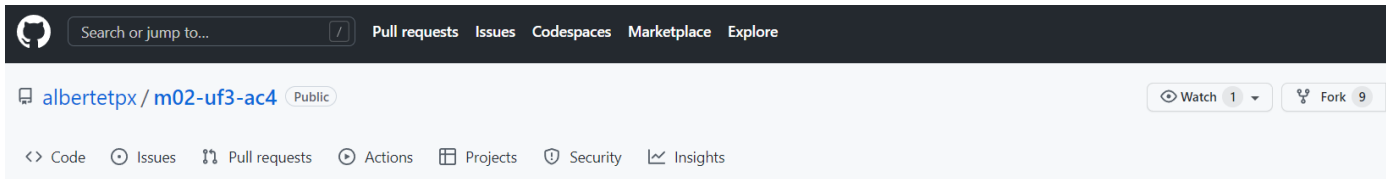


Cristina Claver Luna

Tasca 1

Anàlisis d'una situació d'injecció de codi:

a) Fes un fork del següent repositori al teu compte de GitHub. A continuació, clona'l a la teva màquina local



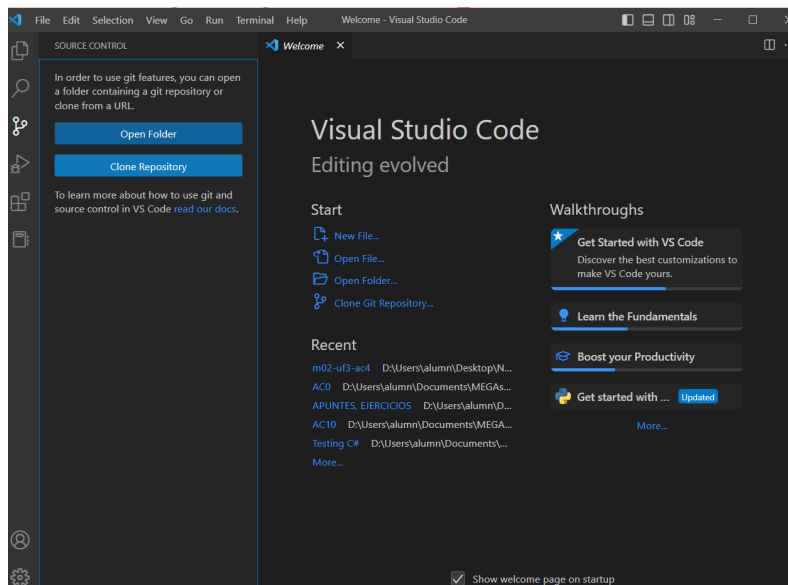
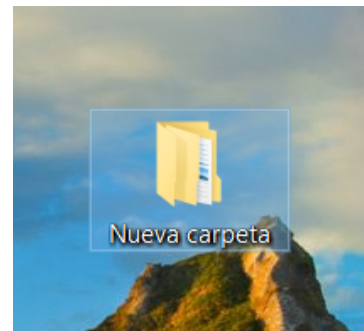
Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

No more forks can be created. These forks already exist:

[faisandelviento/m02-uf3-ac4](#)

[View all existing forks](#)

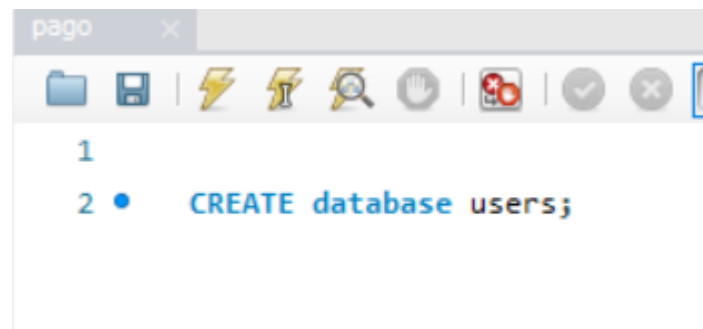


b)Fes les següents operacions per poder desplegar l'aplicació web que s'adjunta (formulario.rar).

- A L'APLICACIÓ FLASK: canvia els paràmetres de connexió a la base de dades (en concret, l'usuari i la contrassenya) perquè pugui connectar al teu servidor MySQL.

```
def connectBD():  
    db = mysql.connector.connect(  
        host = "localhost",  
        user = "root",  
        passwd = "Prin[REDACTED]",  
        database = "users"  
    )  
    return db
```

- AL SERVIDOR MYSQL: crea la base de dades users. No cal que creis cap taula; serà creada per la pròpia aplicació web (observa la funció



The screenshot shows a MySQL command window with a toolbar at the top containing icons for file operations and execution. The command prompt shows the following text:

```
1  
2 • CREATE database users;
```

c)Executa l'aplicació (app.py) i comprova que arrenca sense errors. Obre el navegador a <http://localhost:5000>, i comprova que:

Aplicació web amb base de dades (M02-UF3-AC4)

Log in to application

Enviar

- L'usuari user01 amb contrassenya admin pot fer login correcte i consultar les seves dades.

Log in to application

Enviar

LOGIN CORRECTO

User	Name	Surname 1	Surname 2	Age	Genre
user01	Ramón	Sigüenza	López	35	H

- L'usuari user01 amb contrassenya 1234 fa un login incorrecte

Log in to application

user01
1234
Enviar

LOGIN INCORRECTO

- d) Prova a autenticar l'usuari user01 i la contrassenya ' OR 1=1; (valor exacte).

Log in to application

user01
' OR 1=1;
Enviar

LOGIN CORRECTO

User	Name	Surname 1	Surname 2	Age	Genre
user01	Ramón	Sigüenza	López	35	H

- e) Explica què ocorre, i per què estem davant d'una situació d'injecció de codi.

En el programa de python hay una query que comprueba si la contraseña y el usuario coinciden con los datos de la BD, el usuario escribiendo el la contraseña 'OR 1=1' ha insertado código que modifica la query añadiendo la condición de que si 1 es 1 el login sea correcto, asi pudiendo acceder a cualquier dato.

Es una inyeccion de código poruqe los datos son tratados por el programa como código y mpdifican la funcion original.

f)Reimplementa la funció checkUser perquè faci servir una sentència parametritzada que eviti la situació d'injecció de codi:

```
query=f"SELECT user,name,surname1,surname2,age,genre FROM users WHERE user=%s AND password=%s"
values = (user, password)
print(query)
cursor.execute(query,values)
userData = cursor.fetchall()
bd.close()
```

g)Comprova que, ara, el formulari de login ja no és vulnerable a la injecció de codi.

LOGIN INCORRECTO

h)Explica per què la instrucció parametritzada resol la vulnerabilitat.

Esto separa el código o queries de los datos, asegurando que los datos sean tratados exclusivamente como datos y no como código.

Tasca 2

Completa l'aplicació web amb la funcionalitat de poder crear nous usuaris:

a)Crea una altra plantilla (signin.html), seguint l'estructura de login.html. Aquesta pàgina haurà de contenir un formulari de registre d'usuari, en que es pugui donar d'alta un usuari amb: nom d'usuari, contrassenya, nom, cognom1, cognom2, edat i salari.

Aplicació web amb base de dades (M02-UF3-AC4)

Formulario

usuari
contrassenya
nom
cognom1
cognom1 2
edat
no especificat

Enviar

Aplicació web amb base de dades (M02-UF3-AC4)

SING IN

Enviar

ETPX 2022-2023

```
<body>
  <header>
    <h1>Aplicació web amb base de dades (M02-UF3-AC4)</h1>
  </header>
  <main>
    <form action="{{url_for('newUser')}}" method="POST" class="formulario">
      <p>Formulario</p>

      <input type="text" name="user" placeholder="usuari"/>
      <input type="password" name="password" placeholder="contrasenya"/>
      <input type="text" name="name" placeholder="nom"/>
      <input type="text" name="surname1" placeholder="cognom1"/>
      <input type="text" name="surname2" placeholder="cognom1j2"/>
      <input type="number" name="age" placeholder="edat"/>
      <select name="genre">
        <option value="D">dona</option>
        <option value="H">home</option>
        <option value="NS/NC">no especificar</option>
      </select>
      <input type="submit" value="Enviar">
    </form>
  </main>
  <footer>
    <p>ETPX 2022-2023</p>
  </footer>
</body>
```

b) Modifica la ruta “/signin” a l’aplicació flask per a que mostri la plantilla signin.html que acabes de crear.

```
96 @app.route("/signin")
97 def signin():
98     return render_template("signin.html")
```

c) Crea una nova ruta (“/newUser”) a l’aplicació flask per a rebre i processar les dades del formulari de registre. T’hauràs d’inspirar en la ruta “/results” ja existent. Des d’aquesta ruta, crida la funció createUser.

```
@app.route("/newUser", methods=('GET', 'POST'))
def newUser():
```

```
1 @app.route("/newUser", methods=('GET', 'POST'))
2 def newUser():
3     print("LLEGAMOS A NEW USER")
4     if request.method == ('POST'):
5         formData = request.form
6         user=formData['user']
7         password=formData['password']
8         name=formData['name']
9         surname1=formData['surname1']
10        surname2=formData['surname2']
11        age=formData['age']
12        genre=formData['genre']
13        userData = createUser(user,password,name,surname1,surname2,age,genre)
14
15        if userData == False:
16            return render_template("signinresults.html",login=False)
17        else:
18            return render_template("signinresults.html",login=True,userData=userData)
19    #return "NEW USER"
```

d) Associa l’acció del formulari de registre (atribut action) a la nova ruta que acabes de crear. Observa com es fa en el formulari de login.

```
<main>
<form action="{{url_for('newUser')}}" method="POST" class="formulario">
    <p>Formulario</p>
```

d) Implementa la funció createUser perquè s'escriguin les dades del nou usuari en la base de dades. Utilitza sentències parametritzades.

```
# createUser: crea un nuevo usuario en la BD
def createUser(user,password,name,surname1,surname2,age,genre): ###Cristina
    ##comprobar usuario no esta en uso aun
    if(UsuarioNoExiste(user)):
        bd=connectBD()
        cursor=bd.cursor()
        ##insertar datos en BD
        query=f"INSERT INTO users VALUES (%s,%s,%s,%s,%s,%s,%s,%s);"
        print(query)
        values= (user,password,name,surname1,surname2,age,genre)
        cursor.execute(query,values)
        ##commit para que los datos de la BD se actualicen antes de hacer siguiente consulta
        bd.commit()

        #comprobamos que user existe en BD y seleccionamos los datos
        query2= f"SELECT user,name,surname1,surname2,age,genre FROM users WHERE user=%s"
        values2= (user, )
        cursor.execute(query2,values2)
        userData = cursor.fetchall()
        bd.close()
        if userData == []:
            return False #"El usuario no se ha podido registrar"
        else:
            return userData[0]
    else:
        return False #"este nombre de usuario ya existe"
```

He creado una funció més 'UsuarioNoexiste' que comproba si el usuari està intentant inserir un nom d'usuari ja existent en la BD.

```
def UsuarioNoExiste(user):
    bd=connectBD()
    cursor=bd.cursor()
    query=f"SELECT user FROM users WHERE user=%s;"
    values= (user, )
    print(query)
    cursor.execute(query,values)
    userData = cursor.fetchall()
    bd.close()
    if userData == []:
        return True
    else:
        return False
```

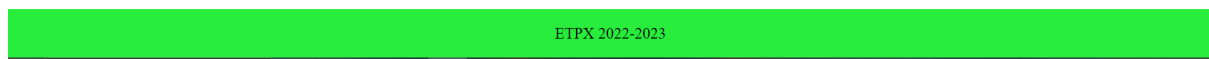

e) Comprova el correcte funcionament de l'aplicació.

Está vivo



USUARIO CREADO CORRECTAMENTE

User	Name	Surname 1	Surname 2	Age	Genre
hewwoo	gvhjb	fcg	kjhbqv	16	D



También lo podemos ver en la BD

	user	password	name	surname1	surname2	age	genre
▶	0	NULL	NULL	NULL	NULL	NULL	NULL
	caaaaa	prueba	prueba	apellido	apellido2	20	D
	hewwoo	ghbj	gvhjb	fcg	kjhbqv	16	D
	lmk	dxfcbh	dascf	htre	gfd	12	D
	PRUEBA	prueba	prueba	apellido	apellido2	20	D
	use74	hola	cris	jdi	notien	12	D
	user01	admin	Ramón	Sigüenza	López	35	H
	user02	admin	Ramón	Sigüenza	López	35	H
	user4	hola	cris	jdi	notien	12	D
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL

f) Puja el codi complet de l'aplicació a un repositori del teu compte de GitHub i inclou l'enllaç en el PDF que entreguis.