


Investigación de un ejemplo de código inseguro y solución.

☰ Etiquetas	Tarea
👤 Autor	 NOE VAZQUEZ POMPA
🕒 Fecha de creación	@August 31, 2023 10:30 PM

Ejemplo de código inseguro en Django.

Al crear un proyecto en Django, en el archivo “**settings.py**” tenemos una sección similar a la siguiente:

```
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-v=052y%7so770j6xvj%$^it@k=$56jaicywq#g+*sij=)wc8!s'
```

Al momento de montar el proyecto de Django en un servidor para producción, es muy común en desarrolladores menos experimentados que dejan el “**SECRET_KEY**” escrito directamente en el archivo “**settings.py**”, dejar la “**SECRET_KEY**” en un acceso fácil puede provocar que los atacantes descifren información como las contraseñas, la administración de sesiones interna de Django, tokens para resetear la contraseña, etc.

Una forma de resolver este código inseguro dentro del proyecto de Django es guardar el “**SECRET_KEY**” en una variable de ambiente del S.O. y obtenerla dentro del código como se muestra en el ejemplo:

```
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist/
import os
SECRET_KEY = os.environ['SECRET_KEY'] #El nombre SECRET_KEY puede ser cambiado por otro.
```

Otra forma de resolverlo (pero no tan recomendable) es guardar el “**SECRET_KEY**” en un archivo **.env** en el cual se pueden almacenar datos.

Aquí un ejemplo de un archivo **.env** :

```
DJANGO_SECRET_KEY=%jjnu7=54g6s%qjfnhbpw0zeoei=$!her*y(p%!&84rs$4l85io
```

Y posteriormente el archivo “**settings.py**”:

```
from dotenv import load_dotenv
import os

load_dotenv()
SECRET_KEY = os.getenv("DJANGO_SECRET_KEY")
```