Installing uncompyle6 in Kali Linux

Python Bytecode Decompiler:

Compiled to Source code conversion

```
root@kali:~# uncompyle6 --version
bash: uncompyle6: command not found
root@kali:~# python --version
Python 2.7.14+
root@kali:~#
```
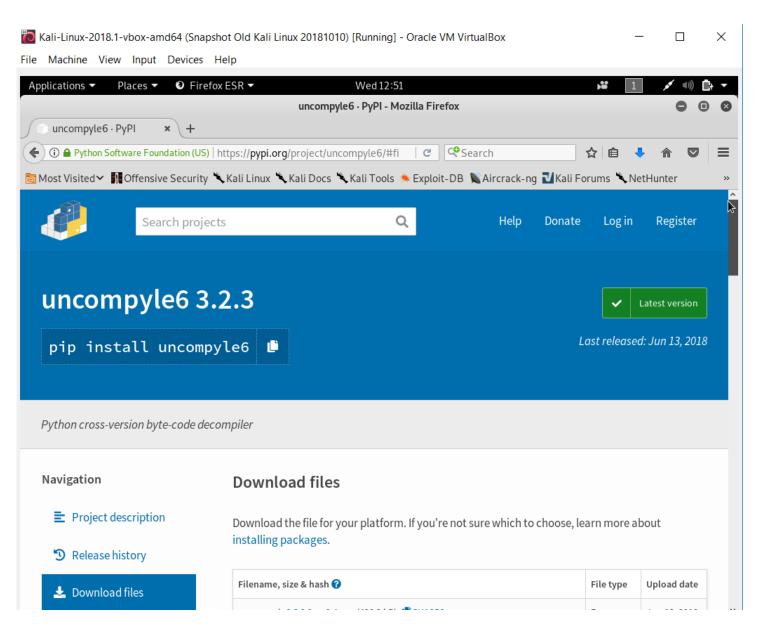
Check to see if you have 'uncompyle6' installed

- Type:
  uncompyle6 --version
- If not, check your Python version by typing:
  python --version

Browse to:
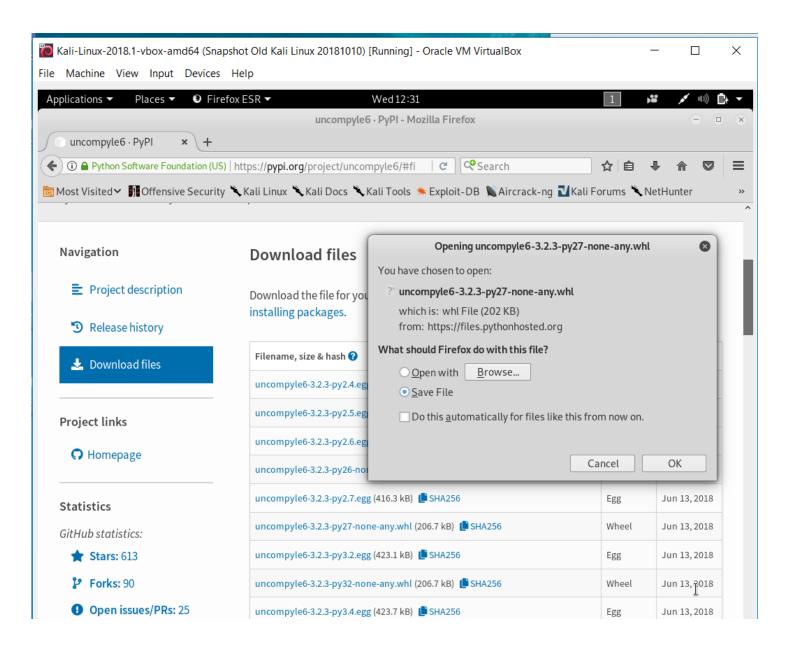pypi.org/project/uncompyle6/
to download the latest version – 3.2.3

- Navigation – Download files

- Find the .whl file that matches your Python version.

Download uncompyle6 to Kali Linux

- Find the file which matches Python version 2.7
- Select the Wheel file to download:

uncompyle6-3.2.3-py27-none-any.whl

Check your Downloads directory for uncompyle6 install file

- Navigate to your Downloads directory
- Check for .whl file

Install uncompyle6 using pip

- Check that you have pip installed by typing:

   pip --version

- If so, then install by typing:

pip install uncompyle6-3.2.3-py27-none-any.whl

Check to see if uncompyle6 has installed

- Type:

uncompyle6 --version

Run uncompyle6

- To decompile a .pyc file, type:

uncompyle6 <filename>.pyc


```
-rw-rw-rw-  1 root root       732 Oct  8 22:24 PYTHON2.pyc
-rw-rw-rw-  1 root root       808 Oct 11 09:28 PYTHON3.PY
-rw-rw-rw-  1 root root       641 Oct 11 10:43 PYTHON3.pyc
-rw-rw-rw-  1 root root      8997 Oct  9 09:55 RE1_64bit
-rwxrwxrwx  1 root root      9223 Oct  9 14:26 RE2_64bit
-rw-rw-rw-  1 root root    169103 Oct 10 09:21 STEG1.jpg
-rw-rw-rw-  1 root root    924174 Oct 10 09:21 STEG2.bmp
-rw-rw-rw-  1 root root   1370874 Oct 10 09:21 STEG3.bmp
-rw-rw-rw-  1 root root   2764854 Oct 10 09:22 Steg4.bmp
drwxrwxrwx  3 root root      4096 Oct 12 07:55 Tables
root@kali:~/Desktop/2018 NCL Gym# uncompyle6 PYTHON2.pyc
# uncompyle6 version 3.2.3
# Python bytecode 2.7 (62211)
# Decompiled from: Python 2.7.15 (default, Jul 28 2018, 11:29:29)
# [GCC 8.1.0]
# Embedded file name: NCL-2015-Python2.py
# Compiled at: 2015-11-12 14:43:01
import sys

def main():
    if len(sys.argv) != 2:
        print 'Invalid args'
        return
    password = sys.argv[1]
    counter = 0
    vals = list('tfzbwlyzljylawhzzdvyk')
    if len(password) != len(vals):
        print 'incorrect'
        return
    while counter < len(password):
        x = ord(password[counter]) + 7
        if x > ord('z'):
            x -= 26
        if chr(x) != vals[counter]:
            print 'incorrect'
            return
        counter += 1

    print 'correct'
```