python源码保护

转自原文：https://www.cnblogs.com/dhcn/p/11077447.html

1. 使用Cython -- 以下内容经过本人测试OK

1.1 思路

虽说 Cython 的主要目的是带来性能的提升，但是基于它的原理：将 .py/.pyx 编译为 .c 文件，再将 .c 文件编译为 .so(Unix) 或 .pyd(Windows)，其带来的另一个好处就是难以破解。

1.2 方法

使用 Cython 进行开发的步骤也不复杂。

1）编写文件 hello.pyx 或 hello.py

```
def hello():
    print('hello')
```

2）编写 setup.py

```
from distutils.core import setup
from Cython.Build import cythonize

setup(name='Hello World app',
     ext_modules=cythonize('hello.pyx'))
```

3）编译为 .c，再进一步编译为 .so 或 .pyd #此处生成出来的so文件不用改名

```
python setup.py build_ext --inplace
```

执行 python -c "from hello import hello;hello()" 即可直接引用生成的二进制文件中的 hello() 函数。

1.3 优点

生成的二进制 .so 或 .pyd 文件难以破解

同时带来了性能提升

另外一方面的好处，其实还可以帮检查语法，而且语法检查强度看起来比python强一点，比如可以检测到变量没有赋值就使用，或者使用了python的关键字作为变量名或def function名等。

1.4 缺点

兼容性稍差，对于不同版本的操作系统，可能需要重新编译

虽然支持大多数 Python 代码，但如果一旦发现部分代码不支持，完善成本较高