"Securing" your Database Credentials in a Script

Tom Kelliher, CS 317

Let's get this out of the way — it's not really possible to secure your database credentials (username and password) in the context of having them available for a script, Python or otherwise, to use to connect to your database. The credentials can be *obscured*, by encoding them as, say, base64 strings in your script and then decoding the strings to connect to your database. This prevents "shoulder surfing," but once anyone has a copy of your script, it's trivial for them to recover your credentials. Alternatively, your credentials can be *hidden* in a python module that only holds the credentials and is imported into your script. This "credentials" module can then be stored outside of the web server's name space, preventing the credentials from accidentally being displayed due to, say, a web server mis-configuration. The credentials file can be stored in a directory that only you and the user that the web server runs as has access to. This latter approach is the approach we take here. It's still not perfect — think about the consequences of mis-configuring directory or file permissions here.

Let's start. First, run the following commands from a shell:

```
cd   # Make sure you're in your home directory.

umask 022   # Set default permission of 'rx' for group and other.

# Create a directory to hold credentials files and set permissions
# on it so that only you and the web server can access it.

mkdir .secrets
chmod go= .secrets
setfacl -m u:apache:rx .secrets
```

Now, create a file named `psqlauth.py` within your `.secrets` directory. The contents of this file should be similar to

```
user='kelliher'         # Replace with your psql username.
pw='SixByNineFortyTwo'  # Replace with your psql password.
```

(Continued on the next page.)

Now, edit your web.py script, near the top, to be similar to this. The first five lines should already be in your script:

```
# If you need custom modules, uncomment the following line, create a
# lib directory under your wsgi directory, and place your modules there.
# Make sure that file/directory permissions are set appropriately.
#
#import sys; sys.path.insert(0, 'lib')

# Replace my phoenix username with your phoenix username
import sys; sys.path.insert(0, '/home/kelliher/.secrets')

# Import the file in your ~/.secrets diretory that holds your DB
# credentials.
import psqlauth
```

Now, change your web.py script's code that creates the database object from something similar to

```
# Create the database connection.
db = web.database(dbn='postgres', user='kelliher', pw='SixByNineFortyTwo'
                  db='kelliher')
```

to something similar to

```
# Create the database connection.
db = web.database(dbn='postgres', user=psqlauth.user, pw=psqlauth.pw,
                  db='kelliher')
```

You're done.