
pypcap
Release 1.1.5

Nov 21, 2017

Contents

1	Windows notes	3
2	Installation	5
2.1	Installation from sources	5
3	Support	7
4	Help the Project	9
4.1	Contributing	9
5	Indices and tables	11

This is a simplified object-oriented Python wrapper for libpcap - the current tcpdump.org version, and the WinPcap port for Windows.

Example use:

```
>>> import pcap
>>> sniffer = pcap.pcap(name=None, promisc=True, immediate=True, timeout_ms=50)
>>> addr = lambda pkt, offset: '.'.join(str(ord(pkt[i])) for i in range(offset,
↳offset + 4))
>>> for ts, pkt in sniffer:
...     print('%d\tSRC %-16s\tDST %-16s' % (ts, addr(pkt, sniffer.dloff + 12),
↳addr(pkt, sniffer.dloff + 16)))
...
...

```


CHAPTER 1

Windows notes

WinPcap has compatibility issues with Windows 10, therefore it's recommended to use [Npcap](#) (Nmap's packet sniffing library for Windows, based on the WinPcap/Libpcap libraries, but with improved speed, portability, security, and efficiency). Please enable WinPcap API-compatible mode during the library installation.

This package requires:

- libpcap-dev
- python-dev

To install run

```
pip install pypcap
```

2.1 Installation from sources

Please clone the sources and run:

```
python setup.py install
```

Note for Windows users: Please download the [Npcap SDK](#), unpack the archive and put it into the sibling directory as wpdpack (setup.py will discover it).

Sample procedure in PowerShell:

```
cd ..  
wget -usebasicparsing -outfile npcac-sdk-0.1.zip https://nmap.org/npcap/dist/npcap-  
→sdk-0.1.zip  
Expand-Archive -LiteralPath npcac-sdk-0.1.zip  
mv npcac-sdk-0.1\npcac-sdk-0.1 wpdpack  
cd pypcap  
python setup.py install
```


CHAPTER 3

Support

Visit <https://github.com/pynetwork/pypcap> for help!

4.1 Contributing

4.1.1 Report a Bug or Make a Feature Request

Please go to the GitHub Issues page: <https://github.com/pynetwork/pypcap/issues>.

4.1.2 Checkout the Code

```
git clone https://github.com/pynetwork/pypcap.git
```

4.1.3 Development notes

Regenerating C code

The project uses Cython to generate the C code, it's recommended to install it from sources: <https://github.com/cython/cython>

To regenerate code please use:

```
cython pcap.pyx
```

Building docs

To build docs you need the following additional dependencies:

```
pip install sphinx mock sphinxcontrib.napoleon
```

Please use *build_sphinx* task to regenerate the docs:

```
python setup.py build_sphinx
```

4.1.4 Become a Developer

pypcap uses the ‘GitHub Flow’ model: [GitHub Flow](#)

- To work on something new, create a descriptively named branch off of master (ie: my-awesome)
- Commit to that branch locally and regularly push your work to the same named branch on the server
- When you need feedback or help, or you think the branch is ready for merging, open a pull request
- After someone else has reviewed and signed off on the feature, you can merge it into master

New Feature or Bug

```
$ git checkout -b my-awesome  
$ git push -u origin my-awesome  
$ <code for a bit>; git push  
$ <code for a bit>; git push  
$ tox (this will run all the tests)
```

- Go to github and hit ‘New pull request’
- Someone reviews it and says ‘AOK’
- Merge the pull request (green button)

CHAPTER 5

Indices and tables

- genindex
- modindex