Python ssdeep Documentation

Release 3.1

DinoTools

August 07, 2014

Contents

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

1.2 Building python-ssdeep on Linux

`python-ssdeep` should build very easily on Linux.

For Debian and Ubuntu, the following command will ensure that the required dependencies are installed:

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

You should now be able to build and install python-ssdeep.

```
$ pip install ssdeep
```

1.3 Building python-ssdeep on Linux with libfuzzy

If the fuzzy library isn't available on you Linux system. You can use the included lib.

On Debian and Ubuntu the following command will ensure that all additional required dependencies are installed.

```
$ sudo apt-get install automake autoconf
```

You should now be able to build and install python-ssdeep with the included library.

```
$ BUILD_LIB=1 pip 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 3.1 - 2014-08-07

- Fix build issue with ssdeep < 2.10

5.2 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

5.3 2.9-0.3 - 2013-03-12

- Fix build issue with Python 2.6

5.4 2.9-0.2 - 2012-10-11

- Fixing small bug in setup.py

5.5 2.9-0.1 - 2012-08-01

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

5.6 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*