

# Ciberseguridad - Práctica 3

## **Atacando una aplicación Web**

---

Alejandro Monterrubio Navarro

Esther Pérez Gil

Grupo 2462 - Pareja 2

## Índice

Introducción y objetivos	2
Posibles vulnerabilidades	3
Programación de la aplicación	3
Página de registro	3
Página de inicio de sesión	5
Ajustes del usuario	7
Página de los proyectos	9
Página de las tareas de proyectos	11
Vulnerabilidades a explotar y procedimiento	13
Django	13
Variable DEBUG (vulnerabilidad explotada)	13
Seguridad de credenciales	14
En inicio de sesión (vulnerabilidad explotada)	14
En el registro (vulnerabilidad explotada)	14
Añadir scripts maliciosos	15
Cambio de foto de perfil (vulnerabilidad explotada)	15
Añadir un archivo a un proyecto (vulnerabilidad explotada)	17
<i>Cross-site scripting</i>	17
En los campos de un proyecto (vulnerabilidad explotada)	17
Registro e inicio de sesión (vulnerabilidades no existentes)	18
Inyecciones SQL	19
Registro e inicio de sesión (vulnerabilidades no existentes)	19
Subida de archivo (vulnerabilidad explotada)	20
Accesos a páginas no autorizadas	21
Acceso a perfiles de otros usuarios (vulnerabilidad explotada)	21
Auto-asignación de permisos de superusuario (vulnerabilidad explotada)	24
Conclusiones	27
Conclusiones técnicas	27
Conclusiones personales	27



## Introducción y objetivos

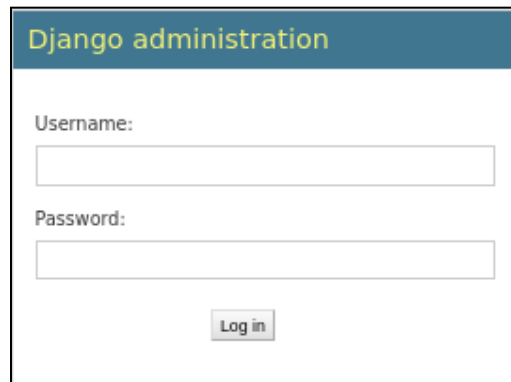
En esta práctica se va a desempeñar el trabajo de un hacker de *Red Hat*. Para ello, la compañía “Tannen Corp.” nos proporciona una máquina virtual con una versión de la aplicación de gestión de proyectos para sus hoteles y casinos en *Hill Valley* que desean publicar. Nuestro trabajo será explorar la aplicación en busca de posibles vulnerabilidades que puedan ser explotadas. Más adelante, se intentará explotar dichas vulnerabilidades, proponiendo una posible solución si el ataque tuviese éxito, o exponiendo por qué la aplicación ha sido resistente en caso de no tener éxito.

## Posibles vulnerabilidades

En este apartado se van a exponer todas las posibles vulnerabilidades a explotar que hemos encontrado tras analizar la aplicación.

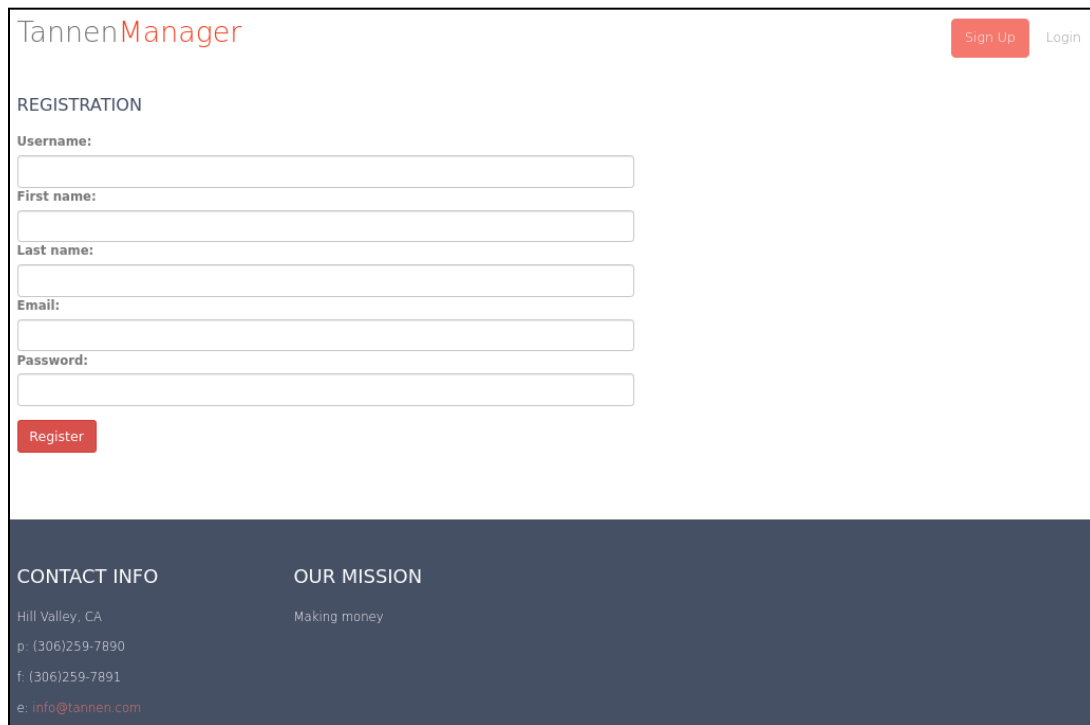
### Programación de la aplicación

Sería útil saber cómo está programada la aplicación para poder buscar vulnerabilidades específicas. Para hacerlo hemos entrado en la página de admin en busca de pistas. Nos ha salido la página de administrador correspondiente a Django, de modo que ahora podemos buscar vulnerabilidades específicas de este framework.

A screenshot of the Django administration login page. The page has a dark blue header with the text "Django administration" in yellow. Below the header, there are two input fields: "Username:" and "Password:". Below the password field is a "Log in" button.

### Página de registro

Desde la página de inicio, clicando en "Sign up" se accede a la página de registro.



TannenManager

Sign Up Login

REGISTRATION

Username:

First name:

Last name:

Email:

Password:

Register

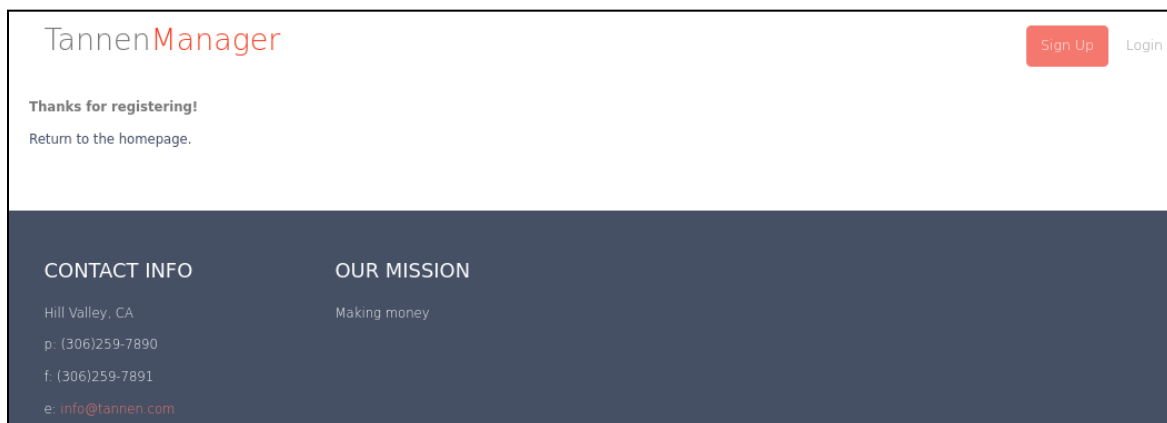
CONTACT INFO

Hill Valley, CA  
p: (306)259-7890  
f: (306)259-7891  
e: [info@tannen.com](mailto:info@tannen.com)

OUR MISSION

Making money

Tras introducir unos datos, se redirige al usuario a otra página indicando que el registro se ha realizado con éxito.



TannenManager

Sign Up Login

Thanks for registering!

Return to the homepage.

CONTACT INFO

Hill Valley, CA  
p: (306)259-7890  
f: (306)259-7891  
e: [info@tannen.com](mailto:info@tannen.com)

OUR MISSION

Making money

Si se vuelven a introducir en la página de registro los datos introducidos anteriormente (es decir, de un usuario que ya existe), la página se recarga.

Respecto al registro de usuarios hemos encontrado las siguientes vulnerabilidades:

1. En ningún momento se confirma que el mail introducido pertenezca a la persona que crea la cuenta (mandando un código de confirmación, por ejemplo).

2. No se comprueba que la contraseña introducida cumpla con unos estándares mínimos de seguridad (número de caracteres, uso de mayúsculas, minúsculas, números o caracteres especiales, etc.).
3. Por la respuesta de la página se puede deducir qué usuarios existen, permitiendo el acceder a su cuenta utilizando un ataque de fuerza bruta con la contraseña. Además, al no haber captchas se facilita esta tarea.
4. Podría hacerse un ataque por inyección de SQL, *cross-site scripting*, o inyección HTML con alguno de los campos del registro.
5. Puede que la contraseña no se esté almacenando en el sistema con un cifrado seguro.
6. Podrían introducirse unas credenciales extremadamente largas para provocar un desbordamiento de buffer.

## Página de inicio de sesión

Clicando desde la página de inicio en “Login” se accede a la aplicación de inicio de sesión.

TannenManager

Sign Up Login

LOGIN TO TANNEN MANAGER

Username

Password

Submit

Forgot your password?

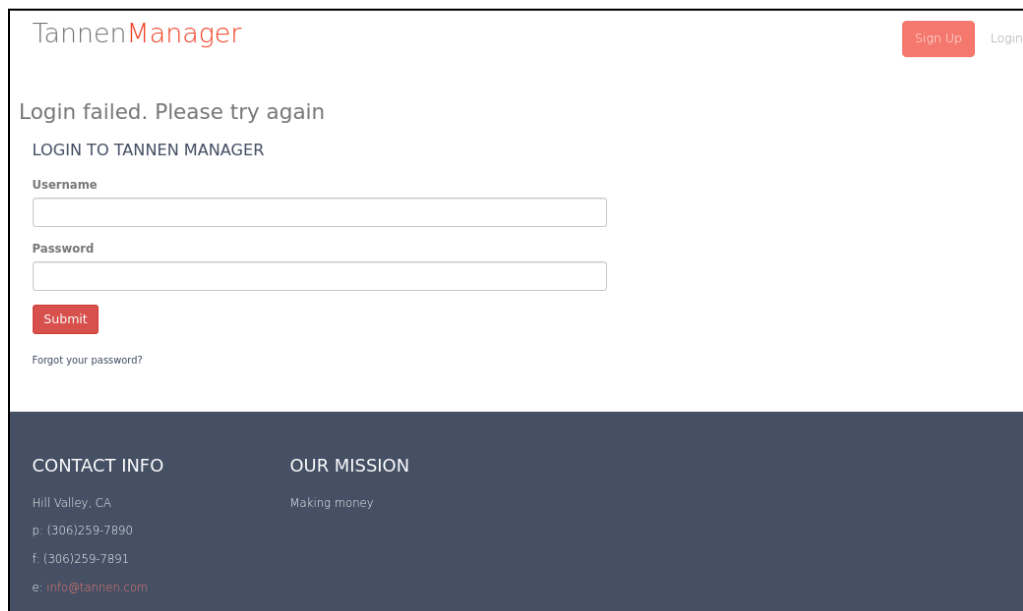
CONTACT INFO

Hill Valley, CA  
p: (306)259-7890  
f: (306)259-7891  
e: info@tannen.com

OUR MISSION

Making money

Si se introduce de manera incorrecta la contraseña de un usuario existente, la página muestra el siguiente mensaje.



TannenManager [Sign Up](#) [Login](#)

Login failed. Please try again

LOGIN TO TANNEN MANAGER

Username

Password

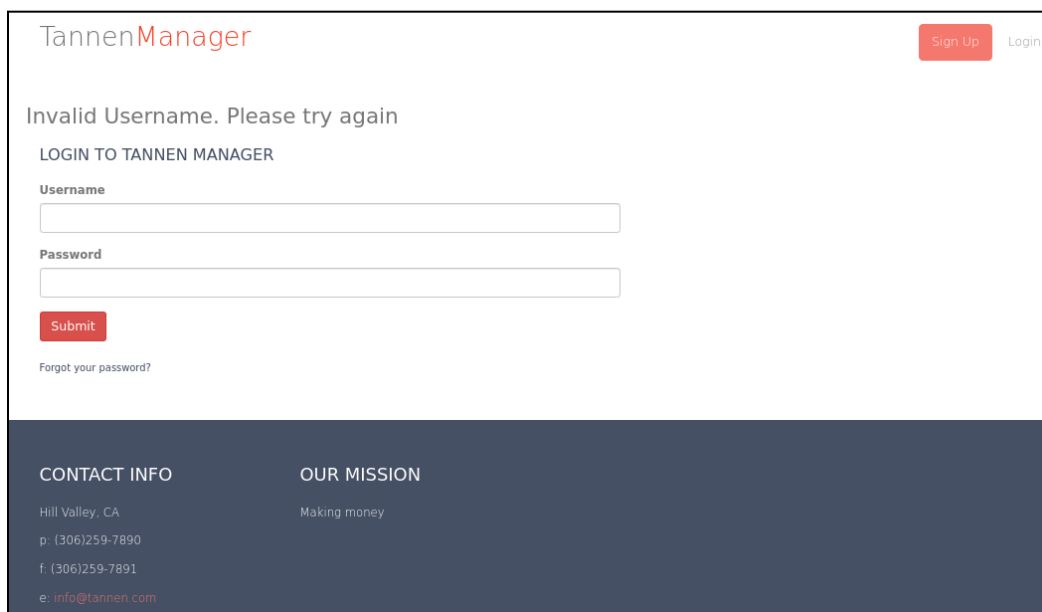
[Submit](#)

[Forgot your password?](#)

**CONTACT INFO**  
Hill Valley, CA  
p: (306)259-7890  
f: (306)259-7891  
e: [info@tannen.com](mailto:info@tannen.com)

**OUR MISSION**  
Making money

Mientras que si se escriben credenciales de un usuario que no existe, se muestra el siguiente mensaje.



TannenManager [Sign Up](#) [Login](#)

Invalid Username. Please try again

LOGIN TO TANNEN MANAGER

Username

Password

[Submit](#)

[Forgot your password?](#)

**CONTACT INFO**  
Hill Valley, CA  
p: (306)259-7890  
f: (306)259-7891  
e: [info@tannen.com](mailto:info@tannen.com)

**OUR MISSION**  
Making money

Respecto al inicio de sesión hemos encontrado las siguientes vulnerabilidades:

1. Para iniciar sesión únicamente se solicita una contraseña, no se utiliza un factor de doble autenticación.

2. La respuesta de la página nuevamente nos permite saber qué usuarios existen en el sistema, facilitando el acceder a su cuenta utilizando un ataque de fuerza bruta con la contraseña. Tampoco encontramos un captcha para iniciar sesión.
3. En relación al punto anterior, no se produce ningún bloqueo de la cuenta tras introducir incorrectamente la contraseña de un usuario numerosas veces.
4. Podría hacerse un ataque por inyección de SQL, *cross-site scripting*, o inyección HTML con alguno de los campos de inicio de sesión.
5. Pueden introducirse unas credenciales extremadamente largas para provocar un desbordamiento de buffer.

## Ajustes del usuario

En la parte superior derecha se permite al usuario acceder a su perfil o cambiar de contraseña. Las opciones del perfil incluyen cambiar el usuario, apellidos, email y foto de perfil.

Edit Profile (biff)

Username

biff

First Name

Biff


Last Name

Tannen

Email

biff@tannen.com

Icon



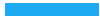
Browse...

No file selected.

Save

Por otro lado, para cambiar la contraseña se solicita la contraseña antigua, y la nueva dos veces.





Change Password

Old Password

New Password

New Password (Confirm)

Save

Si se introduce la contraseña antigua y una contraseña nueva, se muestra un mensaje indicando que se ha actualizado la contraseña. Este mensaje aparece aunque la nueva contraseña sea igual que la antigua.

Password Updated

Change Password

Old Password

New Password

New Password (Confirm)

Save

Si se introduce incorrectamente la contraseña actual, o las nuevas contraseñas no coinciden entre ellas, se muestra un mensaje indicando que la contraseña no es válida.

Invalid Password

Change Password

Old Password

New Password

New Password (Confirm)

Save

Respecto a los ajustes del usuario hemos encontrado las siguientes vulnerabilidades:

1. Una persona no autorizada podría acceder a la cuenta de un usuario y cambiar las credenciales, impidiendo el acceso al usuario real causando una pérdida de control de acceso<sup>1</sup>.
2. No se notifica al usuario ni se solicita otro tipo de confirmación para cambiar los datos del perfil ni la contraseña, de modo que si alguien accediese a la cuenta de un usuario y modificase los campos, el verdadero usuario no sabría que han accedido a su cuenta.
3. Puede que no se confirme que el archivo seleccionado para cambiar la foto de perfil verdaderamente sea una imagen. Esto permitiría que se suban scripts maliciosos que son almacenados en la base de datos de la empresa.
4. Puede que la contraseña no se esté almacenando en el sistema con un cifrado seguro.
5. Pueden introducirse unas credenciales extremadamente largas para provocar un desbordamiento de buffer.

## Página de los proyectos

Clicando en un proyecto se accede a los detalles del mismo.

The screenshot displays the 'TANNENMANAGER' application interface. On the left is a dark sidebar with navigation options: Dashboard, My Projects, My Tasks, and Search. The main content area is titled 'Default' and includes an 'Edit Project' button and a 'Refresh' button. The project details section shows the status as 'Active', creation date as 'Oct. 10, 2014, 8:56 p.m.', and due date as 'April 12, 2024, 6:53 p.m.'. Below this is a 'Participants' section with a profile picture. The 'Tasks' section features a table with columns for Title, Due Date, Status, and Actions. The table lists four tasks: 'News', 'Django', 'Hygiene', and 'Dress yourself', all with due dates of 'April 12, 2024, 6:53 p.m.' and status of 'In Progress'. Each task has 'Edit' and 'Delete' buttons. On the right, the 'Project Description' section contains the text 'This is the first project' and a '0% Complete' progress bar. The 'Priority' section shows 'Low Priority' selected. The 'Files' section has an 'Add files' button.

Title	Due Date	Status	Actions
News	April 12, 2024, 6:53 p.m.	In Progress	Edit Delete
Django	April 12, 2024, 6:53 p.m.	In Progress	Edit Delete
Hygiene	April 12, 2024, 6:53 p.m.	In Progress	Edit Delete
Dress yourself	April 12, 2024, 6:53 p.m.	In Progress	Edit Delete

Clicando en “Edit Project” se pueden editar algunos datos sobre el proyecto.

<sup>1</sup> <https://www.incibe.es/empresas/blog/top-10-vulnerabilidades-web-2021>

Respecto a esta funcionalidad hemos encontrado las siguientes vulnerabilidades:

1. Podría hacerse un ataque por inyección de SQL, *cross-site scripting*, o inyección HTML con alguno de los campos de texto.
2. Podrían introducirse datos extremadamente largos para provocar un desbordamiento de buffer.

Clicando en el apartado “Add files” se puede añadir un archivo al proyecto.

Respecto a esta funcionalidad hemos encontrado las siguientes vulnerabilidades:

1. Se podría introducir un archivo con código malicioso que se almacene en la base de datos de la empresa.
2. Podría hacerse un ataque por inyección de SQL, *cross-site scripting*, o inyección HTML con el nombre del fichero.

3. Podría introducirse un nombre de archivo extremadamente largo para provocar un desbordamiento de buffer.

Clicando en “Add task” se puede añadir una tarea al proyecto.

The screenshot shows the 'TANNENMANAGER' interface. On the left is a dark sidebar with navigation links: 'Dashboard', 'My Projects', 'My Tasks', and 'Search'. The main content area is titled 'Create Task'. It contains three input fields: 'Task Title', 'Task Information', and 'Task Due Date'. The 'Task Due Date' field is expanded, showing a calendar for May 2024 with the 3rd highlighted, and a time picker with options from 12:00 to 17:00. A blue 'Finish' button is located at the bottom of the form.

Respecto a esta funcionalidad hemos encontrado las siguientes vulnerabilidades:

1. Podría hacerse un ataque por inyección de SQL, *cross-site scripting*, o inyección HTML con los campos de texto.
2. Podrían introducirse datos extremadamente largos para provocar un desbordamiento de buffer.

## Página de las tareas de proyectos

Al clicar en una tarea se puede acceder a su información, donde además se permite editar sus campos, añadir una nota o marcarla como completada.

The screenshot shows the 'Edit Task' form in the TANNENMANAGER application. The left sidebar contains navigation links: Dashboard, My Projects, My Tasks (checked), and Search. The main content area has a header 'Edit Task' with a search icon and a user profile 'biff'. Below this is a form with two sections: 'Enter Task Title' with a text input containing 'News', and 'Enter Task Information' with a text area containing 'What's news?'. At the bottom of the form, there is a 'Task Completion' section with radio buttons for 'Yes' and 'No' (selected), and a blue 'Finish' button. A green 'Add Note' button is located in the top right corner of the form area.

The screenshot shows the 'Note Creation' form in the TANNENMANAGER application. The left sidebar is identical to the previous screenshot, with 'My Tasks' checked. The main content area has a header 'Note Creation'. Below this is a form with two sections: 'Enter Note Title' with a text input, and 'Enter Note Description' with a text area. At the bottom of the form is a blue 'Finish' button.

Respecto a esta funcionalidad hemos encontrado las siguientes vulnerabilidades:

1. Podría hacerse un ataque por inyección de SQL, *cross-site scripting*, o inyección HTML con los campos de texto.
2. Podrían introducirse datos extremadamente largos para provocar un desbordamiento de buffer.

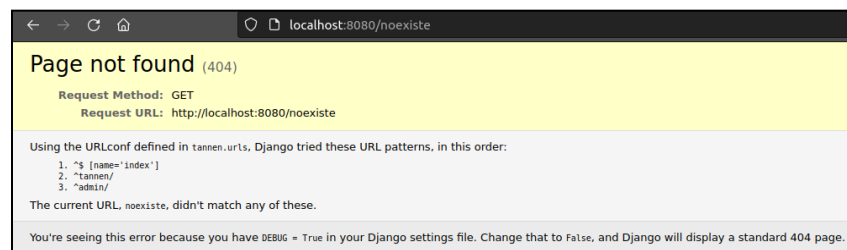
## Vulnerabilidades a explotar y procedimiento

Para este apartado se han seleccionado algunas de las posibles vulnerabilidades expuestas anteriormente para intentar explotarlas.

### Django

#### Variable DEBUG (vulnerabilidad explotada)

Al desarrollar aplicaciones en Django existe la variable “DEBUG”, que es útil tenerla a *True* mientras se programa, pero no cuando la web ya se ha distribuido, ya que aporta información de la programación interna. Para ver el valor de esta variable accedemos a una ruta que no existe, donde nos muestra posibles urls donde entrar o el código de la aplicación.



```
download_profile_pic' with arguments '(None,)' and keyword arguments '{} ' not four
22         <div class="form-group col-lg-7 col-sm-7">
23             <label>Last Name</label>
24             <input name="last_name" class="form-control" value="{{user.last_name}}">
25         </div>
26         <div class="form-group col-lg-7 col-sm-7">
27             <label>Email</label>
28             <input name="email" class="form-control" value="{{user.email}}">
29         </div>
30         <div class="form-group col-lg-7 col-sm-7">
31             <label>Icon</label>
32             <p></p>
33             <input class="btn btn-sm btn-info" id="picture" name="picture" type="file">
34             <hr />
35         </div>
36         <div class="form-group col-lg-7 col-sm-7">
37             <button type="submit" class="btn btn-info">Save</button>
38         </div>
39     </form>
40 </div>
41 </section>
42 </div>
```

La manera de solucionar este error sería simplemente cambiar a *False* la variable en el código.

## Seguridad de credenciales

### En inicio de sesión (vulnerabilidad explotada)

Una posible vulnerabilidad es que no se estén codificando correctamente las credenciales de los usuarios. Para comprobarlo, vamos a utilizar “Burp Suite” y a analizar la petición que se hace al servidor al iniciar sesión.



En la última línea se puede ver cómo el usuario y la contraseña aparecen como texto plano, confirmando que la contraseña no se cifra antes de llegar al servidor<sup>2</sup>, lo que facilita un ataque de *man-in-the-middle*. Para solucionar este problema habría que implementar un sistema que encripte la contraseña al mandar la petición.

### En el registro (vulnerabilidad explotada)

Siguiendo el mismo procedimiento que en el apartado anterior, vamos a comprobar si las credenciales introducidas al registrarse van cifradas o no.

<sup>2</sup> <https://forum.portswigger.net/thread/password-seen-in-clear-text-on-burp-tool-d3e121c9>

```
Request to http://localhost:8080 [127.0.0.1]
Forward Drop Intercept is on Action Open browser Comment this item HTTP/1
Pretty Raw Hex
1 POST /tannen/register/ HTTP/1.1
2 Host: localhost:8080
3 Content-Length: 700
4 Cache-Control: max-age=0
5 sec-ch-ua: "Chromium";v="111", "Not(A:Brand";v="8"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Windows"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://localhost:8080
10 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryBAq7ynIPUiRJhuMn
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5563.111 Safari/537.36
12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://localhost:8080/tannen/register/
18 Accept-Encoding: gzip, deflate
19 Accept-Language: es-ES,es;q=0.9
20 Cookie: csrftoken=Huec1Duk2hvs9P76O4rIWGFIrZ92u79o
21 Connection: close
22
23 -----WebKitFormBoundaryBAq7ynIPUiRJhuMn
24 Content-Disposition: form-data; name="csrftoken"
25
26 Huec1Duk2hvs9P76O4rIWGFIrZ92u79o
27 -----WebKitFormBoundaryBAq7ynIPUiRJhuMn
28 Content-Disposition: form-data; name="username"
29
30 prueba
31 -----WebKitFormBoundaryBAq7ynIPUiRJhuMn
32 Content-Disposition: form-data; name="first_name"
33
34 nombre
35 -----WebKitFormBoundaryBAq7ynIPUiRJhuMn
36 Content-Disposition: form-data; name="last_name"
37
38 apellido
39 -----WebKitFormBoundaryBAq7ynIPUiRJhuMn
40 Content-Disposition: form-data; name="email"
41
42 mail@mail.com
43 -----WebKitFormBoundaryBAq7ynIPUiRJhuMn
44 Content-Disposition: form-data; name="password"
45
46 contrasena
47 -----WebKitFormBoundaryBAq7ynIPUiRJhuMn--
```

Analizando la petición, podemos ver en rojo los campos introducidos: “prueba”, “nombre”, “apellido”, “[mail@mail.com](mailto:mail@mail.com)” y “contrasena”. De modo que, tampoco se cifra la contraseña al registrarse. Para solucionar este problema habría que implementar un sistema que encripte la contraseña al mandar la petición.

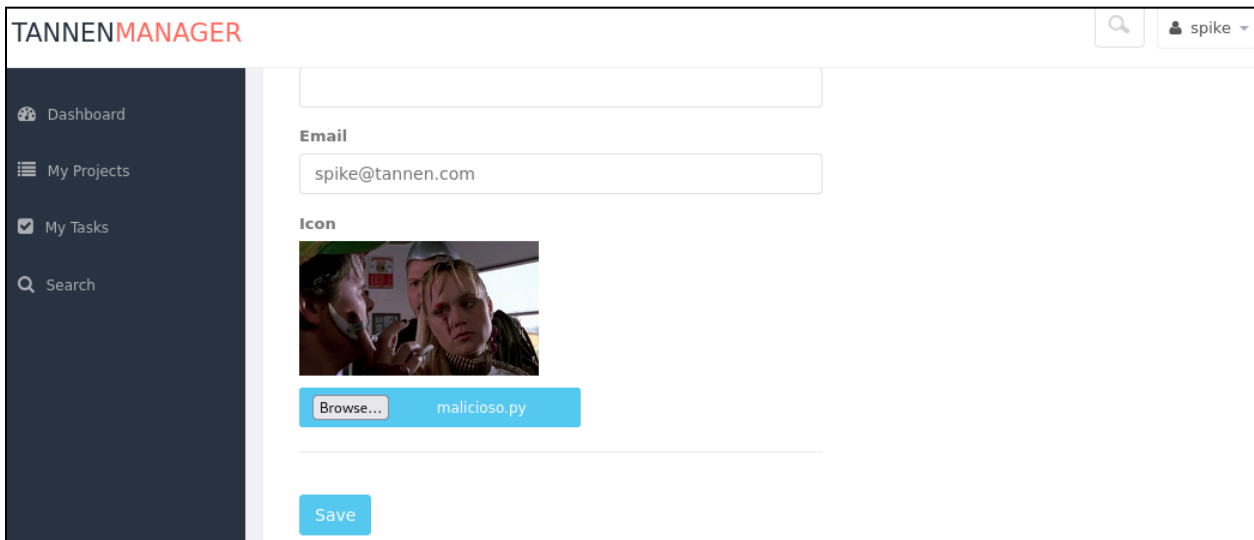
## Añadir scripts maliciosos

### Cambio de foto de perfil (vulnerabilidad explotada)

La aplicación permite cambiar la imagen de perfil de los usuarios. Para comprobar si existe una vulnerabilidad vamos a crear un script “malicioso” y vamos a intentar seleccionarlo como nueva foto de perfil. Si la aplicación acepta este archivo existiría una

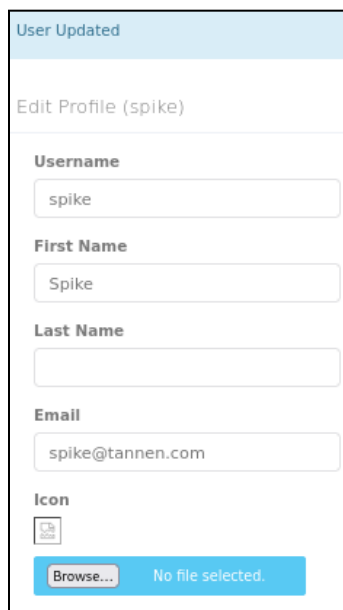


vulnerabilidad, ya que sería posible introducir malware en la base de datos de la empresa.



The screenshot shows the 'TANNENMANAGER' application interface. On the left is a dark sidebar with navigation links: 'Dashboard', 'My Projects', 'My Tasks', and 'Search'. The main content area is titled 'Edit Profile (spike)' and contains the following fields: a top text input field, an 'Email' field with the value 'spike@tannen.com', an 'Icon' field displaying a movie still of a man and a woman, and a 'Browse...' button next to the text 'malicioso.py'. At the bottom of the form is a blue 'Save' button.

Al clicar en “Save” se muestra un mensaje indicando que el perfil se ha actualizado correctamente.



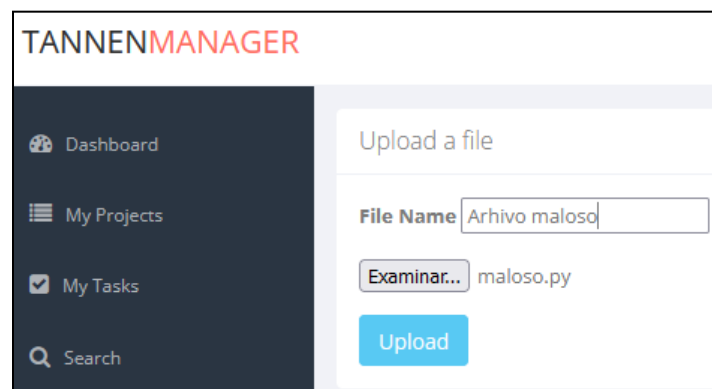
The screenshot shows a confirmation message box titled 'User Updated'. Below the title is the text 'Edit Profile (spike)'. The form contains the following fields: 'Username' with the value 'spike', 'First Name' with the value 'Spike', 'Last Name' (empty), 'Email' with the value 'spike@tannen.com', and 'Icon' with a small image icon and a 'Browse...' button next to the text 'No file selected.'.

Hemos comprobado que existe una vulnerabilidad al no controlar los archivos que se seleccionan como imagen de perfil. La manera más sencilla de solucionar este problema, pero no del todo fiable, es modificar el código para que únicamente se puedan añadir archivos con extensión de imagen. Aun así, es posible añadir información oculta en las

imágenes, de modo que se podrían utilizar adicionalmente métodos de estegoanálisis<sup>3</sup> (como redes neuronales entrenadas) para detectar este posible malware oculto, o conectar con la API de “VirusTotal” para confirmar que es un archivo seguro. Sin embargo, habría que hacer un análisis de riesgos para ver si merecería la pena gastar tantos recursos en este apartado.

### Añadir un archivo a un proyecto (vulnerabilidad explotada)

Un usuario puede añadir un archivo a un proyecto. Podría aprovecharse esta funcionalidad para insertar un archivo con malware que se guardará en la base de datos de la aplicación.



Al guardar los cambios la página simplemente se recarga. De haberse guardado el archivo, se habría conseguido introducir con éxito el archivo de malware. Una manera de solucionar esto es conectar con la API de “VirusTotal” para detectar si se trata de un archivo fraudulento.

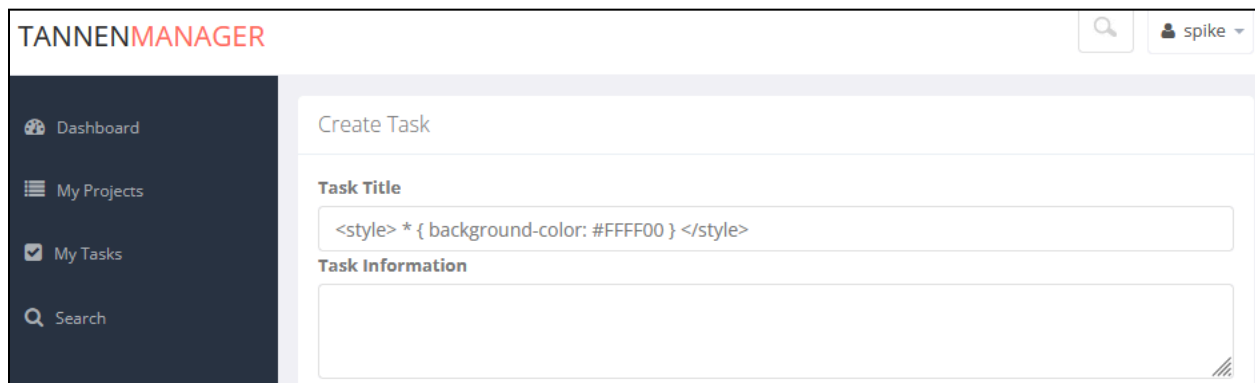
## Cross-site scripting

### En los campos de un proyecto (vulnerabilidad explotada)

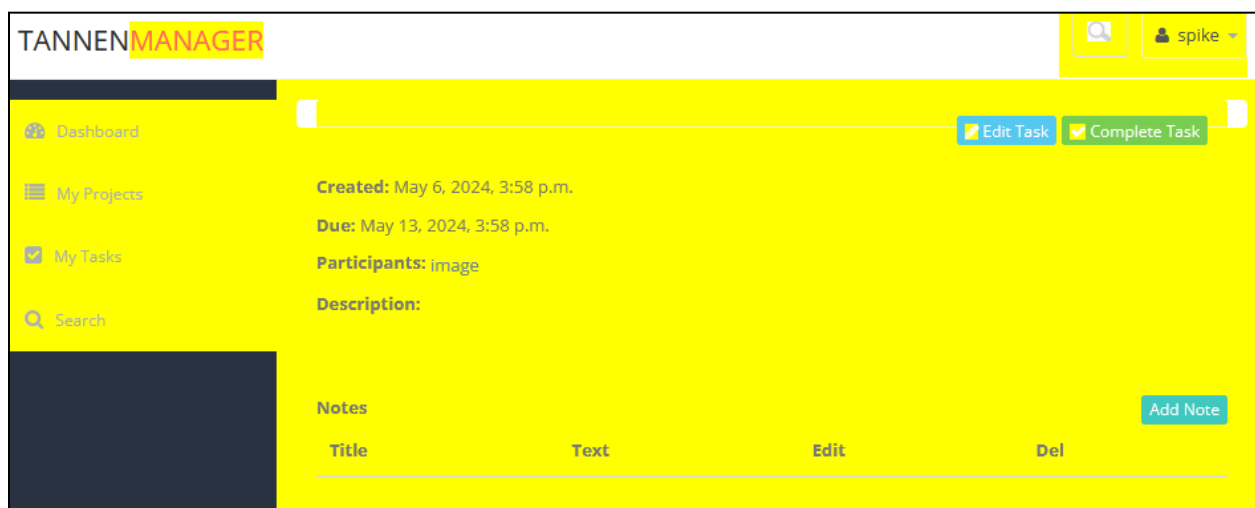
Dentro de un proyecto existen muchos campos de texto que son susceptibles a sufrir un ataque de *cross-site scripting*. Para probar si existe una vulnerabilidad, hemos insertado como texto un código muy simple que cambia el color del fondo<sup>4</sup>. Por ejemplo, se va a crear una tarea en un proyecto con un nombre que cambia el fondo al color amarillo.

<sup>3</sup> <https://www.baeldung.com/cs/malware-hidden-image-files>

<sup>4</sup> <https://learn.snyk.io/lesson/xss/>



Al clicar en la tarea creada se puede ver que la pantalla aparece de color amarillo, confirmando que ha sido posible hacer *cross-site scripting*.

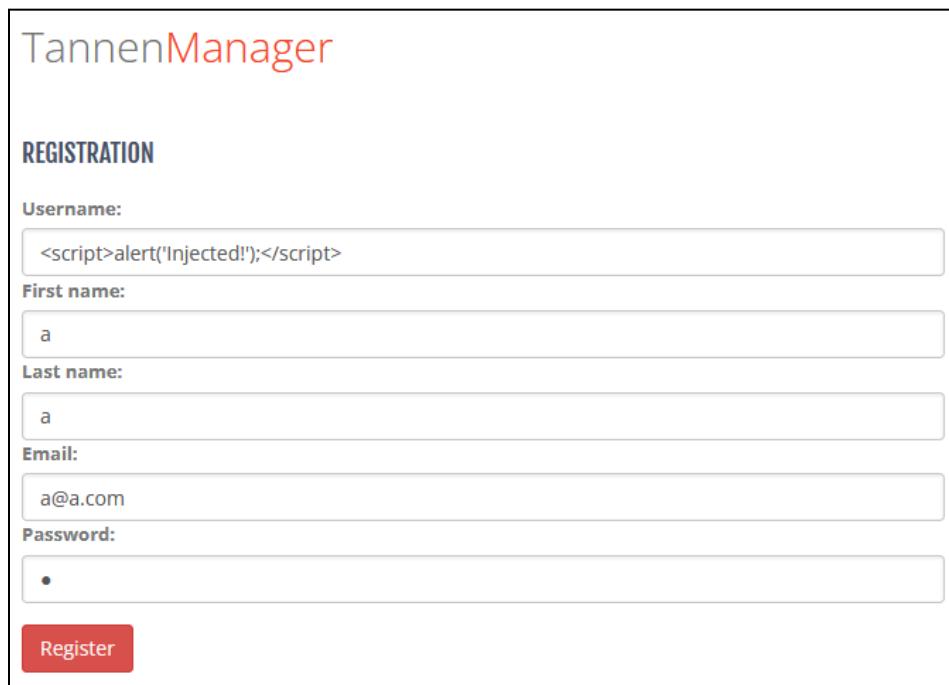


Hemos probado a hacer este tipo de *cross-site scripting* con todos los campos de texto relacionados a proyectos (descripción de un proyecto, añadir ficheros, etc.) y el único lugar donde ha sido posible era con el nombre de las tareas. Una manera de evitar este tipo de ataque podría ser comprobando los caracteres introducidos y asegurando que sólo sean letras o números, por ejemplo, o utilizando alguna biblioteca que reconozca si se trata de una línea de código.

### Registro e inicio de sesión (vulnerabilidades no existentes)

Para probar si es posible realizar un ataque por inyección de HTML<sup>5</sup> en los campos de texto del registro, vamos a introducir “<script>alert('Injected!');</script>” como nombre de usuario. De funcionar, debería aparecer una alerta en la parte superior con la palabra.

<sup>5</sup> <https://www.codeproject.com/Articles/134024/HTML-and-JavaScript-Injection>



The screenshot shows a web application titled "TannenManager" with a "REGISTRATION" section. The form contains five input fields: "Username:", "First name:", "Last name:", "Email:", and "Password:". The "Username:" field contains the payload "<script>alert('Injected!');</script>". The "First name:" field contains "a", the "Last name:" field contains "a", the "Email:" field contains "a@a.com", and the "Password:" field contains a single dot. A red "Register" button is located at the bottom left of the form.

Al registrar el usuario la página se recarga y no aparece el mensaje de que se haya registrado correctamente el usuario. Si intentamos iniciar sesión con las credenciales introducidas nos indica que no existe ese usuario. Podemos confirmar que no existe una vulnerabilidad: la aplicación comprueba que el texto introducido no es una línea de código y no crea un usuario con ese nombre.

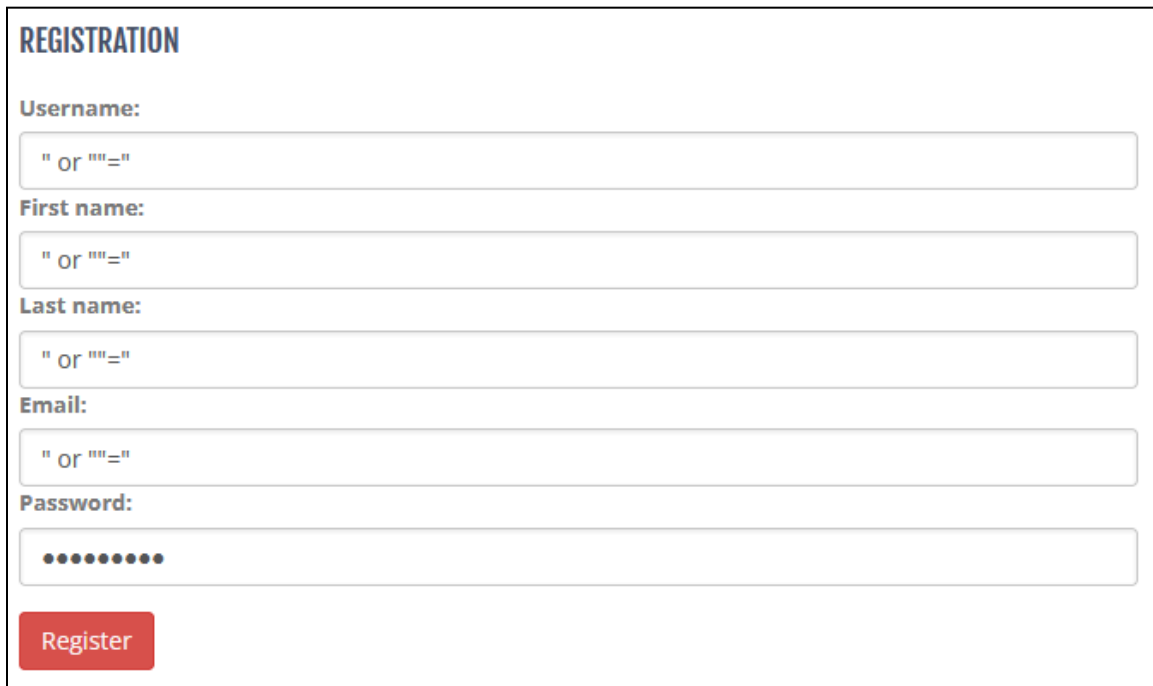
## Inyecciones SQL<sup>6</sup>

### Registro e inicio de sesión (vulnerabilidades no existentes)

No hemos tenido éxito al hacer una inyección de código HTML, así que vamos a probar con SQL. Para ello, hemos introducido "" or ""="" como todos los campos.

---

<sup>6</sup> [https://www.w3schools.com/sql/sql\\_injection.asp](https://www.w3schools.com/sql/sql_injection.asp)

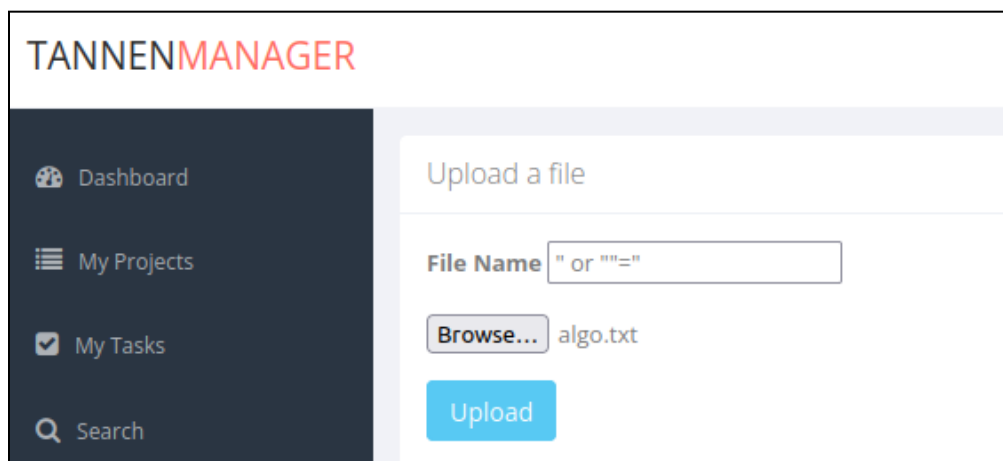


A registration form titled "REGISTRATION" with the following fields: Username, First name, Last name, Email, and Password. Each of the first four fields contains the SQL injection payload: " or ""="". The Password field is masked with dots. A red "Register" button is at the bottom left.

Al registrar el usuario la página se recarga y no aparece el mensaje de que se haya registrado correctamente el usuario. Si intentamos iniciar sesión con las credenciales introducidas nos indica que no existe ese usuario. Podemos confirmar que no existe una vulnerabilidad: la aplicación comprueba que el texto introducido no es una línea de código y no crea un usuario con ese nombre.

#### Subida de archivo (vulnerabilidad explotada)

Para subir un fichero a un proyecto se pide un nombre de fichero y el fichero en sí. Vamos a probar a introducir la misma sentencia SQL anterior en el campo del nombre.



The TANNENMANAGER interface shows a sidebar with "Dashboard", "My Projects", "My Tasks", and "Search". The main area is titled "Upload a file" and contains a "File Name" field with the payload " or ""="". Below the field is a "Browse..." button and the filename "algo.txt". An "Upload" button is at the bottom.

En este caso obtenemos un mensaje de Django indicando que no existe la tabla “taskManager\_file”. Esto significa que, como la base de datos estará en el servidor de la empresa, y no está en nuestro ordenador (ya que esto es una simulación de la página, no una de verdad), no se detecta ninguna tabla con este nombre. Sin embargo, el ataque ha tenido éxito, ya que lo ha detectado como una sentencia SQL. De haber tenido acceso a la tabla, podríamos haber accedido a todos sus datos, modificarlos, o añadir y eliminar filas y columnas.

```
OperationalError at /tannen/2/upload/
no such table: taskManager_file

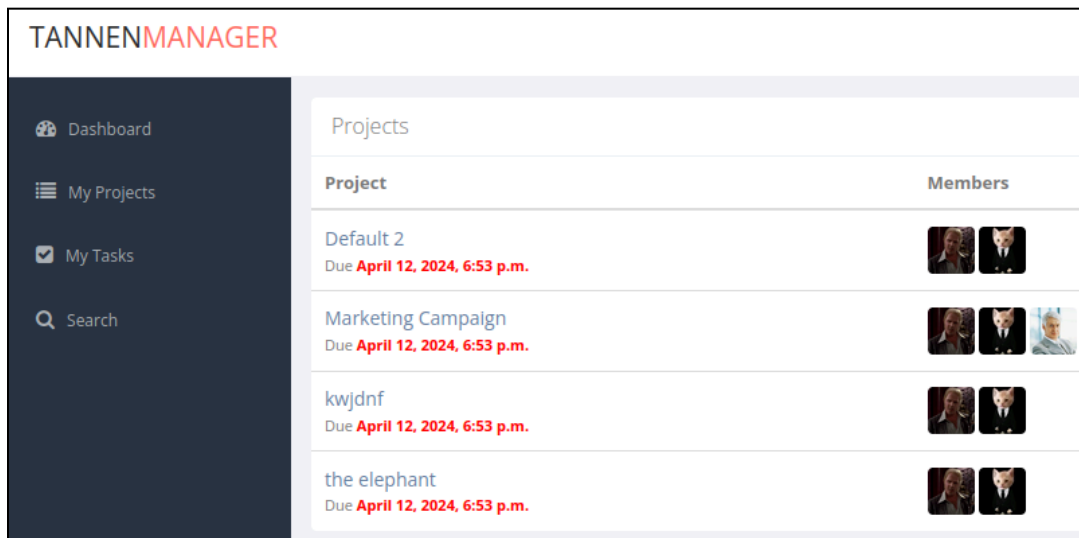
Request Method: POST
Request URL: http://localhost:8080/tannen/2/upload/
Django Version: 1.8.3
Exception Type: OperationalError
Exception Value: no such table: taskManager_file
Exception Location: /app/venv/lib/python3.4/site-packages/django/db/backends/sqlite3/base.py in execute, line 316
Python Executable: /app/venv/bin/python3
Python Version: 3.4.10
Python Path: ['/app',
              '/usr/local/lib/python34.zip',
              '/usr/local/lib/python3.4',
              '/usr/local/lib/python3.4/plat-linux',
              '/usr/local/lib/python3.4/lib-dynload',
              '/app/venv/lib/python3.4/site-packages']
Server time: Wed, 8 May 2024 14:41:18 +0000
```

Una manera de evitar este ataque es utilizando alguna herramienta que no permita introducir sentencias SQL como nombre. Django además cuenta con herramientas específicas (como mark\_safe), ya que es un framework especializado.

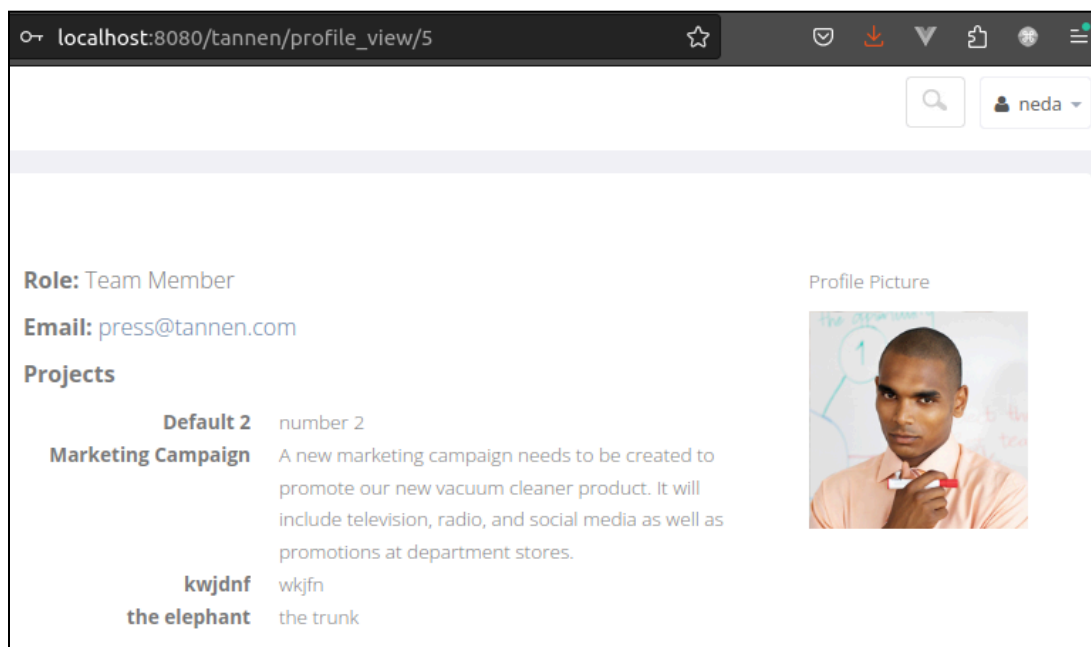
## Accesos a páginas no autorizadas

### Acceso a perfiles de otros usuarios (vulnerabilidad explotada)

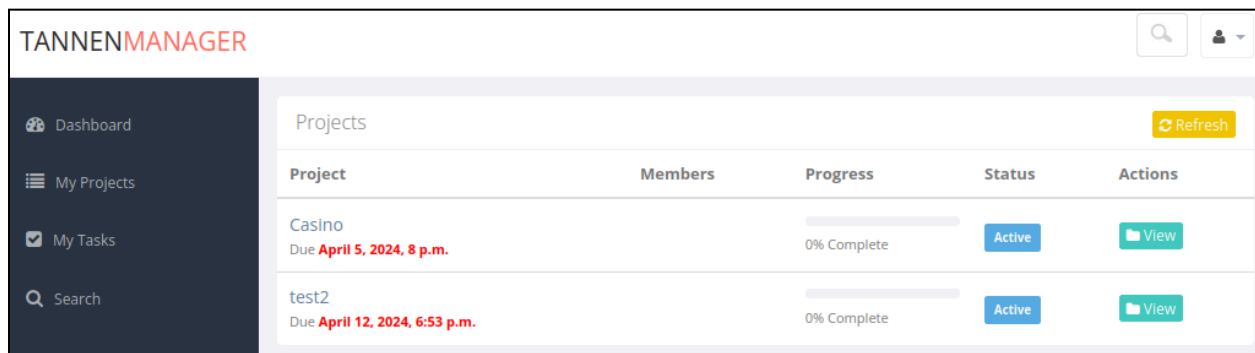
Un usuario puede consultar el perfil de otro usuario desde la lista de miembros de un proyecto.



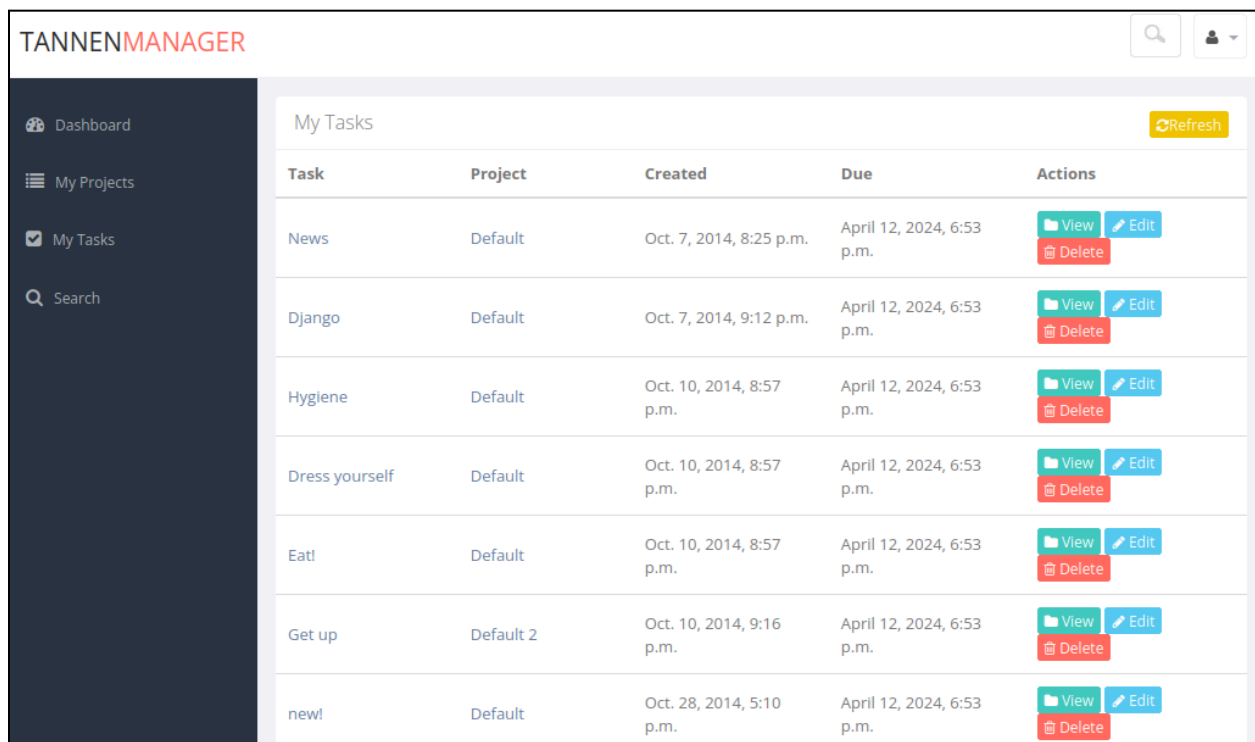
La página a la que se accede tiene la forma: [http://localhost:8080/tannen/profile\\_view/1](http://localhost:8080/tannen/profile_view/1), donde cambia el número final dependiendo del usuario. Manualmente se puede ir cambiando el número del final de la URL para acceder a los perfiles de otros usuarios. De este modo, cualquier empleado de la empresa puede tener acceso al nombre, rol y mail de cualquier empleado, cuando debería ser únicamente una funcionalidad habilitada para el administrador. Adicionalmente, al entrar en el perfil de otro usuario, aparece su nombre en la parte superior derecha como si se hubiese iniciado sesión siendo ese usuario. En el caso de algún conflicto en la empresa, se podría hacer una captura de pantalla y aparentemente sería otro empleado el responsable.



Añadido a esto encontramos una vulnerabilidad aún más grave. Al buscar el perfil de un usuario de la manera comentada anteriormente, si se cierra sesión y se clicla en la flecha hacia atrás del buscador, automáticamente entramos en el perfil del usuario que estábamos mirando. Esto nos permite acceder a todos sus proyectos y tareas, y actuar como si fuéramos esa persona.



Project	Members	Progress	Status	Actions
Casino Due April 5, 2024, 8 p.m.		0% Complete	Active	<a href="#">View</a>
test2 Due April 12, 2024, 6:53 p.m.		0% Complete	Active	<a href="#">View</a>



Task	Project	Created	Due	Actions
News	Default	Oct. 7, 2014, 8:25 p.m.	April 12, 2024, 6:53 p.m.	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
Django	Default	Oct. 7, 2014, 9:12 p.m.	April 12, 2024, 6:53 p.m.	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
Hygiene	Default	Oct. 10, 2014, 8:57 p.m.	April 12, 2024, 6:53 p.m.	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
Dress yourself	Default	Oct. 10, 2014, 8:57 p.m.	April 12, 2024, 6:53 p.m.	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
Eat!	Default	Oct. 10, 2014, 8:57 p.m.	April 12, 2024, 6:53 p.m.	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
Get up	Default 2	Oct. 10, 2014, 9:16 p.m.	April 12, 2024, 6:53 p.m.	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
new!	Default	Oct. 28, 2014, 5:10 p.m.	April 12, 2024, 6:53 p.m.	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

Sin embargo, no es posible cambiar la contraseña de la cuenta del usuario, ya que al no saber la contraseña anterior no se permite hacer el cambio. Para solucionar esta vulnerabilidad, únicamente habría que escribir algunas líneas de código que comprueben que el usuario que accede a una página tiene permisos para acceder. Esto



se puede encontrar fácilmente buscando documentación sobre desarrollo web en Django.

### Auto-asignación de permisos de superusuario (vulnerabilidad explotada)

Inspeccionando la página de registro desde *Google Chrome* (en *Firefox* no se puede editar el código) hemos encontrado una parte del código comentada. Esta parte hace referencia a los permisos de superusuario y permisos de staff. Esto nos permitiría tener permisos de administrador no sólo en la aplicación, sino en Django también.

```
<!DOCTYPE html>
<html>
  <head> ... </head>
  <body>
    <!-- Navbar
    ===== ...>
    <header class="header-frontend"> ... </header>
    <!--header end-->
    <!-- Body/content
    ===== ...>
    <!--container start-->
    <div class="container">
      ::before
      <div class="row">
        ::before
        <div class="col-lg-7 col-sm-7 address">
          <h4>Registration</h4>
          <div class="contact-form">
            <form method="post" role="form" action="/tannen/register/" enctype="multipart/form-data">
              <input type="hidden" name="csrfmiddlewaretoken" value="cypReCM023ryJsv6ZVBAWx22bLZhtvaZ">
              ...
              <div class="form-group"> == $0
                <!-- <tr><th><label for="id_password">Password:</label></th><td><input id="id_password"
                maxlength="128" name="password" type="text" /></td></tr>
                <tr><th><label for="id_is_superuser">Superuser status:</label></th><td><input id="id_is_superuser"
                name="is_superuser" type="checkbox" /><br /><span class="helptext">Designates that this user has
                all permissions without explicitly assigning them.</span></td></tr>
                <tr><th><label for="id_username">Username:</label></th><td><input id="id_username" maxlength="30"
                name="username" type="text" /><br /><span class="helptext">Required. 30 characters or fewer.
                Letters, digits and @/./+/-/_ only.</span></td></tr>
                <tr><th><label for="id_first_name">First name:</label></th><td><input id="id_first_name"
                maxlength="30" name="first_name" type="text" /></td></tr>
                <tr><th><label for="id_last_name">Last name:</label></th><td><input id="id_last_name"
                maxlength="30" name="last_name" type="text" /></td></tr>
                <tr><th><label for="id_email">Email address:</label></th><td><input id="id_email" maxlength="254"
                name="email" type="email" /></td></tr>
                <tr><th><label for="id_is_staff">Staff status:</label></th><td><input id="id_is_staff"
                name="is_staff" type="checkbox" /><br /><span class="helptext">Designates whether the user can log
                into this admin site.</span></td></tr> -->
```

Hemos descomentado el código para ver si afectaba a la página de registro, descubriendo nuevos campos que hemos rellenado con la palabra “hola” como prueba.

# TannenManager

## REGISTRATION

**Password:**  **Superuser status:** ☒

Designates that this user has all permissions without explicitly assigning them. **Username:**

Required. 30 characters or fewer. Letters, digits and @/./+/-/\_ only. **First name:**

**Last name:**  **Email address:**  **Staff status:** ☒

Designates whether the user can log into this admin site. **Username:**

**First name:**

**Last name:**

**Email:**

**Password:**

La aplicación ha registrado nuestro nuevo usuario “hola”, así que hemos intentado acceder a la página de administrador con estas credenciales.

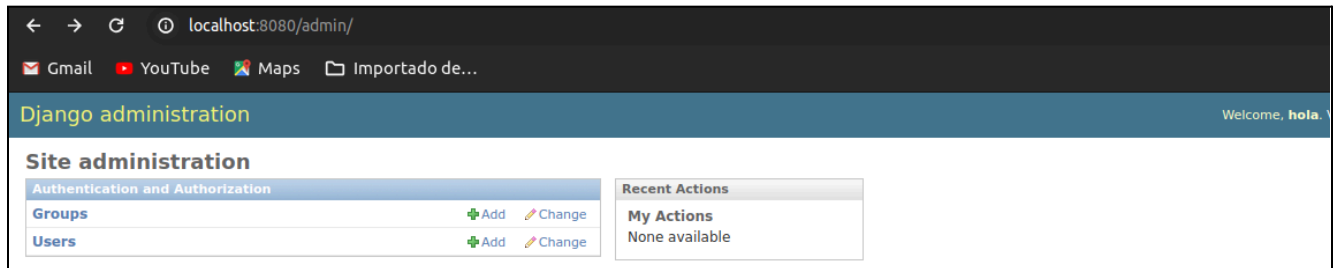
localhost:8080/admin/login/?next=/admin/  
ube Maps Importado de...

### Django administration

**Username:**

**Password:**

Y esto nos ha permitido acceder a la parte de administrador de Django, donde podemos ver todos los grupos y usuarios creados en la aplicación, al igual que nos permite editar los datos.



Esta vulnerabilidad es muy grave, pues permite a cualquier persona no sólo editar los datos de la aplicación, sino acceder a información sobre los empleados o futuros proyectos. La solución para evitar esto es no dejar comentado código que permita explotar vulnerabilidades.



## Conclusiones

### Conclusiones técnicas

Utilizando la herramienta *Burp Suite* hemos podido comprobar que las contraseñas no son cifradas antes de llegar al servidor, lo que facilita un ataque *man-in-the-middle*. Continuando con la idea de explotar vulnerabilidades de las credenciales hemos intentado inyecciones de código en el inicio de sesión y registro, ambos sin éxito. Sin embargo, hemos descubierto código comentado en relación al registro, lo que, junto con el descubrimiento de que la aplicación estaba programada en Django, nos ha permitido obtener permisos de superusuario y administrador.

Todas las funcionalidades donde podían insertarse archivos tenían vulnerabilidades, y hemos conseguido explotarlas. En primer lugar, la aplicación no comprueba el formato de la imagen de perfil nueva, permitiendo introducir scripts maliciosos en la base de datos. Por otro lado, es posible hacer inyecciones SQL al insertar un fichero dentro de un proyecto, permitiendo modificar la base de datos original en el caso de tener acceso.

Las vulnerabilidades que nos han parecido más graves, junto con la que nos permitía ser superusuarios, han sido aquellas que nos han permitido acceder a páginas no autorizadas. No sólo nos ha permitido la aplicación consultar la información de cualquier usuario de la empresa, sino que hemos sido capaces de acceder a su perfil, pudiendo modificar, eliminar o añadir proyectos o tareas.

### Conclusiones personales

Esta práctica nos ha parecido muy interesante, pero nos ha hecho darnos cuenta de que ninguna página web o práctica que hayamos realizado hasta ahora estaba a prueba de ataques. Por otro lado, teníamos preparada la máquina virtual de Kali Linux, para poder probar herramientas específicas (como “sqlmap”), pero no hemos sido capaces de entender cómo funcionaban pese a leer la documentación y ver tutoriales. Nos hubiese gustado poder hacer esta práctica de forma más “profesional” con herramientas que se utilicen en un ámbito real de hacker de *Red Hat*.

