

OSNAP: Python App Deployment using Embedded Python

James Abel

Sep 14, 2016

j@abel.co

@jamesabel

www.abel.co

main.py x	
1	
2	<code>print('hello world')</code>
3	

**Give your
Python
application
to regular
users**



<http://manicallydepressedmoderatelyfunny.blogspot.com/2015/01/halo-5-beta-livedierepeat.html>

- Requirements

- Does not rely on an existing Python environment
- Works with essentially all Python code and packages
- Looks like a native application to the OS (e.g. .exe, .app)
- Creates a regular installer
- Windows and OSX/MacOS

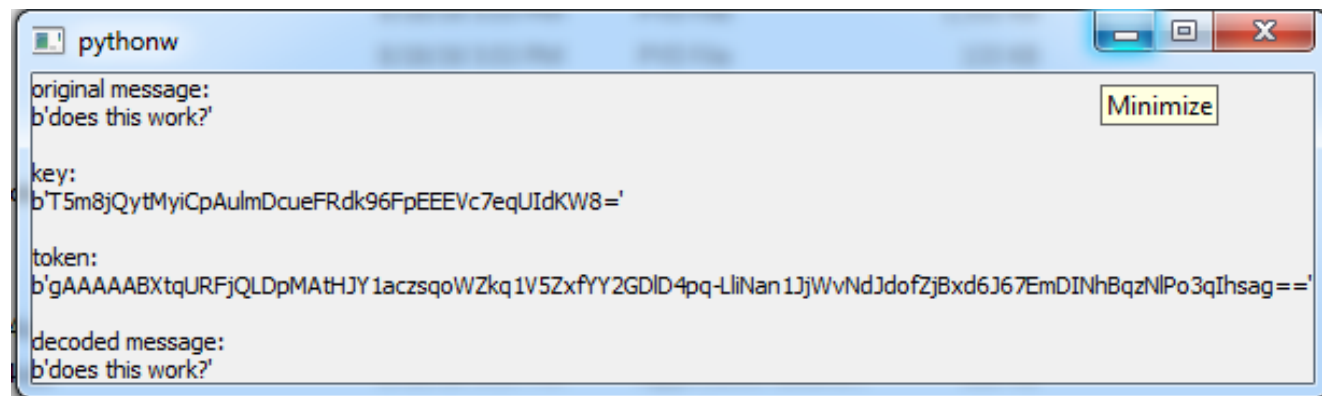
- Existing Tools

- [cx freeze](#)
- [py2exe](#)
- [py2app](#)
- [briefcase](#)
- [pyinstaller](#)
- [pynsist](#)

27 lines (17 sloc) | 530 Bytes

```
1
2 import sys
3
4 from PyQt5.QtWidgets import QApplication, QLabel
5
6 import cryptography.fernet
7
8
9 def run():
10
11     app = QApplication(sys.argv)
12
13     m = b'does this work?'
14     s = 'original message:\n' + str(m) + '\n\n'
15
16     k = cryptography.fernet.Fernet.generate_key()
17     fernet = cryptography.fernet.Fernet(k)
18     t = fernet.encrypt(m)
19     d = fernet.decrypt(t)
20     s += 'key:\n' + str(k) + '\n\n'
21     s += 'token:\n' + str(t) + '\n\n'
22     s += 'decoded message:\n' + str(d)
23
24     window = QLabel(s)
25     window.show()
26
27     app.exec_()
```

Minimal But Effective Test Case



```
c:\temp>testcase.exe
```

```
Traceback (most recent call last):
```

```
File "testcase.py", line 26, in <module>
```

```
File "c:\James\Projects\cryptographybugtestcase\core\crypto.py", line 17, in encrypt
```

```
File "c:\python34\lib\site-packages\cryptography\fernet.py", line 51, in encrypt
```

```
File "c:\python34\lib\site-packages\cryptography\fernet.py", line 61, in  
_encrypt_from_parts
```

```
File "c:\python34\lib\site-packages\cryptography\hazmat\primitives\ciphers\base.py",  
line 43, in encryptor
```

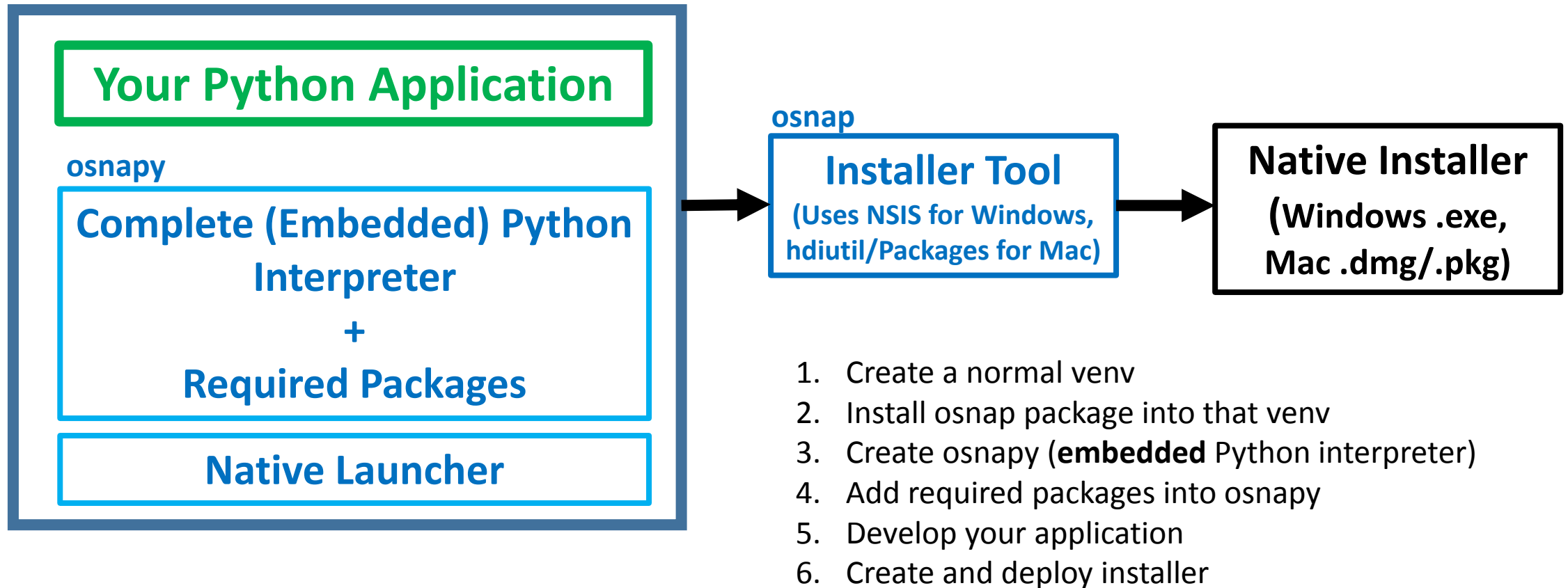
```
File "c:\python34\lib\site-packages\cryptography\hazmat\backends\multibackend.py",  
line 57, in create_symmetric_encryption_ctx
```

```
cryptography.exceptions.UnsupportedAlgorithm: cipher AES in CBC mode is not supported  
by this backend.
```



OSNAP: Overly Simplistic Native Applications tool for Python

complete native application



OSNAP is simplistic ...

- Leverages ***embedded*** Python builds for Windows and OSX
 - Windows: x86-64 embeddable zip file (www.python.org)
 - Mac: eGenix PyRun™ (www.egenix.com/products/python/PyRun/)
 - Python is “embedded” into the launcher
- Not too complex
 - Native Launchers – 82 lines of ***Python*** (not C) code
 - Both Windows and Mac
 - osnapy (independent Python interpreter w/packages) – 80 lines of code
 - Installers - ~300 total lines of code (both Windows and Mac)
- Installer (what’s shipped) is kind of big. Test case:
 - 87 MB on Windows
 - 132 MB on Mac

Get OSNAP

- pip install osnap
- docs: <http://osnap.readthedocs.io/>
- source (including example): <https://github.com/jamesabel/osnap>
- Looking for users and/or collaborators
 - Currently at 0.0.2 ...
 - SF Python Project Night!
- Thank You
 - Thomas Kluyver ([pynsist](#))
 - Glyph
- Contact
 - j@abel.co
 - abel.co

Now on PyPI!

BACKUP

Why is it “overly simplistic”?

- The installer (what you ship to users) is kind of big – even for the small test case:
 - 132 MB for OS X
 - 87 MB for Windows
 - Just for reference, iTunes installers are:
 - 242 MB for OS X (our example app is 55% of that)
 - 122 MB for Windows (our example app is 71% of that)
 - With care our installer size should be able to be reduced
- Also it’s simplistic since our launcher is also written in Python
 - Created with py2exe and py2app
 - Is a simple native program that just executes main

Using embedded Python

- The idea is to have a complete and separate Python interpreter that we can use to run our app against (to make sure it works) and to bundle into the installer.
- Windows embedded zip is in the standard Python install page for 3.5
- [egenix pyrun](#) for Mac
- Doesn't have pip – have to get get-pip.py and install it

The launcher

- Written in Python (to keep it easy to write)
- Use py2exe (Windows) and py2app (Mac) to create the native apps
- Just does a `osnappy/bin/python main.py`
- See `test_example`