Reverse 400

py2exe 的逆向。就在两周前有幸听到了 @seaeast 师兄 AK Flareon2015 的分享，里面也有一道 py2exe 的逆向题，官方给出的 writeup 就是最好的参考资料： Challenge #3 Solution

这里面最重要的部分就是用到了 PyInstaller Extractor 这个工具，下载来后直接执行 `python pyinstxtractor.py Revesre03.exe` 可以得到源文件。打开 **Revesre03.exe_extracted/Revesre03** 就能够得到这道题目的源代码了：

```
data = \
"\x1c\x7a\x16\x77\x10\x2a\x51\x1f\x4c\x0f\x5b\x1d\x42\x2f
\x4b\x7e\x4a\x7a\x4a\x7b" +\
"\x49\x7f\x4a\x7f\x1e\x78\x4c\x75\x10\x28\x18\x2b\x48\x7e
\x46\x23\x12\x24\x11\x72" +\
"\x4b\x2e\x1b\x7e\x4f\x2b\x12\x76\x0b"

'''
char buf[] = "flag:{NSCTF_md5098f6bcd4621d373cade4e832627
b4f6}";

int _tmain(int argc, _TCHAR* argv[])
{
    printf("%d\n", strlen(buf));
    char key = '\x0b';
    buf[47] ^= key;
    for (int i = 1; i < 48; i++)
    {
        buf[48 - i - 1] ^= buf[48 - i];
    }

    return 0;
}
'''

print "Revese it?????????"
```

这是一段 Python 代码并嵌入了一段 C 的注释， buf 里的 flag 并不是真正的 flag。

将 C 语言提取出来编译运行发现 encode 之后的 buf 和上面的 data 很像，尤其是首尾是完全一样的，既然 `encode(buf) ≈ data` ，那么 `decode(data) ≈ buf` ，所以猜测对 data 进行逆向解码就是真正的 flag。解码很简单：

```
data = \
"\x1c\x7a\x16\x77\x10\x2a\x51\x1f\x4c\x0f\x5b\x1d\x42\x2f
\x4b\x7e\x4a\x7a\x4a\x7b" +\
"\x49\x7f\x4a\x7f\x1e\x78\x4c\x75\x10\x28\x18\x2b\x48\x7e
\x46\x23\x12\x24\x11\x72" +\
"\x4b\x2e\x1b\x7e\x4f\x2b\x12\x76\x0b"

flag = ""
for i in range(len(data)-1):
    flag += chr(ord(data[i]) ^ ord(data[i+1]))

print flag
# flag:{NSCTF_md540012655af49e803c68e165c9e5e1d9d}
```