Raíssa Azevedo

Como melhorar a segurança da aplicação Django?

Neste projeto não se faz necessária a criação de um aplicativo, somente do projeto.

```
django-admin startproject seguranca .
```

O próprio arquivo de **settings** do projeto já possui dados de segurança, como o **secret key** que precisam ser confidenciais.

Fazer as configurações básicas do Settings.py.

O Django já possui por padrão várias configurações de segurança, como: Secret_key, Middleware security, auth validations.

RECURSOS AUTOMÁTICOS DE SEGURANÇA DO DJANGO:

- **Cross Site Scripting (XSS):**

  Que permite o atacante a injetarem scripts em sites, com HTML ou JavaScript. Mudando o comportamento das páginas Web.

- **Cross Site Request Forgery (CSRF)**

  {% csrf_token %}

  Recurso que impede a exploração de um website que transições não permitidas sejam relacionadas. Porque toda entrada de usuário pode ser perigosa e precisa ser validada. O Django provê um token nos formulários.

- **SQL Injection:**

O django vem preparado contra SQL Injection, é um dos ataques mais comuns, através de um formulário de entrada de dados, e quando clica no botão para fazer login e ocorre a combinação do dadof fornecido com o salvo no banco de dados. Sites sem essa proteção, o cracker insere comandos que vão modificar o comportamento do que está ssendo escrito, podendo até conceder permissão de adm.

- **Suporta HTTPS e TLS:**
  Quando a aplicação é startada com o *runserver* entra em HTTP (Protocolo da internet sem segurança) e não no HTTPS, ele permite que a aplicação só funcione em HTTPS.

- **Armazenamento Seguro de Senhas**
  Qualquer senha nunca deve ser armazenada em banco de dados em texto puro, ela precisa ser criptografada.

- **Algoritmo PBKDF2 com hash SHA256 recomendado pelo Instituto NIST:**
  É o algoritmo de criptografia de dados utilizados pelo Django.

Por padrão o acesso a área administrativa do django é feito através do **/admin**
Uma coisa que pode ser feita, é mudar o nome do local da administração, por Painel, controle ou outra coisa.:
Na Urls.py do projeto

```
urlpatterns = [
    path('admin/', admin.site.urls),
]
```

Trocar a rota admin, para outro nome.

Outra coisa que pode ser feita, é habilitar novos recursos de segurança, como:

```
# RECURSOS EXTRAS DE SEGURANÇA DO DJANGO
SECURE_HSTS_SECONDS = True
SECURE_HSTS_INCLUDE_SUBDOMAINS = True
SECURE_CONTENT_TYPE_NOSNIFF = True
SECURE_BROWSER_XSS_FILTER = True
SESSION_COOKIE_SECURE = True
CSRF_COOKIE_SECURE = True
CSRF_COOKI_HTTPONLY = True
X_FRAME_OPTIONS = 'DENY'

# SECURE_SSL_REDIRECT = True
```

O ideal é ativar o ultimo recurso somente no momento da aplicação, ou a aplicação não irá rodar localmente no momento do desenvolvimento.