Python ssdeep Documentation

Release 3.2

DinoTools

November 27, 2016

Contents

1 Installation **3**
- 1.1 Requirements . . . 3
- 1.2 Install on CentOS 7 . . . 3
- 1.3 Install on Debian 7 . . . 4
- 1.4 Install on Debian 8 . . . 4
- 1.5 Install on Ubuntu 12.04 . . . 5
- 1.6 Install on Ubuntu 14.04 . . . 5
- 1.7 Install on Ubuntu 16.04 . . . 6

2 Usage **9**

3 API Reference **11**
- 3.1 Classes . . . 11
- 3.2 Functions . . . 12
- 3.3 Exceptions . . . 13

4 FAQ **15**

5 Changelog **17**
- 5.1 3.x (master) . . . 17
- 5.2 3.2 (2016-11-27) . . . 17
- 5.3 3.1.1 (2014-12-20) . . . 17
- 5.4 3.1 (2014-08-07) . . . 17
- 5.5 3.0 (2014-06-25) . . . 17
- 5.6 2.9-0.3 (2013-03-12) . . . 18
- 5.7 2.9-0.2 (2012-10-11) . . . 18
- 5.8 2.9-0.1 (2012-08-01) . . . 18
- 5.9 2.5 (2010-09-03) . . . 18

6 History **19**

7 Indices and tables **21**

This is a straightforward Python wrapper for ssdeep by Jesse Kornblum, which is a library for computing context triggered piecewise hashes (CTPH). Also called fuzzy hashes, CTPH can match inputs that have homologies. Such inputs have sequences of identical bytes in the same order, although bytes in between these sequences may be different in both content and length.

You can install `python-ssdeep` with `pip`:

```
$ pip install ssdeep
```

See Installation for more information.

Contents:

CHAPTER 1

Installation

1.1 Requirements

- Python
  - Python 2.6, 2.7
  - Python >= 3.2
  - PyPy >= 2.0
- ssdeep/libfuzzy >= 2.10 (Some features might not be available with older versions. See *ssdeep.Hash*)
- cffi
- pip
- six

1.2 Install on CentOS 7

1.2.1 Python 2

Use included ssdeep lib

Install required packages.

```
$ sudo yum groupinstall "Development Tools"
$ sudo yum install epel-release
$ sudo yum install libffi-devel python-devel python-pip automake autoconf libtool
```

Build and install Python module.

```
$ sudo BUILD_LIB=1 pip install ssdeep
```

Use lib from epel

Install required packages.

```
$ sudo yum groupinstall "Development Tools"
$ sudo yum install epel-release
$ sudo yum install libffi-devel python-devel python-pip ssdeep-devel ssdeep-lib
```

Build and install Python module.

```
$ sudo pip install ssdeep
```

1.3 Install on Debian 7

1.3.1 Python 2

Use included ssdeep lib

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python python-dev python-pip automake autoconf libt
```

Build and install Python module.

```
$ sudo BUILD_LIB=1 pip install ssdeep
```

1.3.2 Python 3

Use included ssdeep lib

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python3 python3-dev python3-pip automake autoconf
```

Build and install Python module.

```
$ sudo BUILD_LIB=1 pip install ssdeep
```

1.4 Install on Debian 8

1.4.1 Python 2

Use included ssdeep lib

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python python-dev python-pip automake autoconf libt
```

Build and install Python module.

```
$ sudo BUILD_LIB=1 pip install ssdeep
```

Use ssdeep from Debian repository

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python python-dev python-pip libfuzzy-dev
```

Build and install Python module.

```
$ sudo pip install ssdeep
```

1.4.2 Python 3

Use included ssdeep lib

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python3 python3-dev python3-pip automake autoconf l
```

Build and install Python module.

```
$ sudo BUILD_LIB=1 pip3 install ssdeep
```

Use ssdeep from Debian repository

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python3 python3-dev python3-pip libfuzzy-dev
```

Build and install Python module.

```
$ sudo pip3 install ssdeep
```

1.5 Install on Ubuntu 12.04

1.5.1 Python 2

Use included ssdeep lib

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python python-dev python-pip automake autoconf libt
```

Build and install Python module.

```
$ sudo BUILD_LIB=1 pip install ssdeep
```

1.5.2 Python 3

Use included ssdeep lib

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python3 python3-dev python3-setuptools automake aut
```

Build and install Python module.

```
$ sudo easy_install3 pip
$ sudo BUILD_LIB=1 pip3 install ssdeep
```

1.6 Install on Ubuntu 14.04

1.6.1 Python 2

Use included ssdeep lib

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python python-dev python-pip automake autoconf libt
```

Build and install Python module.

```
$ sudo BUILD_LIB=1 pip install ssdeep
```

1.6.2 Python 3

Use included ssdeep lib

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python3 python3-dev python3-pip automake autoconf l
```

Build and install Python module.

```
$ sudo BUILD_LIB=1 pip3 install ssdeep
```

1.7 Install on Ubuntu 16.04

1.7.1 Python 2

Use lib from official Ubuntu repository (recommended)

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python python-dev python-pip libfuzzy-dev
```

Build and install Python module.

```
$ pip install ssdeep
```

Use included ssdeep lib

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python python-dev python-pip automake autoconf libt
```

Build and install Python module.

```
$ BUILD_LIB=1 pip install ssdeep
```

1.7.2 Python 3

Use lib from official Ubuntu repository (recommended)

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python3 python3-dev python3-pip libfuzzy-dev
```

Build and install Python module.

```
$ pip3 install ssdeep
```

Use included ssdeep lib

Install required packages.

```
$ sudo apt-get install build-essential libffi-dev python3 python3-dev python3-pip automake autoconf l
```

Build and install Python module.

```
$ BUILD_LIB=1 pip3 install ssdeep
```

CHAPTER 2

Usage

Import the required module.

```
>>> import ssdeep
```

Use the *ssdeep.hash()* function to compute a fuzzy hash.

```
>>> hash1 = ssdeep.hash('Also called fuzzy hashes, Ctph can match inputs that have homologies.')
>>> hash1
'3:AXGBicFlgVNhBGcL6wCrFQEv:AXGHsNhxLsr2C'
>>> hash2 = ssdeep.hash('Also called fuzzy hashes, CTPH can match inputs that have homologies.')
>>> hash2
'3:AXGBicFlIHBGcL6wCrFQEv:AXGH6xLsr2C'
```

The *ssdeep.compare()* function returns the match score of two hashes. The score is an integer value from 0 (no match) to 100.

```
>>> ssdeep.compare(hash1, hash2)
22
```

The *ssdeep.hash_from_file()* function accepts a filename as argument and calculates the hash of the contents of the file.

```
>>> ssdeep.hash_from_file('/etc/resolv.conf')
'3:S3yE29cFrrMOoiECAaHJgvn:S3m+COoiUCuvn'
```

The *ssdeep.Hash* class provides a hashlib like interface.

```
>>> h = ssdeep.Hash()
>>> h.update('Also called fuzzy hashes, ')
>>> h.digest()
'3:AXGBicFlF:AXGHR'
>>> h.update('Ctph can match inputs that have homologies.')
>>> h.digest()
'3:AXGBicFlgVNhBGcL6wCrFQEv:AXGHsNhxLsr2C'
```

CHAPTER 3

API Reference

3.1 Classes

class ssdeep.**Hash**

Hashlib like object. It is only supported with ssdeep/libfuzzy >= 2.10.

Raises

- **InternalError** – If lib returns internal error
- **NotImplementedError** – Required functions are not available

digest(*elimseq=False*, *notrunc=False*)

Obtain the fuzzy hash.

This operation does not change the state at all. It reports the hash for the concatenation of the data previously fed using update().

Returns The fuzzy hash

Return type String

Raises **InternalError** – If lib returns an internal error

update(*buf*, *encoding='utf-8'*)

Feed the data contained in the given buffer to the state.

Parameters

- **buf** (*String|Byte*) – The data to be hashed
- **encoding** (*String*) – Encoding is used if buf is String

Raises

- **InternalError** – If lib returns an internal error
- **TypeError** – If buf is not Bytes, String or Unicode

class ssdeep.**PseudoHash**

Hashlib like object. Use this class only if Hash() isn't supported by your ssdeep/libfuzzy library. This class stores the provided data in memory, so be careful when hashing large files.

digest(*elimseq=False*, *notrunc=False*)

Obtain the fuzzy hash.

This operation does not change the state at all. It reports the hash for the concatenation of the data previously fed using update().

Returns The fuzzy hash

Return type String

update(*buf*, *encoding='utf-8'*)

Feed the data contained in the given buffer to the state.

Parameters

- **buf** (*String|Byte*) – The data to be hashed
- **encoding** (*String*) – Encoding is used if buf is String

Raises **TypeError** – If buf is not Bytes, String or Unicode

3.2 Functions

ssdeep.**compare**(*sig1*, *sig2*)

Computes the match score between two fuzzy hash signatures.

Returns a value from zero to 100 indicating the match score of the two signatures. A match score of zero indicates the signatures did not match.

Parameters

- **sig1** (*Bytes|String*) – First fuzzy hash signature
- **sig2** (*Bytes|String*) – Second fuzzy hash signature

Returns Match score (0-100)

Return type Integer

Raises

- **InternalError** – If lib returns an internal error
- **TypeError** – If sig is not String, Unicode or Bytes

ssdeep.**hash**(*buf*, *encoding='utf-8'*)

Compute the fuzzy hash of a buffer

Parameters **buf** (*String|Bytes*) – The data to be fuzzy hashed

Returns The fuzzy hash

Return type String

Raises

- **InternalError** – If lib returns an internal error
- **TypeError** – If buf is not String or Bytes

ssdeep.**hash_from_file**(*filename*)

Compute the fuzzy hash of a file.

Opens, reads, and hashes the contents of the file 'filename'

Parameters **filename** (*String|Bytes*) – The name of the file to be hashed

Returns The fuzzy hash of the file

Return type String

Raises

- **IOError** – If Python is unable to read the file
- **InternalError** – If lib returns an internal error

3.3 Exceptions

exception ssdeep.**BaseError**
The base for all other Exceptions

exception ssdeep.**InternalError**
Raised if lib returns internal error

CHAPTER 4

FAQ

If comparing two hashes the result is always 0

The result depends on the algorithms in the ssdeep library. There are some issues if the length of provided data is to short or if the algorithm could not find enough patterns.

The following example must not return the expected value.

```
>>> hash1 = ssdeep.hash('foo' * 4096)
>>> hash2 = ssdeep.hash('foo' * 4096)
>>> ssdeep.compare(hash1, hash2)
0
```

CHAPTER 5

Changelog

5.1 3.x (master)

Note: This version is not yet released and is under development.

5.2 3.2 (2016-11-27)

- Update ssdeep lib to 2.13(thanks to Charles Lindsay)
- Update install instructions
- Add additional CI tests on CentOS 7, Debian 8 and Ubuntu 14.04/16.04

5.3 3.1.1 (2014-12-20)

- Updated ssdeep lib to 2.12
- Added additional tests
- Fixed build issues on Windows(thanks to Paul Chaignon)
- Added option to run tests with PyPy3
- Fixed build to prevent automake version missmatch errors
- Updated documentation

5.4 3.1 (2014-08-07)

- Fix build issue with ssdeep < 2.10

5.5 3.0 (2014-06-25)

- Completely rewritten to use CFFI

- Interface in the spirit of hashlib
- Use pytest and tox for tests
- Use installed fuzzy lib by default

5.6 2.9-0.3 (2013-03-12)

- Fix build issue with Python 2.6

5.7 2.9-0.2 (2012-10-11)

- Fixing small bug in setup.py

5.8 2.9-0.1 (2012-08-01)

- Updated ssdeep from 2.5 to 2.9
- Added Python 3.x support

5.9 2.5 (2010-09-03)

- Initial release

CHAPTER 6

History

- The initial version was published in 2010 by Denis Bilenko on bitbucket.
- Since 2012 the source is maintained by PhiBo (DinoTools) and has been published on github.
- In 2014 the wrapper has been rewritten to use cffi.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

B

BaseError, 13

C

compare() (in module ssdeep), 12

D

digest() (ssdeep.Hash method), 11
digest() (ssdeep.PseudoHash method), 11

H

Hash (class in ssdeep), 11
hash() (in module ssdeep), 12
hash_from_file() (in module ssdeep), 12

I

InternalError, 13

P

PseudoHash (class in ssdeep), 11

U

update() (ssdeep.Hash method), 11
update() (ssdeep.PseudoHash method), 12