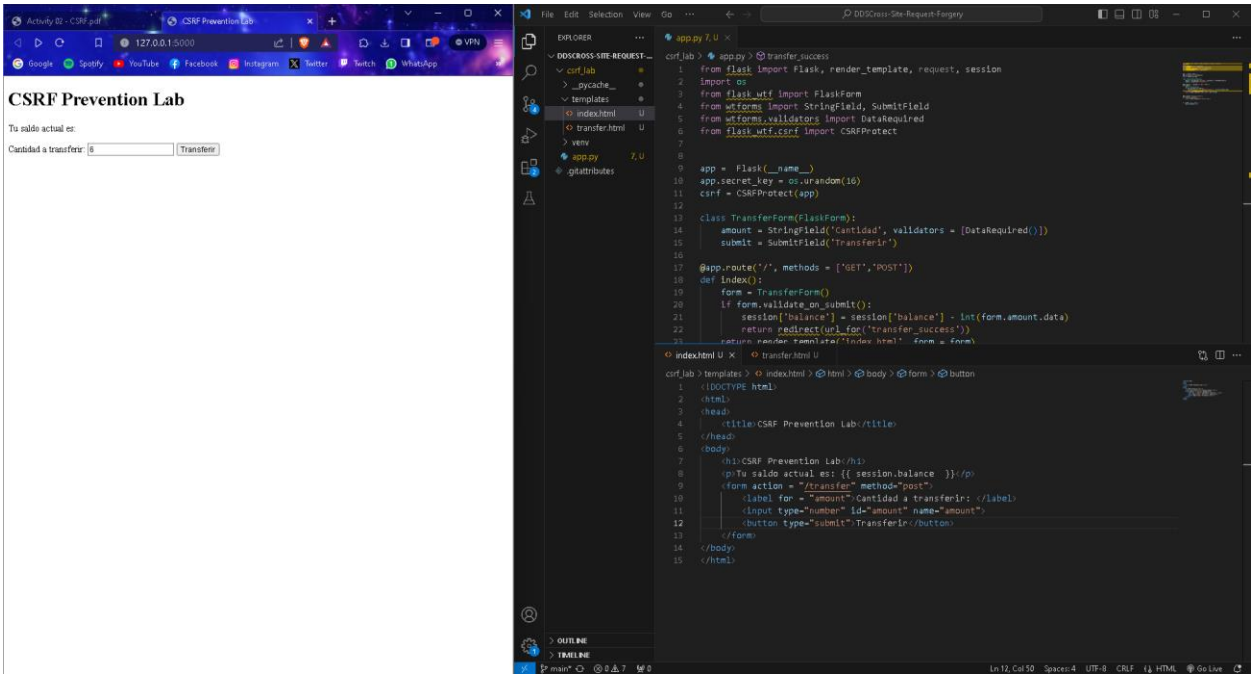
Dificultades:

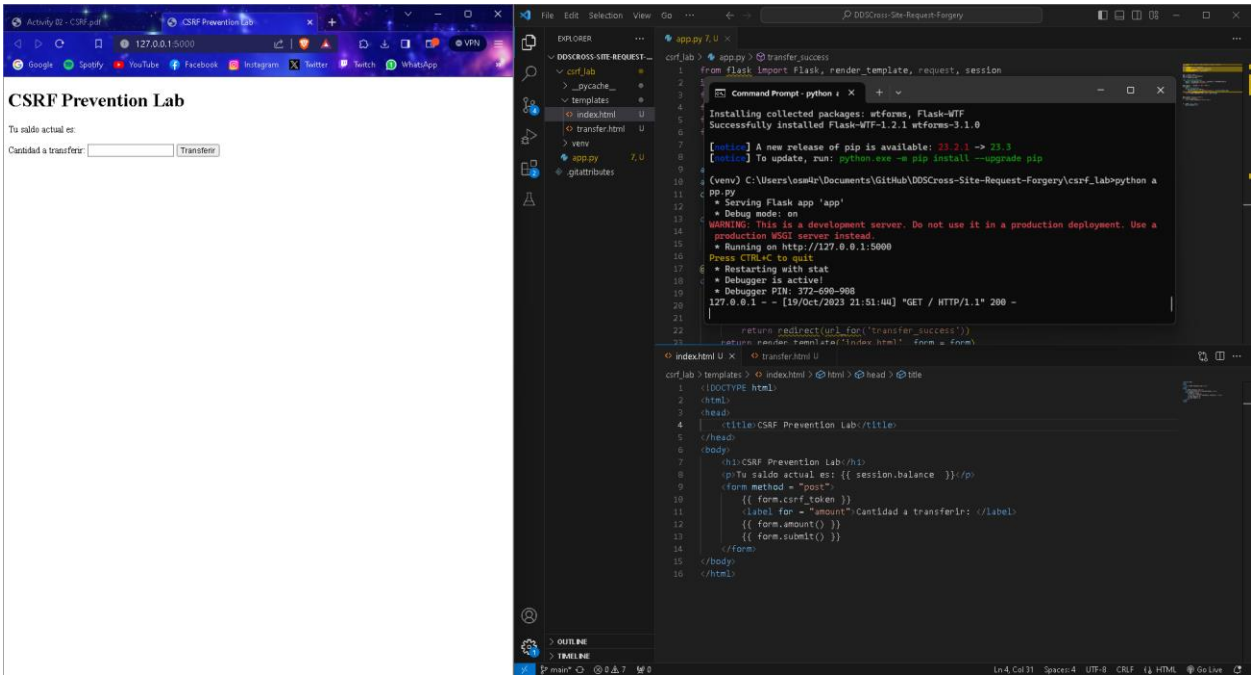
Tuve un poco de problema con la forma en que se almacena el balance, que no te detecta cuando aún no has realizado una transferencia y te manda un error. Lo pude solucionar gracias a una plática con mis compañeros en la cual investigamos que podría ser y utilizamos una validación para declarar el balance en 0 cuando sea la primera vez que se va a hacer una transferencia.

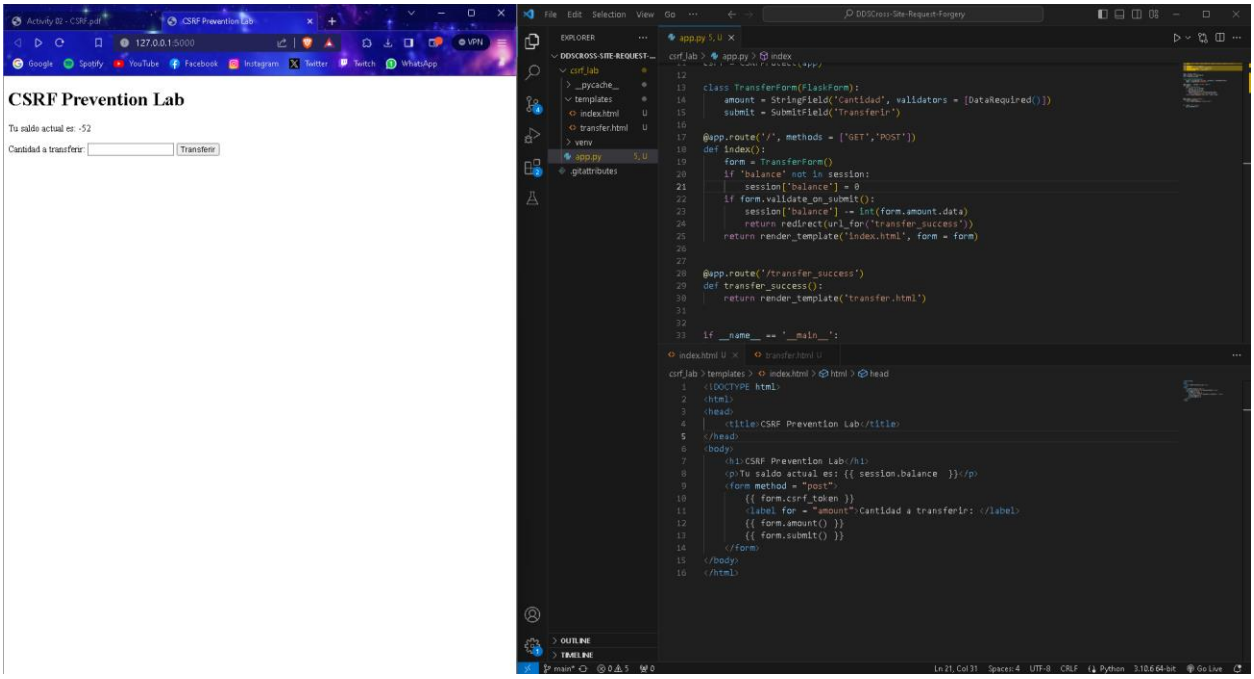
Pruebas y capturas de pantalla:

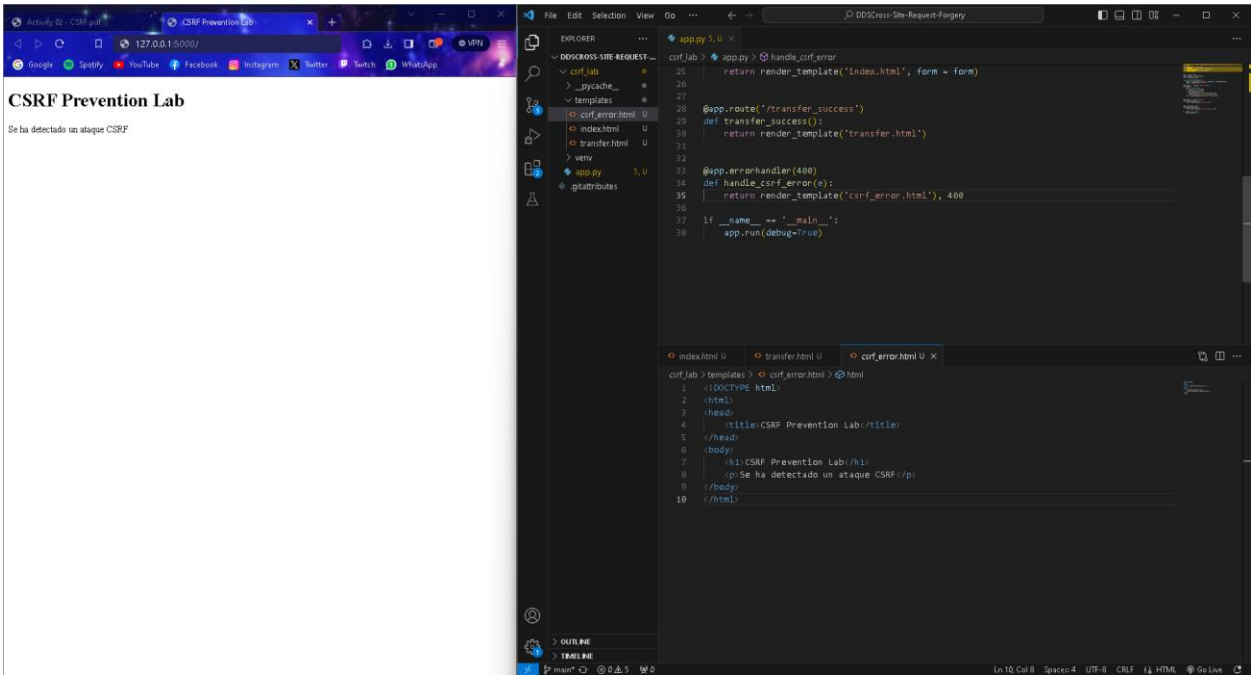


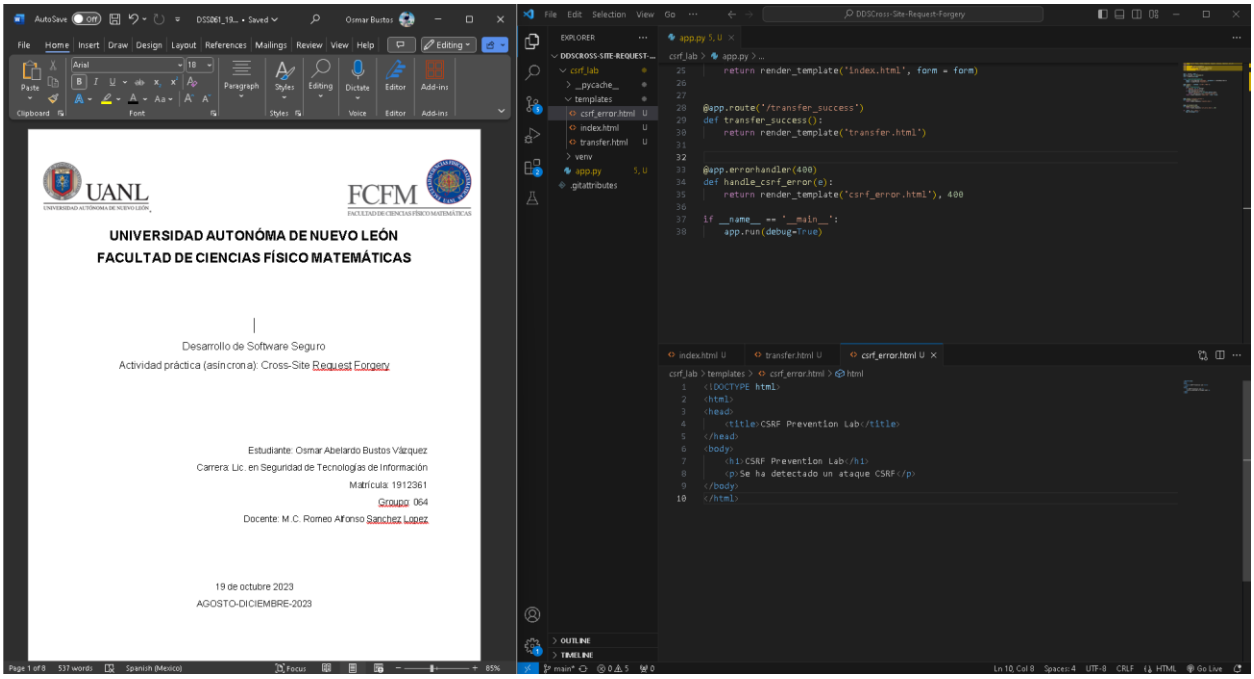

Success
app.py
index.html
transfer.html
Command Prompt - python
production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 372-690-908
127.0.0.1 - - [19/Oct/2023 21:40:00] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [19/Oct/2023 21:40:00] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [19/Oct/2023 21:40:58] "POST /transfer HTTP/1.1" 500 -
Traceback (most recent call last):


CSRF Prevention Lab
Tu saldo actual es:
Cantidad a transferir: 6
Transferir
app.py
index.html
transfer.html


CSRF Prevention Lab
Tu saldo actual es:
Cantidad a transferir:
Transferir
Installing collected packages: wtforms, Flask-WTF
Successfully installed Flask-WTF-1.2.1 wtforms-3.1.0
(venv) C:\Users\osm4r\Documents\GitHub\DDSCross-Site-Request-Forgery\csrf_lab>python a
pp.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a
production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 372-690-908
127.0.0.1 - - [19/Oct/2023 21:51:44] "GET / HTTP/1.1" 200 -


CSRF Prevention Lab
Tu saldo actual es: -52
Cantidad a transferir:
Transferir


CSRF Prevention Lab
Se ha detectado un ataque CSRF
app.py
@app.route('/transfer_success')
def transfer_success():
return render_template('transfer.html')
@app.errorhandler(400)
def handle_csrf_error(e):
return render_template('csrf_error.html'), 400
if __name__ == '__main__':
app.run(debug=True)
<!DOCTYPE html>
<html>
<head>
<title>CSRF Prevention Lab</title>
</head>
<body>
<h1>CSRF Prevention Lab</h1>
Se ha detectado un ataque CSRF
</body>
</html>


UANL
FCFM
UNIVERSIDAD AUTONÓMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS
Desarrollo de Software Seguro
Actividad práctica (asíncrona): Cross-Site Request Forgery
Estudiante: Osmar Abelardo Bustos Vázquez
Carrera: Lic. en Seguridad de Tecnologías de Información
Matrícula: 1912361
Grupo: 064
Docente: M.C. Romeo Alfonso Sanchez Lopez
19 de octubre 2023
AGOSTO-DICIEMBRE-2023

Preguntas:

1. ¿Qué es Cross-Site Request Forgery (CSRF)?

Es un ataque malicioso en el cual un atacante engaña a un usuario para que realice acciones no deseadas en una aplicación web en la que el usuario está previamente autenticado.

2. ¿Cuál es el propósito de incluir un token CSRF en un formulario web?

Principalmente, para mitigar los ataques CSRF. Cuando un usuario envía un formulario el servidor verifica que el token enviado coincida con el token almacenado en el lado del servidor para comprobar que no haya sido falsificado.

3. ¿Cómo se implementa la prevención de CSRF en una aplicación Flask usando FlaskWTF?

Mediante la utilización de la función validate_on_submit(), para validar los datos y analizar cómo se va actualizando el saldo.

4. ¿Cómo se puede interceptar una solicitud maliciosa para realizar un ataque CSRF?

Los atacantes suelen utilizar burpsuite, programa utilizado en esta práctica y en materias pasadas el cual que sirve para eso.

5. ¿Cómo se manejan los errores 400 en Flask cuando se detecta un ataque CSRF?

Utilizando un error handler que nos sirva para personalizar la respuesta cuando se detecta un ataque CSRF. Con este puedes redirigir al usuario a una página de error personalizada o incluso realizar otras acciones.

Reflexión final (conclusión):

En lo personal, me pareció muy interesante esta actividad ya que seguimos con el uso de Flask en Python, lo cual me gusta demasiado ya que es un lenguaje de programación con el cual estoy muy relacionado, y es uno de mis lenguajes favoritos, tanto que lo utilizo para optimizar tareas cotidianas que no son relacionadas a un ambiente educativo ni profesional.

Con esta práctica me llevo de aprendizaje el uso de FLASK_WTF para validar los inicios de sesión en los proyectos con Python, lo cual nos sirve demasiado para aplicar la seguridad de la información, es decir, mantener la confidencialidad, integridad y disponibilidad de la información.

También, el uso de token me parece algo muy importante ya que nos ayuda a cumplir nuestro objetivo de crear aplicaciones seguras.

Por último, me gustaría agregar que fue no fue complicada oa actividad, sino más bien fue un poco tediosa. Pero nada del otro mundo. Por esto me gustaría seguir teniendo actividades con este lenguaje.

Bibliografía:

SecureFlag. (2023, October 13). Cross-Site Request Forgery in python. *SecureFlag Security Knowledge Base*. https://knowledge-base.secureflag.com/vulnerabilities/cross_site_request_forgery/cross_site_request_forgery_python.html