Infos zu Python

Wenn an pip etwas konfiguriert wird dann wird eine Datei `%APPDATA%\pip\pip.ini`(windows) `$HOME/.config/pip/pip.conf`(linux) angelegt. [1)]

Python und pip hinter einem Proxy (ohne Authentifizierung)

Diesen Fehler/Fall habe ich aus der Windows-Welt, es kann sein, dass die gleichen Befehle unter Linux auch funktionieren.

Ich habe schon mal den Fall gehabt, dass ich hinter einem Proxy sitze der ein eigenes Zertifikat mitbringt. Dazu habe ich auch passende Fehlermeldungen erhalten. Dies kam einem Man-in-the-middle Angriff gleich weil der Proxy die Verbindung aufbricht und dann mit seinem eigenen Zertifikat wieder verschlüsselt hat.

```
```

In diesem konkreten Fall hatte ich auch noch ein paar andere selbst erstellte Firmen-Zertifikate denen ich vertrauen wollte/musste.

Als erstes musste ich die Zertifikate alle in einer Datei zusammenführen

```
cat cert1.crt cert2.crt > Zertifikat_Sammlung.crt
```

Das hier geht auch in Powershell 🙂

Danach musste ich diese Sammlung als „trustworthy“ einstellen

```
pip config set global.cert <pfad_zur_datei>Zertifikat_Sammlung.crt
```

[2)]

Seiten von denen ich gelernt habe:

- Python Kurs / Python Course
  Deutsche Seite durchsuchen / Englische Seite durchsuchen
- Stackoverflow python or python3 tagged

Python in verschiedenen Versionen auf Linux installieren

- https://stackoverflow.com/questions/2547554/multiple-python-versions-on-the-same-machine/46258340#46258340
- `Ignoring ensurepip failure: pip 9.0.1 requires SSL/TLS` → https://stackoverflow.com/questions/37723236/pip-error-while-installing-python-ignoring-ensurepip-failure-pip-8-1-1-requir/57219398#57219398

Weitere hilfreiche Links

- zum modul subprocess — Subprocess management
    - Befehle mit pipe

1. ```python
   #shutdown_oralce_db.py
   import subprocess
   import sys
   def shutdown_db():
       '''Stop the oracle database in a clean mode but as fast as
   possible
       sqlplus / as sysdba -> shutdown immediate'''
       print('Starting to shutdown the DB')
       try:
           cmd = ['sqlplus', '/', 'as', 'sysdba']
           addinp = b'shutdown immediate'
           print(str(cmd)+'|'+str(addinp))
           subprocess.check_output(cmd, input=addinp)
           print('DB Shut down')
       except subprocess.CalledProcessError:
           print('Error while Callign a subprocess')
           print(subprocess.CalledProcessError)
       except Exception as e:
           print("Error while shutting down the DB")
           print(e)
           send_mail()
           sys.exit(12)
   ```

2. **an other more complex version when more than one Pipe is needed**

   ```python
   #subprocess_pipes.py

   import subprocess

   cat = subprocess.Popen(
       ['cat', 'index.rst'],
       stdout=subprocess.PIPE,
   )

   grep = subprocess.Popen(
       ['grep', '.. literalinclude::'],
       stdin=cat.stdout,
       stdout=subprocess.PIPE,
   )

   cut = subprocess.Popen(
       ['cut', '-f', '3', '-d:'],
       stdin=grep.stdout,
       stdout=subprocess.PIPE,
   )
   ```
   ```

```
end_of_pipe = cut.stdout

print('Included files:')
for line in end_of_pipe:
    print(line.decode('utf-8').strip())
```

The example reproduces the command line:

```
$ cat index.rst | grep ".. literalinclude" | cut -f 3 -d:
```

3)

1)
https://pip.pypa.io/en/latest/user_guide/#config-file
2)
https://stackoverflow.com/a/52961564
3)
https://pymotw.com/3/subprocess/#connecting-segments-of-a-pipe