sslkeylog Documentation

Release 0.4.0

Segev Finer

Oct 31, 2021

CONTENTS

CHAPTER

ONE

CHANGELOG

1.1 Unreleased

1.1.1 Added

- CI using GitHub Actions. Sadly can't test on Windows Python 2.7 anymore since Microsoft just nuked the compilers for it from existence some time ago.
- Added *get_server_random* and *export_keying_material*.
- Add a guard to prevent using sslkeylog with a different version of OpenSSL at runtime then build time.

1.2 v0.3.0 (2020-07-10)

1.2.1 Changed

- `set_keylog(None)` will no longer trigger monkey patching, making it easier to use it conditionally.
- Updated documentation to show how to support `SSLKEYLOGFILE`.
- Document that some methods don't work with TLS v1.3 as the values they are meant to return don't exist anymore in TLS v1.3.

1.2.2 Fixed

- Fix tests when TLS v1.3 is enabled by default.

1.3 v0.2.0 (2019-07-16)

1.3.1 Added

- Support for OpenSSL 1.1.1 and TLS 1.3
- Support setting keylog to `None` to disable.

1.3.2 Fixed

- Fix tests on Linux.

1.4 v0.1.1 (2019-01-24)

Fix a broken URL in README.rst.

1.5 v0.1.0 (2019-01-24)

Initial release.

This is an implementation of the `SSLKEYLOGFILE` facility, available in Firefox and Chromium/Google Chrome, that is supported by Wireshark in order to decrypt SSL/TLS connections even when you don't have the private key, or when using key exchange methods that will prevent decryption even if you do (Such as Diffie-Hellman).

This is for the standard library `ssl` module, it won't work for other ssl modules.

Note: Python 3.8+ includes built-in support for generating an SSL key log file via `ssl.SSLContext.keylog_filename`, and will also enable it when the `SSLKEYLOGFILE` environment variable is set when creating a context via `ssl.create_default_context()`.

This package uses the same callback the built-in implementation is using, which will likely cause both implementations to trample each other, causing the other not to work, or other unintended consequences. As such, you should probably not enable both at the same time.

CHAPTER

TWO

SSLKEYLOG

This module provides a facility for logging SSL/TLS keys that can be used for decrypting SSL/TLS connections.

Quickstart:

```
import os
import sslkeylog

sslkeylog.set_keylog(os.environ.get('SSLKEYLOGFILE'))  # Or directly specify a path

# Do anything involving SSL (Using the built-in ssl module)
```

Set the `SSLKEYLOGFILE` environment variable if you use it, and set "(Pre)-Master-Secret log filename" in Wireshark's SSL protocol preferences to the resulting file.

`sslkeylog.`**`get_client_random`**`(`*sock*`)`

Get the client random from an `ssl.SSLSocket` or `ssl.SSLObject`.

> **Note:** Does not work with TLS v1.3+ sockets.

`sslkeylog.`**`get_server_random`**`(`*sock*`)`

Get the server random from an `ssl.SSLSocket` or `ssl.SSLObject`.

> **Note:** Does not work with TLS v1.3+ sockets.

New in version 0.4.0.

`sslkeylog.`**`get_master_key`**`(`*sock*`)`

Get the master key from an `ssl.SSLSocket` or `ssl.SSLObject`.

> **Note:** Does not work with TLS v1.3+ sockets.

`sslkeylog.`**`export_keying_material`**`(`*sock*, *length*, *label*, *context=None*`)`

Obtain keying material for application use from an `ssl.SSLSocket` or `ssl.SSLObject`.

> **Note:** Does not work with SSL v3.0 or below sockets.

New in version 0.4.0.

`sslkeylog.get_keylog_line(sock)`

Generate a key log line from an `ssl.SSLSocket` or `ssl.SSLObject`.

> **Note:** Does not work with TLS v1.3+ sockets.

`sslkeylog.set_keylog(dest)`

Set the key log to *dest* which can be either a path, a file-like object or a callback.

The key log is process-wide and will log keys for all SSL/TLS connections in the process.

A callback will be called with the socket, and a key log line which should be written to the key log.

This will apply the monkey patch needed to implement this if it's not already applied, see *`patch()`*.

`sslkeylog.patch()`

Apply the monkey patch used to implement the key log, if not already patched.

`sslkeylog.unpatch()`

Unapply the monkey patch used to implement the key log, if it was applied.

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

INDEX