SilverStrike

Setup Instructions for Ubuntu

Here is a guide on how to get up and running using Ubuntu.

Install dependencies

```
sudo apt update
sudo apt install nginx postgresql uwsgi uwsgi-plugin-python3 python3-venv git
```

Setup the database

The recommended database to use is postgres since it comes with sane defaults.

Postgres

In this guide we will use the www-data user because that will allow us to proceed without modifications to a standard postgres installation. If you know more you can very much do this step in a different way.

```
sudo su -c "createuser www-data && createdb -O www-data silverstrikedb" postgres
```

Get the silverstrike code

We will serve silverstrike from /srv/webapps/silverstrike and actually install it inside a virtualenv in /srv/webapps/silverstrike/env

I highly recommend using a virtualenv to install silverstrike (Python projects in general)

Depending on the database you are using, you will need to install one other peace inside the virtualenv. Replace <DATABASE_DRIVER> in the last command mentioned below with what you are using:

- Postgresql: `psycopg2`
- Mariadb: `mysqlclient`
- Sqlite: Yeah, you don't need anything

```
# get the project to deploy silverstrike
sudo mkdir /srv/webapps
chown www-data:www-data /srv/webapps
sudo su www-data -s /bin/bash
cd /srv/webapps
git clone https://github.com/agstrike/production-project silverstrike
cd silverstrike
python3 -m venv env
source env/bin/activate

pip install silverstrike <DATABASE_DRIVER>
```

Configure SilverStrike

You will need to edit the file `settings.py` doing the following:

- Uncomment the database block you are using (DATABASE) and update according to what you choose.
  In case of the recommended setup using postgres with `www-data` as a user, you can choose anything for the password.
- Change the value of SECRET_KEY
- Add your public hostname to ALLOWED_HOSTS

The respective settings should look similar to this:

```
SECRET_KEY = '$_99bc($w4iv1h8qn=q5*g-gtb&99db+0kls8m6+avmzu8w^8x'

ALLOWED_HOSTS = ['www.example.com']

DATABASES = {
    #'default': {
    #    'ENGINE': 'django.db.backends.sqlite3',
    #    'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    #}
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'silverstrikedb',
        'USER': 'www-data',
        'PASSWORD': 'password',
    }
    #'default': {
    #    'ENGINE': 'django.db.backends.mysql',
    #    'NAME': 'silverstrikedb',
    #    'USER': 'silverstrike',
    #    'PASSWORD': 'password',
    #}
}
```

Initialize everything

The first command initializes the database.
The second command collects all static files that will be served by nginx directly
The third command creates a superuseraccount. Provide a username, email and password.
You will need this information to login later

```
sudo su www-data -s /bin/bash
cd /srv/webapps/silverstrike
source env/bin/activate

python manage.py migrate
python manage.py collectstatic
python manage.py createsuperuser
```

Configure uwsgi

There is a uwsgi configuration file provided. You will need to comment out the socket option.

In case you didn't adhere to the guide, you might need to change more.

Next symlink the uwsgi configuration to where uwsgi expects it, and restart uwsgi

```
sudo ln -s /srv/webapps/silverstrike/uwsgi.ini /etc/uwsgi/apps-enabled/silverstrike.ini
sudo systemctl restart uwsgi
```

Configure the webserver

Nginx

A nginx sample config is provided in the project. The important things to change are the `server_name` entries and to make sure the paths are correct. The static files are placed inside a subdirectory of the silverstrike project called `public/static` by default. So in case you didn't clone the project to `/srv/webapps/silverstrike` you will only need to change that part.

Ubuntu uses a different standard location for the uwsi socket files, so you will need to update it. If you simlinked the uwsgi config to `/etc/uwsgi/apps-enabled/silverstrike.ini` the location of the socket would be `/run/uwsgi/app/silverstrike/socket`

You should really use ssl, I recommend using letsencrypt. There should be enough documentation on the internet on how to do that. This is the bare minimum config for nginx.

After you modified the file, symlink it and restart nginx:

```
sudo ln -s /srv/webapps/silverstrike/nginx.conf /etc/nginx/sites-enabled/silverstrike
sudo systemctl restart nginx
```

Last bits & Enjoy

This should be it, visit the name you provided in nginx and you should be greeted with the loginscreen of SilverStrike

If everything works as expected, you should modify the settings one last time, turning of DEBUG. It is insecure to use DEBUG=True in production!

```
DEBUG = False
```

You need to restart uwsgi after this:

```
sudo systemctl restart uwsgi
```