:mod:`ssl` --- TLS/SSL wrapper for socket objects

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 4)

Unknown directive type "module".

```
.. module:: ssl
   :synopsis: TLS/SSL wrapper for socket objects
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 7)

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Bill Janssen <bill.janssen@gmail.com>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 8)

Unknown directive type "sectionauthor".

```
.. sectionauthor::  Bill Janssen <bill.janssen@gmail.com>
```

Source code: :source:`Lib/ssl.py`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 10); *backlink*

Unknown interpreted text role "source".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 12)

Unknown directive type "index".

```
.. index:: single: OpenSSL; (use in module ssl)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 14)

Unknown directive type "index".

```
.. index:: TLS, SSL, Transport Layer Security, Secure Sockets Layer
```

This module provides access to Transport Layer Security (often known as "Secure Sockets Layer") encryption and peer authentication facilities for network sockets, both client-side and server-side. This module uses the OpenSSL library. It is available on all modern Unix systems, Windows, macOS, and probably additional platforms, as long as OpenSSL is installed on that platform.

Note

Some behavior may be platform dependent, since calls are made to the operating system socket APIs. The installed version of OpenSSL may also cause variations in behavior. For example, TLSv1.3 with OpenSSL version 1.1.1.

Warning

Don't use this module without reading the :ref:`ssl-security`. Doing so may lead to a false sense of security, as the default settings of the ssl module are not necessarily appropriate for your application.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 32); *backlink*

Unknown interpreted text role "ref".

This section documents the objects and functions in the `ssl` module; for more general information about TLS, SSL, and certificates, the reader is referred to the documents in the "See Also" section at the bottom.

This module provides a class, :class:`ssl.SSLSocket`, which is derived from the :class:`socket.socket` type, and provides a socket-like wrapper that also encrypts and decrypts the data going over the socket with SSL. It supports additional methods such as :meth:`getpeercert`, which retrieves the certificate of the other side of the connection, and :meth:`cipher`, which retrieves the cipher being used for the secure connection.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 41); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 41); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 41); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 41); *backlink*

Unknown interpreted text role "meth".

For more sophisticated applications, the :class:`ssl.SSLContext` class helps manage settings and certificates, which can then be inherited by SSL sockets created through the :meth:`SSLContext.wrap_socket` method.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 48); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 48); backlink

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 52)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.5.3
   Updated to support linking with OpenSSL 1.1.0
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 55)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6

   OpenSSL 0.9.8, 1.0.0 and 1.0.1 are deprecated and no longer supported.
   In the future the ssl module will require at least OpenSSL 1.0.2 or
   1.1.0.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 61)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.10

   :pep:`644` has been implemented. The ssl module requires OpenSSL 1.1.1
   or newer.

   Use of deprecated constants and functions result in deprecation warnings.
```

Functions, Constants, and Exceptions

Socket creation

Since Python 3.2 and 2.7.9, it is recommended to use the :meth:`SSLContext.wrap_socket` of an :class:`SSLContext` instance to wrap sockets as :class:`SSLSocket` objects. The helper functions :func:`create_default_context` returns a new context with secure default settings. The old :func:`wrap_socket` function is deprecated since it is both inefficient and has no support for server name indication (SNI) and hostname matching.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 76); backlink

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 76); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 76); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 76); backlink

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 76); backlink

Unknown interpreted text role "func".

Client socket example with default context and IPv4/IPv6 dual stack:

```
import socket
import ssl

hostname = 'www.python.org'
context = ssl.create_default_context()

with socket.create_connection((hostname, 443)) as sock:
    with context.wrap_socket(sock, server_hostname=hostname) as ssock:
        print(ssock.version())
```

Client socket example with custom context and IPv4:

```
hostname = 'www.python.org'
# PROTOCOL_TLS_CLIENT requires valid cert chain and hostname
context = ssl.SSLContext(ssl.PROTOCOL_TLS_CLIENT)
context.load_verify_locations('path/to/cabundle.pem')

with socket.socket(socket.AF_INET, socket.SOCK_STREAM, 0) as sock:
    with context.wrap_socket(sock, server_hostname=hostname) as ssock:
        print(ssock.version())
```

Server socket example listening on localhost IPv4:

```
context = ssl.SSLContext(ssl.PROTOCOL_TLS_SERVER)
context.load_cert_chain('/path/to/certchain.pem', '/path/to/private.key')
```

```
with socket.socket(socket.AF_INET, socket.SOCK_STREAM, 0) as sock:
    sock.bind(('127.0.0.1', 8443))
    sock.listen(5)
    with context.wrap_socket(sock, server_side=True) as ssock:
        conn, addr = ssock.accept()
        ...
```

Context creation

A convenience function helps create :class:`SSLContext` objects for common purposes.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 125); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 128)

Unknown directive type "function".

```
.. function:: create_default_context(purpose=Purpose.SERVER_AUTH, cafile=None, capath=None, cadata=None)

   Return a new :class:`SSLContext` object with default settings for
   the given *purpose*.  The settings are chosen by the :mod:`ssl` module,
   and usually represent a higher security level than when calling the
   :class:`SSLContext` constructor directly.

   *cafile*, *capath*, *cadata* represent optional CA certificates to
   trust for certificate verification, as in
   :meth:`SSLContext.load_verify_locations`.  If all three are
   :const:`None`, this function can choose to trust the system's default
   CA certificates instead.

   The settings are: :data:`PROTOCOL_TLS_CLIENT` or
   :data:`PROTOCOL_TLS_SERVER`, :data:`OP_NO_SSLv2`, and :data:`OP_NO_SSLv3`
   with high encryption cipher suites without RC4 and
   without unauthenticated cipher suites. Passing :data:`~Purpose.SERVER_AUTH`
   as *purpose* sets :data:`~SSLContext.verify_mode` to :data:`CERT_REQUIRED`
   and either loads CA certificates (when at least one of *cafile*, *capath* or
   *cadata* is given) or uses :meth:`SSLContext.load_default_certs` to load
   default CA certificates.

   When :attr:`~SSLContext.keylog_filename` is supported and the environment
   variable :envvar:`SSLKEYLOGFILE` is set, :func:`create_default_context`
   enables key logging.

   .. note::
      The protocol, options, cipher and other settings may change to more
      restrictive values anytime without prior deprecation.  The values
      represent a fair balance between compatibility and security.

      If your application needs specific settings, you should create a
      :class:`SSLContext` and apply the settings yourself.

   .. note::
      If you find that when certain older clients or servers attempt to connect
      with a :class:`SSLContext` created by this function that they get an error
      stating "Protocol or cipher suite mismatch", it may be that they only
      support SSL3.0 which this function excludes using the
      :data:`OP_NO_SSLv3`. SSL3.0 is widely considered to be `completely broken
      <https://en.wikipedia.org/wiki/POODLE>`_. If you still wish to continue to
      use this function but still allow SSL 3.0 connections you can re-enable
      them using::

         ctx = ssl.create_default_context(Purpose.CLIENT_AUTH)
         ctx.options &= ~ssl.OP_NO_SSLv3

   .. versionadded:: 3.4

   .. versionchanged:: 3.4.4

     RC4 was dropped from the default cipher string.

   .. versionchanged:: 3.6

     ChaCha20/Poly1305 was added to the default cipher string.

     3DES was dropped from the default cipher string.

   .. versionchanged:: 3.8

      Support for key logging to :envvar:`SSLKEYLOGFILE` was added.

   .. versionchanged:: 3.10

      The context now uses :data:`PROTOCOL_TLS_CLIENT` or
      :data:`PROTOCOL_TLS_SERVER` protocol instead of generic
      :data:`PROTOCOL_TLS`.
```

Exceptions

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 201)

Unknown directive type "exception".

```
.. exception:: SSLError

   Raised to signal an error from the underlying SSL implementation
   (currently provided by the OpenSSL library).  This signifies some
   problem in the higher-level encryption and authentication layer that's
   superimposed on the underlying network connection.  This error
   is a subtype of :exc:`OSError`.  The error code and message of
   :exc:`SSLError` instances are provided by the OpenSSL library.

   .. versionchanged:: 3.3
      :exc:`SSLError` used to be a subtype of :exc:`socket.error`.

   .. attribute:: library

      A string mnemonic designating the OpenSSL submodule in which the error
      occurred, such as ``SSL``, ``PEM`` or ``X509``.  The range of possible
      values depends on the OpenSSL version.

      .. versionadded:: 3.3

   .. attribute:: reason

      A string mnemonic designating the reason this error occurred, for
```

```
   example ``CERTIFICATE_VERIFY_FAILED``.  The range of possible
   values depends on the OpenSSL version.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 229)

Unknown directive type "exception".

```
.. exception:: SSLZeroReturnError

   A subclass of :exc:`SSLError` raised when trying to read or write and
   the SSL connection has been closed cleanly.  Note that this doesn't
   mean that the underlying transport (read TCP) has been closed.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 237)

Unknown directive type "exception".

```
.. exception:: SSLWantReadError

   A subclass of :exc:`SSLError` raised by a :ref:`non-blocking SSL socket
   <ssl-nonblocking>` when trying to read or write data, but more data needs
   to be received on the underlying TCP transport before the request can be
   fulfilled.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 246)

Unknown directive type "exception".

```
.. exception:: SSLWantWriteError

   A subclass of :exc:`SSLError` raised by a :ref:`non-blocking SSL socket
   <ssl-nonblocking>` when trying to read or write data, but more data needs
   to be sent on the underlying TCP transport before the request can be
   fulfilled.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 255)

Unknown directive type "exception".

```
.. exception:: SSLSyscallError

   A subclass of :exc:`SSLError` raised when a system error was encountered
   while trying to fulfill an operation on a SSL socket.  Unfortunately,
   there is no easy way to inspect the original errno number.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 263)

Unknown directive type "exception".

```
.. exception:: SSLEOFError

   A subclass of :exc:`SSLError` raised when the SSL connection has been
   terminated abruptly.  Generally, you shouldn't try to reuse the underlying
   transport when this error is encountered.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 271)

Unknown directive type "exception".

```
.. exception:: SSLCertVerificationError

   A subclass of :exc:`SSLError` raised when certificate validation has
   failed.

   .. versionadded:: 3.7

   .. attribute:: verify_code

      A numeric error number that denotes the verification error.

   .. attribute:: verify_message

      A human readable string of the verification error.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 286)

Unknown directive type "exception".

```
.. exception:: CertificateError

   An alias for :exc:`SSLCertVerificationError`.

   .. versionchanged:: 3.7
      The exception is now an alias for :exc:`SSLCertVerificationError`.
```

Random generation

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 297)

Unknown directive type "function".

```
.. function:: RAND_bytes(num)

   Return *num* cryptographically strong pseudo-random bytes. Raises an
   :class:`SSLError` if the PRNG has not been seeded with enough data or if the
   operation is not supported by the current RAND method. :func:`RAND_status`
   can be used to check the status of the PRNG and :func:`RAND_add` can be used
   to seed the PRNG.

   For almost all applications :func:`os.urandom` is preferable.

   Read the Wikipedia article, `Cryptographically secure pseudorandom number
   generator (CSPRNG)
   <https://en.wikipedia.org/wiki/Cryptographically_secure_pseudorandom_number_generator>`_,
   to get the requirements of a cryptographically strong generator.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 314)

Unknown directive type "function".

```
.. function:: RAND_pseudo_bytes(num)

   Return (bytes, is_cryptographic): bytes are *num* pseudo-random bytes,
   is_cryptographic is ``True`` if the bytes generated are cryptographically
   strong. Raises an :class:`SSLError` if the operation is not supported by the
   current RAND method.

   Generated pseudo-random byte sequences will be unique if they are of
   sufficient length, but are not necessarily unpredictable. They can be used
   for non-cryptographic purposes and for certain purposes in cryptographic
   protocols, but usually not for key generation etc.

   For almost all applications :func:`os.urandom` is preferable.

   .. versionadded:: 3.3

   .. deprecated:: 3.6

      OpenSSL has deprecated :func:`ssl.RAND_pseudo_bytes`, use
      :func:`ssl.RAND_bytes` instead.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 335)

Unknown directive type "function".

```
.. function:: RAND_status()

   Return ``True`` if the SSL pseudo-random number generator has been seeded
   with 'enough' randomness, and ``False`` otherwise.  You can use
   :func:`ssl.RAND_egd` and :func:`ssl.RAND_add` to increase the randomness of
   the pseudo-random number generator.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 342)

Unknown directive type "function".

```
.. function:: RAND_add(bytes, entropy)

   Mix the given *bytes* into the SSL pseudo-random number generator.  The
   parameter *entropy* (a float) is a lower bound on the entropy contained in
   string (so you can always use :const:`0.0`).  See :rfc:`1750` for more
   information on sources of entropy.

   .. versionchanged:: 3.5
      Writable :term:`bytes-like object` is now accepted.
```

Certificate handling

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 355)

Unknown directive type "testsetup".

```
.. testsetup::

   import ssl
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 359)

Unknown directive type "function".

```
.. function:: match_hostname(cert, hostname)

   Verify that *cert* (in decoded format as returned by
   :meth:`SSLSocket.getpeercert`) matches the given *hostname*.  The rules
   applied are those for checking the identity of HTTPS servers as outlined
   in :rfc:`2818`, :rfc:`5280` and :rfc:`6125`.  In addition to HTTPS, this
   function should be suitable for checking the identity of servers in
   various SSL-based protocols such as FTPS, IMAPS, POPS and others.

   :exc:`CertificateError` is raised on failure. On success, the function
   returns nothing::

      >>> cert = {'subject': ((('commonName', 'example.com'),),)}
      >>> ssl.match_hostname(cert, "example.com")
      >>> ssl.match_hostname(cert, "example.org")
      Traceback (most recent call last):
        File "<stdin>", line 1, in <module>
        File "/home/py3k/Lib/ssl.py", line 130, in match_hostname
      ssl.CertificateError: hostname 'example.org' doesn't match 'example.com'

   .. versionadded:: 3.2

   .. versionchanged:: 3.3.3
      The function now follows :rfc:`6125`, section 6.4.3 and does neither
      match multiple wildcards (e.g. ``*.*.com`` or ``*a*.example.org``) nor
      a wildcard inside an internationalized domain names (IDN) fragment.
      IDN A-labels such as ``www*.xn--pthon-kva.org`` are still supported,
```

```
      but ``x*.python.org`` no longer matches ``xn--tda.python.org``.

   .. versionchanged:: 3.5
      Matching of IP addresses, when present in the subjectAltName field
      of the certificate, is now supported.

   .. versionchanged:: 3.7
      The function is no longer used to TLS connections. Hostname matching
      is now performed by OpenSSL.

      Allow wildcard when it is the leftmost and the only character
      in that segment. Partial wildcards like ``www*.example.com`` are no
      longer supported.

   .. deprecated:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 402)

Unknown directive type "function".

```
.. function:: cert_time_to_seconds(cert_time)

   Return the time in seconds since the Epoch, given the ``cert_time``
   string representing the "notBefore" or "notAfter" date from a
   certificate in ``"%b %d %H:%M:%S %Y %Z"`` strptime format (C
   locale).

   Here's an example:

   .. doctest:: newcontext

      >>> import ssl
      >>> timestamp = ssl.cert_time_to_seconds("Jan  5 09:34:43 2018 GMT")
      >>> timestamp  # doctest: +SKIP
      1515144883
      >>> from datetime import datetime
      >>> print(datetime.utcfromtimestamp(timestamp))  # doctest: +SKIP
      2018-01-05 09:34:43

   "notBefore" or "notAfter" dates must use GMT (:rfc:`5280`).

   .. versionchanged:: 3.5
      Interpret the input time as a time in UTC as specified by 'GMT'
      timezone in the input string. Local timezone was used
      previously. Return an integer (no fractions of a second in the
      input format)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 429)

Unknown directive type "function".

```
.. function:: get_server_certificate(addr, ssl_version=PROTOCOL_TLS_CLIENT, \
                                     ca_certs=None[, timeout])

   Given the address ``addr`` of an SSL-protected server, as a (*hostname*,
   *port-number*) pair, fetches the server's certificate, and returns it as a
   PEM-encoded string.  If ``ssl_version`` is specified, uses that version of
   the SSL protocol to attempt to connect to the server.  If ``ca_certs`` is
   specified, it should be a file containing a list of root certificates, the
   same format as used for the same parameter in
   :meth:`SSLContext.wrap_socket`.  The call will attempt to validate the
   server certificate against that set of root certificates, and will fail
   if the validation attempt fails.  A timeout can be specified with the
   ``timeout`` parameter.

   .. versionchanged:: 3.3
      This function is now IPv6-compatible.

   .. versionchanged:: 3.5
      The default *ssl_version* is changed from :data:`PROTOCOL_SSLv3` to
      :data:`PROTOCOL_TLS` for maximum compatibility with modern servers.

   .. versionchanged:: 3.10
      The *timeout* parameter was added.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 453)

Unknown directive type "function".

```
.. function:: DER_cert_to_PEM_cert(DER_cert_bytes)

   Given a certificate as a DER-encoded blob of bytes, returns a PEM-encoded
   string version of the same certificate.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 458)

Unknown directive type "function".

```
.. function:: PEM_cert_to_DER_cert(PEM_cert_string)

   Given a certificate as an ASCII PEM string, returns a DER-encoded sequence of
   bytes for that same certificate.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 463)

Unknown directive type "function".

```
.. function:: get_default_verify_paths()

   Returns a named tuple with paths to OpenSSL's default cafile and capath.
   The paths are the same as used by
   :meth:`SSLContext.set_default_verify_paths`. The return value is a
   :term:`named tuple` ``DefaultVerifyPaths``:

   * :attr:`cafile` - resolved path to cafile or ``None`` if the file doesn't exist,
   * :attr:`capath` - resolved path to capath or ``None`` if the directory doesn't exist,
   * :attr:`openssl_cafile_env` - OpenSSL's environment key that points to a cafile,
   * :attr:`openssl_cafile` - hard coded path to a cafile,
   * :attr:`openssl_capath_env` - OpenSSL's environment key that points to a capath,
   * :attr:`openssl_capath` - hard coded path to a capath directory
```

```
   .. availability:: LibreSSL ignores the environment vars
     :attr:`openssl_cafile_env` and :attr:`openssl_capath_env`.

   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 482)

Unknown directive type "function".

```
.. function:: enum_certificates(store_name)

   Retrieve certificates from Windows' system cert store. *store_name* may be
   one of ``CA``, ``ROOT`` or ``MY``. Windows may provide additional cert
   stores, too.

   The function returns a list of (cert_bytes, encoding_type, trust) tuples.
   The encoding_type specifies the encoding of cert_bytes. It is either
   :const:`x509_asn` for X.509 ASN.1 data or :const:`pkcs_7_asn` for
   PKCS#7 ASN.1 data. Trust specifies the purpose of the certificate as a set
   of OIDS or exactly ``True`` if the certificate is trustworthy for all
   purposes.

   Example::

      >>> ssl.enum_certificates("CA")
      [(b'data...', 'x509_asn', {'1.3.6.1.5.5.7.3.1', '1.3.6.1.5.5.7.3.2'}),
       (b'data...', 'x509_asn', True)]

   .. availability:: Windows.

   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 505)

Unknown directive type "function".

```
.. function:: enum_crls(store_name)

   Retrieve CRLs from Windows' system cert store. *store_name* may be
   one of ``CA``, ``ROOT`` or ``MY``. Windows may provide additional cert
   stores, too.

   The function returns a list of (cert_bytes, encoding_type, trust) tuples.
   The encoding_type specifies the encoding of cert_bytes. It is either
   :const:`x509_asn` for X.509 ASN.1 data or :const:`pkcs_7_asn` for
   PKCS#7 ASN.1 data.

   .. availability:: Windows.

   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 520)

Unknown directive type "function".

```
.. function:: wrap_socket(sock, keyfile=None, certfile=None, \
       server_side=False, cert_reqs=CERT_NONE, ssl_version=PROTOCOL_TLS, \
       ca_certs=None, do_handshake_on_connect=True, \
       suppress_ragged_eofs=True, ciphers=None)

   Takes an instance ``sock`` of :class:`socket.socket`, and returns an instance
   of :class:`ssl.SSLSocket`, a subtype of :class:`socket.socket`, which wraps
   the underlying socket in an SSL context.  ``sock`` must be a
   :data:`~socket.SOCK_STREAM` socket; other socket types are unsupported.

   Internally, function creates a :class:`SSLContext` with protocol
   *ssl_version* and :attr:`SSLContext.options` set to *cert_reqs*. If
   parameters *keyfile*, *certfile*, *ca_certs* or *ciphers* are set, then
   the values are passed to :meth:`SSLContext.load_cert_chain`,
   :meth:`SSLContext.load_verify_locations`, and
   :meth:`SSLContext.set_ciphers`.

   The arguments *server_side*, *do_handshake_on_connect*, and
   *suppress_ragged_eofs* have the same meaning as
   :meth:`SSLContext.wrap_socket`.

   .. deprecated:: 3.7

      Since Python 3.2 and 2.7.9, it is recommended to use the
      :meth:`SSLContext.wrap_socket` instead of :func:`wrap_socket`. The
      top-level function is limited and creates an insecure client socket
      without server name indication or hostname matching.
```

Constants

All constants are now :class:`enum.IntEnum` or :class:`enum.IntFlag` collections.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 551); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 551); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 553)

Unknown directive type "versionadded".

```
   .. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 555)

Unknown directive type "data".

```
.. data:: CERT_NONE

   Possible value for :attr:`SSLContext.verify_mode`, or the ``cert_reqs``
   parameter to :func:`wrap_socket`.  Except for :const:`PROTOCOL_TLS_CLIENT`,
   it is the default mode.  With client-side sockets, just about any
   cert is accepted.  Validation errors, such as untrusted or expired cert,
   are ignored and do not abort the TLS/SSL handshake.

   In server mode, no certificate is requested from the client, so the client
   does not send any for client cert authentication.

   See the discussion of :ref:`ssl-security` below.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 568)

Unknown directive type "data".

```
.. data:: CERT_OPTIONAL

   Possible value for :attr:`SSLContext.verify_mode`, or the ``cert_reqs``
   parameter to :func:`wrap_socket`.  In client mode, :const:`CERT_OPTIONAL`
   has the same meaning as :const:`CERT_REQUIRED`. It is recommended to
   use :const:`CERT_REQUIRED` for client-side sockets instead.

   In server mode, a client certificate request is sent to the client.  The
   client may either ignore the request or send a certificate in order
   perform TLS client cert authentication.  If the client chooses to send
   a certificate, it is verified.  Any verification error immediately aborts
   the TLS handshake.

   Use of this setting requires a valid set of CA certificates to
   be passed, either to :meth:`SSLContext.load_verify_locations` or as a
   value of the ``ca_certs`` parameter to :func:`wrap_socket`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 585)

Unknown directive type "data".

```
.. data:: CERT_REQUIRED

   Possible value for :attr:`SSLContext.verify_mode`, or the ``cert_reqs``
   parameter to :func:`wrap_socket`.  In this mode, certificates are
   required from the other side of the socket connection; an :class:`SSLError`
   will be raised if no certificate is provided, or if its validation fails.
   This mode is **not** sufficient to verify a certificate in client mode as
   it does not match hostnames.  :attr:`~SSLContext.check_hostname` must be
   enabled as well to verify the authenticity of a cert.
   :const:`PROTOCOL_TLS_CLIENT` uses :const:`CERT_REQUIRED` and
   enables :attr:`~SSLContext.check_hostname` by default.

   With server socket, this mode provides mandatory TLS client cert
   authentication.  A client certificate request is sent to the client and
   the client must provide a valid and trusted certificate.

   Use of this setting requires a valid set of CA certificates to
   be passed, either to :meth:`SSLContext.load_verify_locations` or as a
   value of the ``ca_certs`` parameter to :func:`wrap_socket`.
```

:class:`enum.IntEnum` collection of CERT_* constants.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 607); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 609)

Unknown directive type "versionadded".

```
.. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 611)

Unknown directive type "data".

```
.. data:: VERIFY_DEFAULT

   Possible value for :attr:`SSLContext.verify_flags`. In this mode, certificate
   revocation lists (CRLs) are not checked. By default OpenSSL does neither
   require nor verify CRLs.

   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 619)

Unknown directive type "data".

```
.. data:: VERIFY_CRL_CHECK_LEAF

   Possible value for :attr:`SSLContext.verify_flags`. In this mode, only the
   peer cert is checked but none of the intermediate CA certificates. The mode
   requires a valid CRL that is signed by the peer cert's issuer (its direct
   ancestor CA). If no proper CRL has been loaded with
   :attr:`SSLContext.load_verify_locations`, validation will fail.

   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 629)

Unknown directive type "data".

```
.. data:: VERIFY_CRL_CHECK_CHAIN

   Possible value for :attr:`SSLContext.verify_flags`. In this mode, CRLs of
```

```
   all certificates in the peer cert chain are checked.

   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 636)

Unknown directive type "data".

```
.. data:: VERIFY_X509_STRICT

   Possible value for :attr:`SSLContext.verify_flags` to disable workarounds
   for broken X.509 certificates.

   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 643)

Unknown directive type "data".

```
.. data:: VERIFY_ALLOW_PROXY_CERTS

   Possible value for :attr:`SSLContext.verify_flags` to enables proxy
   certificate verification.

   .. versionadded:: 3.10
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 650)

Unknown directive type "data".

```
.. data:: VERIFY_X509_TRUSTED_FIRST

   Possible value for :attr:`SSLContext.verify_flags`. It instructs OpenSSL to
   prefer trusted certificates when building the trust chain to validate a
   certificate. This flag is enabled by default.

   .. versionadded:: 3.4.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 658)

Unknown directive type "data".

```
.. data:: VERIFY_X509_PARTIAL_CHAIN

   Possible value for :attr:`SSLContext.verify_flags`. It instructs OpenSSL to
   accept intermediate CAs in the trust store to be treated as trust-anchors,
   in the same way as the self-signed root CA certificates. This makes it
   possible to trust certificates issued by an intermediate CA without having
   to trust its ancestor root CA.

   .. versionadded:: 3.10
```

:class:`enum.IntFlag` collection of VERIFY_* constants.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 671); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 673)

Unknown directive type "versionadded".

```
.. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 675)

Unknown directive type "data".

```
.. data:: PROTOCOL_TLS

   Selects the highest protocol version that both the client and server support.
   Despite the name, this option can select both "SSL" and "TLS" protocols.

   .. versionadded:: 3.6

   .. deprecated:: 3.10

      TLS clients and servers require different default settings for secure
      communication. The generic TLS protocol constant is deprecated in
      favor of :data:`PROTOCOL_TLS_CLIENT` and :data:`PROTOCOL_TLS_SERVER`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 688)

Unknown directive type "data".

```
.. data:: PROTOCOL_TLS_CLIENT

   Auto-negotiate the highest protocol version that both the client and
   server support, and configure the context client-side connections. The
   protocol enables :data:`CERT_REQUIRED` and
   :attr:`~SSLContext.check_hostname` by default.

   .. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 697)

Unknown directive type "data".

```
   .. data:: PROTOCOL_TLS_SERVER

      Auto-negotiate the highest protocol version that both the client and
      server support, and configure the context server-side connections.

      .. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 704)

Unknown directive type "data".

```
   .. data:: PROTOCOL_SSLv23

      Alias for :data:`PROTOCOL_TLS`.

      .. deprecated:: 3.6

         Use :data:`PROTOCOL_TLS` instead.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 712)

Unknown directive type "data".

```
   .. data:: PROTOCOL_SSLv2

      Selects SSL version 2 as the channel encryption protocol.

      This protocol is not available if OpenSSL is compiled with the
      ``OPENSSL_NO_SSL2`` flag.

      .. warning::

         SSL version 2 is insecure.  Its use is highly discouraged.

      .. deprecated:: 3.6

         OpenSSL has removed support for SSLv2.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 727)

Unknown directive type "data".

```
   .. data:: PROTOCOL_SSLv3

      Selects SSL version 3 as the channel encryption protocol.

      This protocol is not be available if OpenSSL is compiled with the
      ``OPENSSL_NO_SSLv3`` flag.

      .. warning::

         SSL version 3 is insecure.  Its use is highly discouraged.

      .. deprecated:: 3.6

         OpenSSL has deprecated all version specific protocols. Use the default
         protocol :data:`PROTOCOL_TLS_SERVER` or :data:`PROTOCOL_TLS_CLIENT`
         with :attr:`SSLContext.minimum_version` and
         :attr:`SSLContext.maximum_version` instead.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 746)

Unknown directive type "data".

```
   .. data:: PROTOCOL_TLSv1

      Selects TLS version 1.0 as the channel encryption protocol.

      .. deprecated:: 3.6

         OpenSSL has deprecated all version specific protocols.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 754)

Unknown directive type "data".

```
   .. data:: PROTOCOL_TLSv1_1

      Selects TLS version 1.1 as the channel encryption protocol.
      Available only with openssl version 1.0.1+.

      .. versionadded:: 3.4

      .. deprecated:: 3.6

         OpenSSL has deprecated all version specific protocols.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 765)

Unknown directive type "data".

```
   .. data:: PROTOCOL_TLSv1_2

      Selects TLS version 1.2 as the channel encryption protocol.
      Available only with openssl version 1.0.1+.

      .. versionadded:: 3.4

      .. deprecated:: 3.6

         OpenSSL has deprecated all version specific protocols.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 776)

Unknown directive type "data".

```
.. data:: OP_ALL

   Enables workarounds for various bugs present in other SSL implementations.
   This option is set by default.  It does not necessarily set the same
   flags as OpenSSL's ``SSL_OP_ALL`` constant.

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 784)

Unknown directive type "data".

```
.. data:: OP_NO_SSLv2

   Prevents an SSLv2 connection.  This option is only applicable in
   conjunction with :const:`PROTOCOL_TLS`.  It prevents the peers from
   choosing SSLv2 as the protocol version.

   .. versionadded:: 3.2

   .. deprecated:: 3.6

      SSLv2 is deprecated
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 796)

Unknown directive type "data".

```
.. data:: OP_NO_SSLv3

   Prevents an SSLv3 connection.  This option is only applicable in
   conjunction with :const:`PROTOCOL_TLS`.  It prevents the peers from
   choosing SSLv3 as the protocol version.

   .. versionadded:: 3.2

   .. deprecated:: 3.6

      SSLv3 is deprecated
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 808)

Unknown directive type "data".

```
.. data:: OP_NO_TLSv1

   Prevents a TLSv1 connection.  This option is only applicable in
   conjunction with :const:`PROTOCOL_TLS`.  It prevents the peers from
   choosing TLSv1 as the protocol version.

   .. versionadded:: 3.2

   .. deprecated:: 3.7
      The option is deprecated since OpenSSL 1.1.0, use the new
      :attr:`SSLContext.minimum_version` and
      :attr:`SSLContext.maximum_version` instead.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 821)

Unknown directive type "data".

```
.. data:: OP_NO_TLSv1_1

   Prevents a TLSv1.1 connection. This option is only applicable in conjunction
   with :const:`PROTOCOL_TLS`. It prevents the peers from choosing TLSv1.1 as
   the protocol version. Available only with openssl version 1.0.1+.

   .. versionadded:: 3.4

   .. deprecated:: 3.7
      The option is deprecated since OpenSSL 1.1.0.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 832)

Unknown directive type "data".

```
.. data:: OP_NO_TLSv1_2

   Prevents a TLSv1.2 connection. This option is only applicable in conjunction
   with :const:`PROTOCOL_TLS`. It prevents the peers from choosing TLSv1.2 as
   the protocol version. Available only with openssl version 1.0.1+.

   .. versionadded:: 3.4

   .. deprecated:: 3.7
      The option is deprecated since OpenSSL 1.1.0.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 843)

Unknown directive type "data".

```
.. data:: OP_NO_TLSv1_3

   Prevents a TLSv1.3 connection. This option is only applicable in conjunction
   with :const:`PROTOCOL_TLS`. It prevents the peers from choosing TLSv1.3 as
   the protocol version. TLS 1.3 is available with OpenSSL 1.1.1 or later.
   When Python has been compiled against an older version of OpenSSL, the
   flag defaults to *0*.

   .. versionadded:: 3.7

   .. deprecated:: 3.7
      The option is deprecated since OpenSSL 1.1.0. It was added to 2.7.15,
      3.6.3 and 3.7.0 for backwards compatibility with OpenSSL 1.0.2.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 857)

Unknown directive type "data".

```
.. data:: OP_NO_RENEGOTIATION

   Disable all renegotiation in TLSv1.2 and earlier. Do not send
   HelloRequest messages, and ignore renegotiation requests via ClientHello.

   This option is only available with OpenSSL 1.1.0h and later.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 866)

Unknown directive type "data".

```
.. data:: OP_CIPHER_SERVER_PREFERENCE

   Use the server's cipher ordering preference, rather than the client's.
   This option has no effect on client sockets and SSLv2 server sockets.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 873)

Unknown directive type "data".

```
.. data:: OP_SINGLE_DH_USE

   Prevents re-use of the same DH key for distinct SSL sessions.  This
   improves forward secrecy but requires more computational resources.
   This option only applies to server sockets.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 881)

Unknown directive type "data".

```
.. data:: OP_SINGLE_ECDH_USE

   Prevents re-use of the same ECDH key for distinct SSL sessions.  This
   improves forward secrecy but requires more computational resources.
   This option only applies to server sockets.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 889)

Unknown directive type "data".

```
.. data:: OP_ENABLE_MIDDLEBOX_COMPAT

   Send dummy Change Cipher Spec (CCS) messages in TLS 1.3 handshake to make
   a TLS 1.3 connection look more like a TLS 1.2 connection.

   This option is only available with OpenSSL 1.1.1 and later.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 898)

Unknown directive type "data".

```
.. data:: OP_NO_COMPRESSION

   Disable compression on the SSL channel.  This is useful if the application
   protocol supports its own compression scheme.

   .. versionadded:: 3.3
```

:class:`enum.IntFlag` collection of OP_* constants.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 907); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 909)

Unknown directive type "data".

```
.. data:: OP_NO_TICKET

   Prevent client side from requesting a session ticket.

   .. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 915)

Unknown directive type "data".

```
.. data:: OP_IGNORE_UNEXPECTED_EOF

   Ignore unexpected shutdown of TLS connections.

   This option is only available with OpenSSL 3.0.0 and later.

   .. versionadded:: 3.10
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 923)

Unknown directive type "data".

```
.. data:: HAS_ALPN

   Whether the OpenSSL library has built-in support for the *Application-Layer
   Protocol Negotiation* TLS extension as described in :rfc:`7301`.

   .. versionadded:: 3.5
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 930)

Unknown directive type "data".

```
.. data:: HAS_NEVER_CHECK_COMMON_NAME

   Whether the OpenSSL library has built-in support not checking subject
   common name and :attr:`SSLContext.hostname_checks_common_name` is
   writeable.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 938)

Unknown directive type "data".

```
.. data:: HAS_ECDH

   Whether the OpenSSL library has built-in support for the Elliptic Curve-based
   Diffie-Hellman key exchange.  This should be true unless the feature was
   explicitly disabled by the distributor.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 946)

Unknown directive type "data".

```
.. data:: HAS_SNI

   Whether the OpenSSL library has built-in support for the *Server Name
   Indication* extension (as defined in :rfc:`6066`).

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 953)

Unknown directive type "data".

```
.. data:: HAS_NPN

   Whether the OpenSSL library has built-in support for the *Next Protocol
   Negotiation* as described in the `Application Layer Protocol
   Negotiation <https://en.wikipedia.org/wiki/Application-Layer_Protocol_Negotiation>`_.
   When true, you can use the :meth:`SSLContext.set_npn_protocols` method to advertise
   which protocols you want to support.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 963)

Unknown directive type "data".

```
.. data:: HAS_SSLv2

   Whether the OpenSSL library has built-in support for the SSL 2.0 protocol.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 969)

Unknown directive type "data".

```
.. data:: HAS_SSLv3

   Whether the OpenSSL library has built-in support for the SSL 3.0 protocol.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 975)

Unknown directive type "data".

```
.. data:: HAS_TLSv1

   Whether the OpenSSL library has built-in support for the TLS 1.0 protocol.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 981)

Unknown directive type "data".

```
.. data:: HAS_TLSv1_1

   Whether the OpenSSL library has built-in support for the TLS 1.1 protocol.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 987)

Unknown directive type "data".

```
.. data:: HAS_TLSv1_2

   Whether the OpenSSL library has built-in support for the TLS 1.2 protocol.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 993)

Unknown directive type "data".

```
.. data:: HAS_TLSv1_3

   Whether the OpenSSL library has built-in support for the TLS 1.3 protocol.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 999)

Unknown directive type "data".

```
.. data:: CHANNEL_BINDING_TYPES

   List of supported TLS channel binding types.  Strings in this list
   can be used as arguments to :meth:`SSLSocket.get_channel_binding`.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1006)

Unknown directive type "data".

```
.. data:: OPENSSL_VERSION

   The version string of the OpenSSL library loaded by the interpreter::

    >>> ssl.OPENSSL_VERSION
    'OpenSSL 1.0.2k  26 Jan 2017'

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1015)

Unknown directive type "data".

```
.. data:: OPENSSL_VERSION_INFO

   A tuple of five integers representing version information about the
   OpenSSL library::

    >>> ssl.OPENSSL_VERSION_INFO
    (1, 0, 2, 11, 15)

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1025)

Unknown directive type "data".

```
.. data:: OPENSSL_VERSION_NUMBER

   The raw version number of the OpenSSL library, as a single integer::

    >>> ssl.OPENSSL_VERSION_NUMBER
    268443839
    >>> hex(ssl.OPENSSL_VERSION_NUMBER)
    '0x100020bf'

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1036)

Unknown directive type "data".

```
.. data:: ALERT_DESCRIPTION_HANDSHAKE_FAILURE
          ALERT_DESCRIPTION_INTERNAL_ERROR
          ALERT_DESCRIPTION_*

   Alert Descriptions from :rfc:`5246` and others. The `IANA TLS Alert Registry
   <https://www.iana.org/assignments/tls-parameters/tls-parameters.xml#tls-parameters-6>`_
   contains this list and references to the RFCs where their meaning is defined.

   Used as the return value of the callback function in
   :meth:`SSLContext.set_servername_callback`.

   .. versionadded:: 3.4
```

:class:`enum.IntEnum` collection of ALERT_DESCRIPTION_* constants.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1051); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1053)

Unknown directive type "versionadded".

```
.. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1055)

Unknown directive type "data".

```
.. data:: Purpose.SERVER_AUTH

   Option for :func:`create_default_context` and
   :meth:`SSLContext.load_default_certs`.  This value indicates that the
   context may be used to authenticate web servers (therefore, it will
   be used to create client-side sockets).

   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1064)

Unknown directive type "data".

```
.. data:: Purpose.CLIENT_AUTH

   Option for :func:`create_default_context` and
   :meth:`SSLContext.load_default_certs`.  This value indicates that the
   context may be used to authenticate web clients (therefore, it will
   be used to create server-side sockets).

   .. versionadded:: 3.4
```

:class:`enum.IntEnum` collection of SSL_ERROR_* constants.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1075); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1077)

Unknown directive type "versionadded".

```
.. versionadded:: 3.6
```

:class:`enum.IntEnum` collection of SSL and TLS versions for :attr:`SSLContext.maximum_version` and :attr:`SSLContext.minimum_version`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1081); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1081); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1081); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1084)

Unknown directive type "versionadded".

```
.. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1086)

Unknown directive type "attribute".

```
.. attribute:: TLSVersion.MINIMUM_SUPPORTED
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1087)

Unknown directive type "attribute".

```
.. attribute:: TLSVersion.MAXIMUM_SUPPORTED

   The minimum or maximum supported SSL or TLS version. These are magic
   constants. Their values don't reflect the lowest and highest available
   TLS/SSL versions.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1093)

Unknown directive type "attribute".

```
.. attribute:: TLSVersion.SSLv3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1094)

Unknown directive type "attribute".

```
.. attribute:: TLSVersion.TLSv1
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1095)

Unknown directive type "attribute".

```
.. attribute:: TLSVersion.TLSv1_1
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1096)

Unknown directive type "attribute".

```
.. attribute:: TLSVersion.TLSv1_2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1097)

Unknown directive type "attribute".

```
.. attribute:: TLSVersion.TLSv1_3

   SSL 3.0 to TLS 1.3.

   .. deprecated:: 3.10

      All :class:`TLSVersion` members except :attr:`TLSVersion.TLSv1_2` and
      :attr:`TLSVersion.TLSv1_3` are deprecated.
```

SSL Sockets

SSL sockets provide the following methods of :ref:`socket-objects`:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1112); *backlink*

Unknown interpreted text role "ref".

- :meth:`~socket.socket.accept()`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1114); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~socket.socket.bind()`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1115); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~socket.socket.close()`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1116); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~socket.socket.connect()`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1117); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~socket.socket.detach()`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1118); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~socket.socket.fileno()`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1119); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~socket.socket.getpeername()`, :meth:`~socket.socket.getsockname()`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1120); *backlink*

  Unknown interpreted text role "meth".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1120); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~socket.socket.getsockopt()`, :meth:`~socket.socket.setsockopt()`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1121); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1121); backlink

Unknown interpreted text role "meth".

- :meth:`~socket.socket.gettimeout()`, :meth:`~socket.socket.settimeout()`, :meth:`~socket.socket.setblocking()`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1122); backlink

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1122); backlink

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1122); backlink

Unknown interpreted text role "meth".

- :meth:`~socket.socket.listen()`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1124); backlink

Unknown interpreted text role "meth".

- :meth:`~socket.socket.makefile()`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1125); backlink

Unknown interpreted text role "meth".

- :meth:`~socket.socket.recv()`, :meth:`~socket.socket.recv_into()` (but passing a non-zero `flags` argument is not allowed)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1126); backlink

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1126); backlink

Unknown interpreted text role "meth".

- :meth:`~socket.socket.send()`, :meth:`~socket.socket.sendall()` (with the same limitation)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1128); backlink

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1128); backlink

Unknown interpreted text role "meth".

- :meth:`~socket.socket.sendfile()` (but :mod:`os.sendfile` will be used for plain-text sockets only, else :meth:`~socket.socket.send()` will be used)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1130); backlink

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1130); backlink

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1130); backlink

Unknown interpreted text role "meth".

- :meth:`~socket.socket.shutdown()`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1132); backlink

Unknown interpreted text role "meth".

However, since the SSL (and TLS) protocol has its own framing atop of TCP, the SSL sockets abstraction can, in certain respects, diverge from the specification of normal, OS-level sockets. See especially the :ref`notes on non-blocking sockets <ssl-nonblocking>`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1134); *backlink*

Unknown interpreted text role "ref".

Instances of :class:`SSLSocket` must be created using the :meth:`SSLContext.wrap_socket` method.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1139); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1139); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1142)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.5
   The :meth:`sendfile` method was added.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1145)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.5
   The :meth:`shutdown` does not reset the socket timeout each time bytes
   are received or sent. The socket timeout is now to maximum total duration
   of the shutdown.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1150)

Unknown directive type "deprecated".

```
.. deprecated:: 3.6
   It is deprecated to create a :class:`SSLSocket` instance directly, use
   :meth:`SSLContext.wrap_socket` to wrap a socket.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1154)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.7
   :class:`SSLSocket` instances must to created with
   :meth:`~SSLContext.wrap_socket`. In earlier versions, it was possible
   to create instances directly. This was never documented or officially
   supported.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1160)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.10
   Python now uses ``SSL_read_ex`` and ``SSL_write_ex`` internally. The
   functions support reading and writing of data larger than 2 GB. Writing
   zero-length data no longer fails with a protocol violation error.
```

SSL sockets also have the following additional methods and attributes:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1167)

Unknown directive type "method".

```
.. method:: SSLSocket.read(len=1024, buffer=None)

   Read up to *len* bytes of data from the SSL socket and return the result as
   a ``bytes`` instance. If *buffer* is specified, then read into the buffer
   instead, and return the number of bytes read.

   Raise :exc:`SSLWantReadError` or :exc:`SSLWantWriteError` if the socket is
   :ref:`non-blocking <ssl-nonblocking>` and the read would block.

   As at any time a re-negotiation is possible, a call to :meth:`read` can also
   cause write operations.

   .. versionchanged:: 3.5
      The socket timeout is no more reset each time bytes are received or sent.
      The socket timeout is now to maximum total duration to read up to *len*
      bytes.

   .. deprecated:: 3.6
      Use :meth:`~SSLSocket.recv` instead of :meth:`~SSLSocket.read`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1187)

Unknown directive type "method".

```
.. method:: SSLSocket.write(buf)

   Write *buf* to the SSL socket and return the number of bytes written. The
   *buf* argument must be an object supporting the buffer interface.

   Raise :exc:`SSLWantReadError` or :exc:`SSLWantWriteError` if the socket is
   :ref:`non-blocking <ssl-nonblocking>` and the write would block.
```

```
   As at any time a re-negotiation is possible, a call to :meth:`write` can
   also cause read operations.

   .. versionchanged:: 3.5
      The socket timeout is no more reset each time bytes are received or sent.
      The socket timeout is now to maximum total duration to write *buf*.

   .. deprecated:: 3.6
      Use :meth:`~SSLSocket.send` instead of :meth:`~SSLSocket.write`.
```

Note

The :meth:`~SSLSocket.read` and :meth:`~SSLSocket.write` methods are the low-level methods that read and write unencrypted, application-level data and decrypt/encrypt it to encrypted, wire-level data. These methods require an active SSL connection, i.e. the handshake was completed and :meth:`SSLSocket.unwrap` was not called.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1207); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1207); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1207); *backlink*

Unknown interpreted text role "meth".

Normally you should use the socket API methods like :meth:`~socket.socket.recv` and :meth:`~socket.socket.send` instead of these methods.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1213); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1213); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1217)

Unknown directive type "method".

```
.. method:: SSLSocket.do_handshake()

   Perform the SSL setup handshake.

   .. versionchanged:: 3.4
      The handshake method also performs :func:`match_hostname` when the
      :attr:`~SSLContext.check_hostname` attribute of the socket's
      :attr:`~SSLSocket.context` is true.

   .. versionchanged:: 3.5
      The socket timeout is no more reset each time bytes are received or sent.
      The socket timeout is now to maximum total duration of the handshake.

   .. versionchanged:: 3.7
      Hostname or IP address is matched by OpenSSL during handshake. The
      function :func:`match_hostname` is no longer used. In case OpenSSL
      refuses a hostname or IP address, the handshake is aborted early and
      a TLS alert message is send to the peer.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1236)

Unknown directive type "method".

```
.. method:: SSLSocket.getpeercert(binary_form=False)

   If there is no certificate for the peer on the other end of the connection,
   return ``None``.  If the SSL handshake hasn't been done yet, raise
   :exc:`ValueError`.

   If the ``binary_form`` parameter is :const:`False`, and a certificate was
   received from the peer, this method returns a :class:`dict` instance.  If the
   certificate was not validated, the dict is empty.  If the certificate was
   validated, it returns a dict with several keys, amongst them ``subject``
   (the principal for which the certificate was issued) and ``issuer``
   (the principal issuing the certificate).  If a certificate contains an
   instance of the *Subject Alternative Name* extension (see :rfc:`3280`),
   there will also be a ``subjectAltName`` key in the dictionary.

   The ``subject`` and ``issuer`` fields are tuples containing the sequence
   of relative distinguished names (RDNs) given in the certificate's data
   structure for the respective fields, and each RDN is a sequence of
   name-value pairs.  Here is a real-world example::

      {'issuer': ((('countryName', 'IL'),),
                  (('organizationName', 'StartCom Ltd.'),),
                  (('organizationalUnitName',
                    'Secure Digital Certificate Signing'),),
                  (('commonName',
                    'StartCom Class 2 Primary Intermediate Server CA'),)),
       'notAfter': 'Nov 22 08:15:19 2013 GMT',
       'notBefore': 'Nov 21 03:09:52 2011 GMT',
       'serialNumber': '95F0',
       'subject': ((('description', '571208-SLe257oHY9fVQ07Z'),),
                   (('countryName', 'US'),),
                   (('stateOrProvinceName', 'California'),),
```

```
                  (('localityName', 'San Francisco'),),
                  (('organizationName', 'Electronic Frontier Foundation, Inc.'),),
                  (('commonName', '*.eff.org'),),
                  (('emailAddress', 'hostmaster@eff.org'),)),
     'subjectAltName': (('DNS', '*.eff.org'), ('DNS', 'eff.org')),
     'version': 3}

.. note::

   To validate a certificate for a particular service, you can use the
   :func:`match_hostname` function.

If the ``binary_form`` parameter is :const:`True`, and a certificate was
provided, this method returns the DER-encoded form of the entire certificate
as a sequence of bytes, or :const:`None` if the peer did not provide a
certificate.  Whether the peer provides a certificate depends on the SSL
socket's role:

* for a client SSL socket, the server will always provide a certificate,
  regardless of whether validation was required;

* for a server SSL socket, the client will only provide a certificate
  when requested by the server; therefore :meth:`getpeercert` will return
  :const:`None` if you used :const:`CERT_NONE` (rather than
  :const:`CERT_OPTIONAL` or :const:`CERT_REQUIRED`).

.. versionchanged:: 3.2
   The returned dictionary includes additional items such as ``issuer``
   and ``notBefore``.

.. versionchanged:: 3.4
   :exc:`ValueError` is raised when the handshake isn't done.
   The returned dictionary includes additional X509v3 extension items
     such as ``crlDistributionPoints``, ``caIssuers`` and ``OCSP`` URIs.

.. versionchanged:: 3.9
   IPv6 address strings no longer have a trailing new line.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1306)

Unknown directive type "method".

```
.. method:: SSLSocket.cipher()

   Returns a three-value tuple containing the name of the cipher being used, the
   version of the SSL protocol that defines its use, and the number of secret
   bits being used.  If no connection has been established, returns ``None``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1312)

Unknown directive type "method".

```
.. method:: SSLSocket.shared_ciphers()

   Return the list of ciphers shared by the client during the handshake.  Each
   entry of the returned list is a three-value tuple containing the name of the
   cipher, the version of the SSL protocol that defines its use, and the number
   of secret bits the cipher uses.  :meth:`~SSLSocket.shared_ciphers` returns
   ``None`` if no connection has been established or the socket is a client
   socket.

   .. versionadded:: 3.5
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1323)

Unknown directive type "method".

```
.. method:: SSLSocket.compression()

   Return the compression algorithm being used as a string, or ``None``
   if the connection isn't compressed.

   If the higher-level protocol supports its own compression mechanism,
   you can use :data:`OP_NO_COMPRESSION` to disable SSL-level compression.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1333)

Unknown directive type "method".

```
.. method:: SSLSocket.get_channel_binding(cb_type="tls-unique")

   Get channel binding data for current connection, as a bytes object.  Returns
   ``None`` if not connected or the handshake has not been completed.

   The *cb_type* parameter allow selection of the desired channel binding
   type. Valid channel binding types are listed in the
   :data:`CHANNEL_BINDING_TYPES` list.  Currently only the 'tls-unique' channel
   binding, defined by :rfc:`5929`, is supported.  :exc:`ValueError` will be
   raised if an unsupported channel binding type is requested.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1346)

Unknown directive type "method".

```
.. method:: SSLSocket.selected_alpn_protocol()

   Return the protocol that was selected during the TLS handshake.  If
   :meth:`SSLContext.set_alpn_protocols` was not called, if the other party does
   not support ALPN, if this socket does not support any of the client's
   proposed protocols, or if the handshake has not happened yet, ``None`` is
   returned.

   .. versionadded:: 3.5
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1356)

Unknown directive type "method".

```
.. method:: SSLSocket.selected_npn_protocol()

   Return the higher-level protocol that was selected during the TLS/SSL
   handshake. If :meth:`SSLContext.set_npn_protocols` was not called, or
   if the other party does not support NPN, or if the handshake has not yet
   happened, this will return ``None``.

   .. versionadded:: 3.3

   .. deprecated:: 3.10

      NPN has been superseded by ALPN
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1369)

Unknown directive type "method".

```
.. method:: SSLSocket.unwrap()

   Performs the SSL shutdown handshake, which removes the TLS layer from the
   underlying socket, and returns the underlying socket object.  This can be
   used to go from encrypted operation over a connection to unencrypted.  The
   returned socket should always be used for further communication with the
   other side of the connection, rather than the original socket.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1377)

Unknown directive type "method".

```
.. method:: SSLSocket.verify_client_post_handshake()

   Requests post-handshake authentication (PHA) from a TLS 1.3 client. PHA
   can only be initiated for a TLS 1.3 connection from a server-side socket,
   after the initial TLS handshake and with PHA enabled on both sides, see
   :attr:`SSLContext.post_handshake_auth`.

   The method does not perform a cert exchange immediately. The server-side
   sends a CertificateRequest during the next write event and expects the
   client to respond with a certificate on the next read event.

   If any precondition isn't met (e.g. not TLS 1.3, PHA not enabled), an
   :exc:`SSLError` is raised.

   .. note::
      Only available with OpenSSL 1.1.1 and TLS 1.3 enabled. Without TLS 1.3
      support, the method raises :exc:`NotImplementedError`.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1397)

Unknown directive type "method".

```
.. method:: SSLSocket.version()

   Return the actual SSL protocol version negotiated by the connection
   as a string, or ``None`` if no secure connection is established.
   As of this writing, possible return values include ``"SSLv2"``,
   ``"SSLv3"``, ``"TLSv1"``, ``"TLSv1.1"`` and ``"TLSv1.2"``.
   Recent OpenSSL versions may define more return values.

   .. versionadded:: 3.5
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1407)

Unknown directive type "method".

```
.. method:: SSLSocket.pending()

   Returns the number of already decrypted bytes available for read, pending on
   the connection.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1412)

Unknown directive type "attribute".

```
.. attribute:: SSLSocket.context

   The :class:`SSLContext` object this SSL socket is tied to.  If the SSL
   socket was created using the deprecated :func:`wrap_socket` function
   (rather than :meth:`SSLContext.wrap_socket`), this is a custom context
   object created for this SSL socket.

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1421)

Unknown directive type "attribute".

```
.. attribute:: SSLSocket.server_side

   A boolean which is ``True`` for server-side sockets and ``False`` for
   client-side sockets.

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1428)

Unknown directive type "attribute".

```
.. attribute:: SSLSocket.server_hostname
```

```
Hostname of the server: :class:`str` type, or ``None`` for server-side
socket or if the hostname was not specified in the constructor.

.. versionadded:: 3.2

.. versionchanged:: 3.7
   The attribute is now always ASCII text. When ``server_hostname`` is
   an internationalized domain name (IDN), this attribute now stores the
   A-label form (``"xn--pythn-mua.org"``), rather than the U-label form
   (``"pythÃ¶n.org"``).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1441)

Unknown directive type "attribute".

```
.. attribute:: SSLSocket.session

   The :class:`SSLSession` for this SSL connection. The session is available
   for client and server side sockets after the TLS handshake has been
   performed. For client sockets the session can be set before
   :meth:`~SSLSocket.do_handshake` has been called to reuse a session.

   .. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1450)

Unknown directive type "attribute".

```
.. attribute:: SSLSocket.session_reused

   .. versionadded:: 3.6
```

SSL Contexts

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1458)

Unknown directive type "versionadded".

```
.. versionadded:: 3.2
```

An SSL context holds various data longer-lived than single SSL connections, such as SSL configuration options, certificate(s) and private key(s). It also manages a cache of SSL sessions for server-side sockets, in order to speed up repeated connections from the same clients.

Create a new SSL context. You may pass *protocol* which must be one of the `PROTOCOL_*` constants defined in this module. The parameter specifies which version of the SSL protocol to use. Typically, the server chooses a particular protocol version, and the client must adapt to the server's choice. Most of the versions are not interoperable with the other versions. If not specified, the default is :data:`PROTOCOL_TLS`; it provides the most compatibility with other versions.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1467); *backlink*

Unknown interpreted text role "data".

Here's a table showing which versions in a client (down the side) can connect to which versions in a server (along the top):

client / **server**	**SSLv2**	**SSLv3**	**TLS** [3]	**TLSv1**	**TLSv1.1**	**TLSv1.2**
SSLv2	yes	no	no [1]	no	no	no
SSLv3	no	yes	no [2]	no	no	no
TLS (SSLv23) [3]	no [1]	no [2]	yes	yes	yes	yes
TLSv1	no	no	yes	yes	no	no
TLSv1.1	no	no	yes	no	yes	no
TLSv1.2	no	no	yes	no	no	yes

Footnotes

[1] (*1*,*2*) :class:`SSLContext` disables SSLv2 with :data:`OP_NO_SSLv2` by default.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1493); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1493); *backlink*

Unknown interpreted text role "data".

[2] (*1*,*2*) :class:`SSLContext` disables SSLv3 with :data:`OP_NO_SSLv3` by default.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1494); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1494); *backlink*

Unknown interpreted text role "data".

[3] (*1*,*2*) TLS 1.3 protocol will be available with :data:`PROTOCOL_TLS` in OpenSSL >= 1.1.1. There is no dedicated PROTOCOL constant for just TLS 1.3.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1495);

backlink

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1499)

Unknown directive type "seealso".

```
.. seealso::
   :func:`create_default_context` lets the :mod:`ssl` module choose
   security settings for a given purpose.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1503)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6

   The context is created with secure default values. The options
   :data:`OP_NO_COMPRESSION`, :data:`OP_CIPHER_SERVER_PREFERENCE`,
   :data:`OP_SINGLE_DH_USE`, :data:`OP_SINGLE_ECDH_USE`,
   :data:`OP_NO_SSLv2` (except for :data:`PROTOCOL_SSLv2`),
   and :data:`OP_NO_SSLv3` (except for :data:`PROTOCOL_SSLv3`) are
   set by default. The initial cipher suite list contains only ``HIGH``
   ciphers, no ``NULL`` ciphers and no ``MD5`` ciphers (except for
   :data:`PROTOCOL_SSLv2`).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1514)

Unknown directive type "deprecated".

```
.. deprecated:: 3.10

   :class:`SSLContext` without protocol argument is deprecated. The
   context class will either require :data:`PROTOCOL_TLS_CLIENT` or
   :data:`PROTOCOL_TLS_SERVER` protocol in the future.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1520)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.10

   The default cipher suites now include only secure AES and ChaCha20
   ciphers with forward secrecy and security level 2. RSA and DH keys with
   less than 2048 bits and ECC keys with less than 224 bits are prohibited.
   :data:`PROTOCOL_TLS`, :data:`PROTOCOL_TLS_CLIENT`, and
   :data:`PROTOCOL_TLS_SERVER` use TLS 1.2 as minimum TLS version.
```

:class:`SSLContext` objects have the following methods and attributes:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1529); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1531)

Unknown directive type "method".

```
.. method:: SSLContext.cert_store_stats()

   Get statistics about quantities of loaded X.509 certificates, count of
   X.509 certificates flagged as CA certificates and certificate revocation
   lists as dictionary.

   Example for a context with one CA cert and one other cert::

      >>> context.cert_store_stats()
      {'crl': 0, 'x509_ca': 1, 'x509': 2}

   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1545)

Unknown directive type "method".

```
.. method:: SSLContext.load_cert_chain(certfile, keyfile=None, password=None)

   Load a private key and the corresponding certificate.  The *certfile*
   string must be the path to a single file in PEM format containing the
   certificate as well as any number of CA certificates needed to establish
   the certificate's authenticity.  The *keyfile* string, if present, must
   point to a file containing the private key.  Otherwise the private
   key will be taken from *certfile* as well.  See the discussion of
   :ref:`ssl-certificates` for more information on how the certificate
   is stored in the *certfile*.

   The *password* argument may be a function to call to get the password for
   decrypting the private key.  It will only be called if the private key is
   encrypted and a password is necessary.  It will be called with no arguments,
   and it should return a string, bytes, or bytearray.  If the return value is
   a string it will be encoded as UTF-8 before using it to decrypt the key.
   Alternatively a string, bytes, or bytearray value may be supplied directly
   as the *password* argument.  It will be ignored if the private key is not
   encrypted and no password is needed.

   If the *password* argument is not specified and a password is required,
   OpenSSL's built-in password prompting mechanism will be used to
   interactively prompt the user for a password.

   An :class:`SSLError` is raised if the private key doesn't
   match with the certificate.

   .. versionchanged:: 3.3
      New optional argument *password*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1575)

Unknown directive type "method".

```
.. method:: SSLContext.load_default_certs(purpose=Purpose.SERVER_AUTH)

   Load a set of default "certification authority" (CA) certificates from
   default locations. On Windows it loads CA certs from the ``CA`` and
   ``ROOT`` system stores. On all systems it calls
   :meth:`SSLContext.set_default_verify_paths`. In the future the method may
   load CA certificates from other locations, too.

   The *purpose* flag specifies what kind of CA certificates are loaded. The
   default settings :data:`Purpose.SERVER_AUTH` loads certificates, that are
   flagged and trusted for TLS web server authentication (client side
   sockets). :data:`Purpose.CLIENT_AUTH` loads CA certificates for client
   certificate verification on the server side.

   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1591)

Unknown directive type "method".

```
.. method:: SSLContext.load_verify_locations(cafile=None, capath=None, cadata=None)

   Load a set of "certification authority" (CA) certificates used to validate
   other peers' certificates when :data:`verify_mode` is other than
   :data:`CERT_NONE`.  At least one of *cafile* or *capath* must be specified.

   This method can also load certification revocation lists (CRLs) in PEM or
   DER format. In order to make use of CRLs, :attr:`SSLContext.verify_flags`
   must be configured properly.

   The *cafile* string, if present, is the path to a file of concatenated
   CA certificates in PEM format. See the discussion of
   :ref:`ssl-certificates` for more information about how to arrange the
   certificates in this file.

   The *capath* string, if present, is
   the path to a directory containing several CA certificates in PEM format,
   following an `OpenSSL specific layout
   <https://www.openssl.org/docs/manmaster/man3/SSL_CTX_load_verify_locations.html>`_.

   The *cadata* object, if present, is either an ASCII string of one or more
   PEM-encoded certificates or a :term:`bytes-like object` of DER-encoded
   certificates. Like with *capath* extra lines around PEM-encoded
   certificates are ignored but at least one certificate must be present.

   .. versionchanged:: 3.4
      New optional argument *cadata*
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1619)

Unknown directive type "method".

```
.. method:: SSLContext.get_ca_certs(binary_form=False)

   Get a list of loaded "certification authority" (CA) certificates. If the
   ``binary_form`` parameter is :const:`False` each list
   entry is a dict like the output of :meth:`SSLSocket.getpeercert`. Otherwise
   the method returns a list of DER-encoded certificates. The returned list
   does not contain certificates from *capath* unless a certificate was
   requested and loaded by a SSL connection.

   .. note::
      Certificates in a capath directory aren't loaded unless they have
      been used at least once.

   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1634)

Unknown directive type "method".

```
.. method:: SSLContext.get_ciphers()

   Get a list of enabled ciphers. The list is in order of cipher priority.
   See :meth:`SSLContext.set_ciphers`.

   Example::

       >>> ctx = ssl.SSLContext(ssl.PROTOCOL_SSLv23)
       >>> ctx.set_ciphers('ECDHE+AESGCM:!ECDSA')
       >>> ctx.get_ciphers()
       [{'aead': True,
         'alg_bits': 256,
         'auth': 'auth-rsa',
         'description': 'ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH     Au=RSA  '
                        'Enc=AESGCM(256) Mac=AEAD',
         'digest': None,
         'id': 50380848,
         'kea': 'kx-ecdhe',
         'name': 'ECDHE-RSA-AES256-GCM-SHA384',
         'protocol': 'TLSv1.2',
         'strength_bits': 256,
         'symmetric': 'aes-256-gcm'},
        {'aead': True,
         'alg_bits': 128,
         'auth': 'auth-rsa',
         'description': 'ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  '
                        'Enc=AESGCM(128) Mac=AEAD',
         'digest': None,
         'id': 50380847,
         'kea': 'kx-ecdhe',
         'name': 'ECDHE-RSA-AES128-GCM-SHA256',
         'protocol': 'TLSv1.2',
         'strength_bits': 128,
         'symmetric': 'aes-128-gcm'}]

   .. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1671)

Unknown directive type "method".

```
.. method:: SSLContext.set_default_verify_paths()

   Load a set of default "certification authority" (CA) certificates from
   a filesystem path defined when building the OpenSSL library.  Unfortunately,
   there's no easy way to know whether this method succeeds: no error is
   returned if no certificates are to be found.  When the OpenSSL library is
   provided as part of the operating system, though, it is likely to be
   configured properly.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1680)

Unknown directive type "method".

```
.. method:: SSLContext.set_ciphers(ciphers)

   Set the available ciphers for sockets created with this context.
   It should be a string in the `OpenSSL cipher list format
   <https://www.openssl.org/docs/manmaster/man1/ciphers.html>`_.
   If no cipher can be selected (because compile-time options or other
   configuration forbids use of all the specified ciphers), an
   :class:`SSLError` will be raised.

   .. note::
      when connected, the :meth:`SSLSocket.cipher` method of SSL sockets will
      give the currently selected cipher.

      TLS 1.3 cipher suites cannot be disabled with
      :meth:`~SSLContext.set_ciphers`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1696)

Unknown directive type "method".

```
.. method:: SSLContext.set_alpn_protocols(protocols)

   Specify which protocols the socket should advertise during the SSL/TLS
   handshake. It should be a list of ASCII strings, like ``['http/1.1',
   'spdy/2']``, ordered by preference. The selection of a protocol will happen
   during the handshake, and will play out according to :rfc:`7301`. After a
   successful handshake, the :meth:`SSLSocket.selected_alpn_protocol` method will
   return the agreed-upon protocol.

   This method will raise :exc:`NotImplementedError` if :data:`HAS_ALPN` is
   ``False``.

   .. versionadded:: 3.5
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1710)

Unknown directive type "method".

```
.. method:: SSLContext.set_npn_protocols(protocols)

   Specify which protocols the socket should advertise during the SSL/TLS
   handshake. It should be a list of strings, like ``['http/1.1', 'spdy/2']``,
   ordered by preference. The selection of a protocol will happen during the
   handshake, and will play out according to the `Application Layer Protocol Negotiation
   <https://en.wikipedia.org/wiki/Application-Layer_Protocol_Negotiation>`_. After a
   successful handshake, the :meth:`SSLSocket.selected_npn_protocol` method will
   return the agreed-upon protocol.

   This method will raise :exc:`NotImplementedError` if :data:`HAS_NPN` is
   ``False``.

   .. versionadded:: 3.3

   .. deprecated:: 3.10

      NPN has been superseded by ALPN
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1729)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.sni_callback

   Register a callback function that will be called after the TLS Client Hello
   handshake message has been received by the SSL/TLS server when the TLS client
   specifies a server name indication. The server name indication mechanism
   is specified in :rfc:`6066` section 3 - Server Name Indication.

   Only one callback can be set per ``SSLContext``.  If *sni_callback*
   is set to ``None`` then the callback is disabled. Calling this function a
   subsequent time will disable the previously registered callback.

   The callback function will be called with three
   arguments; the first being the :class:`ssl.SSLSocket`, the second is a string
   that represents the server name that the client is intending to communicate
   (or :const:`None` if the TLS Client Hello does not contain a server name)
   and the third argument is the original :class:`SSLContext`. The server name
   argument is text. For internationalized domain name, the server
   name is an IDN A-label (``"xn--pythn-mua.org"``).

   A typical use of this callback is to change the :class:`ssl.SSLSocket`'s
   :attr:`SSLSocket.context` attribute to a new object of type
   :class:`SSLContext` representing a certificate chain that matches the server
   name.

   Due to the early negotiation phase of the TLS connection, only limited
   methods and attributes are usable like
   :meth:`SSLSocket.selected_alpn_protocol` and :attr:`SSLSocket.context`.
   The :meth:`SSLSocket.getpeercert`,
   :meth:`SSLSocket.cipher` and :meth:`SSLSocket.compression` methods require that
   the TLS connection has progressed beyond the TLS Client Hello and therefore
   will not return meaningful values nor can they be called safely.

   The *sni_callback* function must return ``None`` to allow the
   TLS negotiation to continue.  If a TLS failure is required, a constant
```

```
:const:`ALERT_DESCRIPTION_* <ALERT_DESCRIPTION_INTERNAL_ERROR>` can be
returned.  Other return values will result in a TLS fatal error with
:const:`ALERT_DESCRIPTION_INTERNAL_ERROR`.

If an exception is raised from the *sni_callback* function the TLS
connection will terminate with a fatal TLS alert message
:const:`ALERT_DESCRIPTION_HANDSHAKE_FAILURE`.

This method will raise :exc:`NotImplementedError` if the OpenSSL library
had OPENSSL_NO_TLSEXT defined when it was built.

.. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 1776)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.set_servername_callback(server_name_callback)

   This is a legacy API retained for backwards compatibility. When possible,
   you should use :attr:`sni_callback` instead. The given *server_name_callback*
   is similar to *sni_callback*, except that when the server hostname is an
   IDN-encoded internationalized domain name, the *server_name_callback*
   receives a decoded U-label (``"pythÃ¶n.org"``).

   If there is an decoding error on the server name, the TLS connection will
   terminate with an :const:`ALERT_DESCRIPTION_INTERNAL_ERROR` fatal TLS
   alert message to the client.

   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 1790)

Unknown directive type "method".

```
.. method:: SSLContext.load_dh_params(dhfile)

   Load the key generation parameters for Diffie-Hellman (DH) key exchange.
   Using DH key exchange improves forward secrecy at the expense of
   computational resources (both on the server and on the client).
   The *dhfile* parameter should be the path to a file containing DH
   parameters in PEM format.

   This setting doesn't apply to client sockets.  You can also use the
   :data:`OP_SINGLE_DH_USE` option to further improve security.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 1803)

Unknown directive type "method".

```
.. method:: SSLContext.set_ecdh_curve(curve_name)

   Set the curve name for Elliptic Curve-based Diffie-Hellman (ECDH) key
   exchange.  ECDH is significantly faster than regular DH while arguably
   as secure.  The *curve_name* parameter should be a string describing
   a well-known elliptic curve, for example ``prime256v1`` for a widely
   supported curve.

   This setting doesn't apply to client sockets.  You can also use the
   :data:`OP_SINGLE_ECDH_USE` option to further improve security.

   This method is not available if :data:`HAS_ECDH` is ``False``.

   .. versionadded:: 3.3

   .. seealso::
      `SSL/TLS & Perfect Forward Secrecy <https://vincent.bernat.im/en/blog/2011-ssl-perfect-forward-secrecy>`_
        Vincent Bernat.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 1822)

Unknown directive type "method".

```
.. method:: SSLContext.wrap_socket(sock, server_side=False, \
      do_handshake_on_connect=True, suppress_ragged_eofs=True, \
      server_hostname=None, session=None)

   Wrap an existing Python socket *sock* and return an instance of
   :attr:`SSLContext.sslsocket_class` (default :class:`SSLSocket`). The
   returned SSL socket is tied to the context, its settings and certificates.
   *sock* must be a :data:`~socket.SOCK_STREAM` socket; other
   socket types are unsupported.

   The parameter ``server_side`` is a boolean which identifies whether
   server-side or client-side behavior is desired from this socket.

   For client-side sockets, the context construction is lazy; if the
   underlying socket isn't connected yet, the context construction will be
   performed after :meth:`connect` is called on the socket.  For
   server-side sockets, if the socket has no remote peer, it is assumed
   to be a listening socket, and the server-side SSL wrapping is
   automatically performed on client connections accepted via the
   :meth:`accept` method. The method may raise :exc:`SSLError`.

   On client connections, the optional parameter *server_hostname* specifies
   the hostname of the service which we are connecting to.  This allows a
   single server to host multiple SSL-based services with distinct certificates,
   quite similarly to HTTP virtual hosts. Specifying *server_hostname* will
   raise a :exc:`ValueError` if *server_side* is true.

   The parameter ``do_handshake_on_connect`` specifies whether to do the SSL
   handshake automatically after doing a :meth:`socket.connect`, or whether the
   application program will call it explicitly, by invoking the
   :meth:`SSLSocket.do_handshake` method.  Calling
   :meth:`SSLSocket.do_handshake` explicitly gives the program control over the
   blocking behavior of the socket I/O involved in the handshake.

   The parameter ``suppress_ragged_eofs`` specifies how the
   :meth:`SSLSocket.recv` method should signal unexpected EOF from the other end
   of the connection.  If specified as :const:`True` (the default), it returns a
   normal EOF (an empty bytes object) in response to unexpected EOF errors
   raised from the underlying socket; if :const:`False`, it will raise the
```

```
   exceptions back to the caller.

   *session*, see :attr:`~SSLSocket.session`.

   .. versionchanged:: 3.5
      Always allow a server_hostname to be passed, even if OpenSSL does not
      have SNI.

   .. versionchanged:: 3.6
      *session* argument was added.

    .. versionchanged:: 3.7
      The method returns an instance of :attr:`SSLContext.sslsocket_class`
      instead of hard-coded :class:`SSLSocket`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 1876)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.sslsocket_class

   The return type of :meth:`SSLContext.wrap_socket`, defaults to
   :class:`SSLSocket`. The attribute can be overridden on instance of class
   in order to return a custom subclass of :class:`SSLSocket`.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 1884)

Unknown directive type "method".

```
.. method:: SSLContext.wrap_bio(incoming, outgoing, server_side=False, \
                                server_hostname=None, session=None)

   Wrap the BIO objects *incoming* and *outgoing* and return an instance of
   :attr:`SSLContext.sslobject_class` (default :class:`SSLObject`). The SSL
   routines will read input data from the incoming BIO and write data to the
   outgoing BIO.

   The *server_side*, *server_hostname* and *session* parameters have the
   same meaning as in :meth:`SSLContext.wrap_socket`.

   .. versionchanged:: 3.6
      *session* argument was added.

   .. versionchanged:: 3.7
      The method returns an instance of :attr:`SSLContext.sslobject_class`
      instead of hard-coded :class:`SSLObject`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 1902)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.sslobject_class

   The return type of :meth:`SSLContext.wrap_bio`, defaults to
   :class:`SSLObject`. The attribute can be overridden on instance of class
   in order to return a custom subclass of :class:`SSLObject`.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 1910)

Unknown directive type "method".

```
.. method:: SSLContext.session_stats()

   Get statistics about the SSL sessions created or managed by this context.
   A dictionary is returned which maps the names of each `piece of information <https://www.openssl.org/docs/man1.1.1/ssl/SSL_C
   numeric values.  For example, here is the total number of hits and misses
   in the session cache since the context was created::

      >>> stats = context.session_stats()
      >>> stats['hits'], stats['misses']
      (0, 0)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 1921)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.check_hostname

   Whether to match the peer cert's hostname in
   :meth:`SSLSocket.do_handshake`. The context's
   :attr:`~SSLContext.verify_mode` must be set to :data:`CERT_OPTIONAL` or
   :data:`CERT_REQUIRED`, and you must pass *server_hostname* to
   :meth:`~SSLContext.wrap_socket` in order to match the hostname.  Enabling
   hostname checking automatically sets :attr:`~SSLContext.verify_mode` from
   :data:`CERT_NONE` to :data:`CERT_REQUIRED`.  It cannot be set back to
   :data:`CERT_NONE` as long as hostname checking is enabled. The
   :data:`PROTOCOL_TLS_CLIENT` protocol enables hostname checking by default.
   With other protocols, hostname checking must be enabled explicitly.

   Example::

      import socket, ssl

      context = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2)
      context.verify_mode = ssl.CERT_REQUIRED
      context.check_hostname = True
      context.load_default_certs()

      s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
      ssl_sock = context.wrap_socket(s, server_hostname='www.verisign.com')
      ssl_sock.connect(('www.verisign.com', 443))

   .. versionadded:: 3.4

   .. versionchanged:: 3.7

      :attr:`~SSLContext.verify_mode` is now automatically changed
```

```
      to :data:`CERT_REQUIRED`  when hostname checking is enabled and
      :attr:`~SSLContext.verify_mode` is :data:`CERT_NONE`. Previously
      the same operation would have failed with a :exc:`ValueError`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1956)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.keylog_filename

   Write TLS keys to a keylog file, whenever key material is generated or
   received. The keylog file is designed for debugging purposes only. The
   file format is specified by NSS and used by many traffic analyzers such
   as Wireshark. The log file is opened in append-only mode. Writes are
   synchronized between threads, but not between processes.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1966)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.maximum_version

   A :class:`TLSVersion` enum member representing the highest supported
   TLS version. The value defaults to :attr:`TLSVersion.MAXIMUM_SUPPORTED`.
   The attribute is read-only for protocols other than :attr:`PROTOCOL_TLS`,
   :attr:`PROTOCOL_TLS_CLIENT`, and :attr:`PROTOCOL_TLS_SERVER`.

   The attributes :attr:`~SSLContext.maximum_version`,
   :attr:`~SSLContext.minimum_version` and
   :attr:`SSLContext.options` all affect the supported SSL
   and TLS versions of the context. The implementation does not prevent
   invalid combination. For example a context with
   :attr:`OP_NO_TLSv1_2` in :attr:`~SSLContext.options` and
   :attr:`~SSLContext.maximum_version` set to :attr:`TLSVersion.TLSv1_2`
   will not be able to establish a TLS 1.2 connection.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1984)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.minimum_version

   Like :attr:`SSLContext.maximum_version` except it is the lowest
   supported version or :attr:`TLSVersion.MINIMUM_SUPPORTED`.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1991)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.num_tickets

   Control the number of TLS 1.3 session tickets of a
   :attr:`PROTOCOL_TLS_SERVER` context. The setting has no impact on TLS
   1.0 to 1.2 connections.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 1999)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.options

   An integer representing the set of SSL options enabled on this context.
   The default value is :data:`OP_ALL`, but you can specify other options
   such as :data:`OP_NO_SSLv2` by ORing them together.

   .. versionchanged:: 3.6
      :attr:`SSLContext.options` returns :class:`Options` flags:

         >>> ssl.create_default_context().options  # doctest: +SKIP
         <Options.OP_ALL|OP_NO_SSLv3|OP_NO_SSLv2|OP_NO_COMPRESSION: 2197947391>

   .. deprecated:: 3.7

      All ``OP_NO_SSL*`` and ``OP_NO_TLS*`` options have been deprecated since
      Python 3.7. Use :attr:`SSLContext.minimum_version` and
      :attr:`SSLContext.maximum_version` instead.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2017)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.post_handshake_auth

   Enable TLS 1.3 post-handshake client authentication. Post-handshake auth
   is disabled by default and a server can only request a TLS client
   certificate during the initial handshake. When enabled, a server may
   request a TLS client certificate at any time after the handshake.

   When enabled on client-side sockets, the client signals the server that
   it supports post-handshake authentication.

   When enabled on server-side sockets, :attr:`SSLContext.verify_mode` must
   be set to :data:`CERT_OPTIONAL` or :data:`CERT_REQUIRED`, too. The
   actual client cert exchange is delayed until
   :meth:`SSLSocket.verify_client_post_handshake` is called and some I/O is
   performed.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2035)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.protocol

   The protocol version chosen when constructing the context.  This attribute
   is read-only.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2040)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.hostname_checks_common_name

   Whether :attr:`~SSLContext.check_hostname` falls back to verify the cert's
   subject common name in the absence of a subject alternative name
   extension (default: true).

   .. versionadded:: 3.7

   .. versionchanged:: 3.10

      The flag had no effect with OpenSSL before version 1.1.1k. Python 3.8.9,
      3.9.3, and 3.10 include workarounds for previous versions.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2053)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.security_level

   An integer representing the `security level
   <https://www.openssl.org/docs/manmaster/man3/SSL_CTX_get_security_level.html>`_
   for the context. This attribute is read-only.

   .. versionadded:: 3.10
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2061)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.verify_flags

   The flags for certificate verification operations. You can set flags like
   :data:`VERIFY_CRL_CHECK_LEAF` by ORing them together. By default OpenSSL
   does neither require nor verify certificate revocation lists (CRLs).

   .. versionadded:: 3.4

   .. versionchanged:: 3.6
      :attr:`SSLContext.verify_flags` returns :class:`VerifyFlags` flags:

          >>> ssl.create_default_context().verify_flags  # doctest: +SKIP
          <VerifyFlags.VERIFY_X509_TRUSTED_FIRST: 32768>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2075)

Unknown directive type "attribute".

```
.. attribute:: SSLContext.verify_mode

   Whether to try to verify other peers' certificates and how to behave
   if verification fails.  This attribute must be one of
   :data:`CERT_NONE`, :data:`CERT_OPTIONAL` or :data:`CERT_REQUIRED`.

   .. versionchanged:: 3.6
      :attr:`SSLContext.verify_mode` returns :class:`VerifyMode` enum:

          >>> ssl.create_default_context().verify_mode  # doctest: +SKIP
          <VerifyMode.CERT_REQUIRED: 2>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2087)

Unknown directive type "index".

```
.. index:: single: certificates
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2089)

Unknown directive type "index".

```
.. index:: single: X509 certificate
```

Certificates

Certificates in general are part of a public-key / private-key system. In this system, each *principal*, (which may be a machine, or a person, or an organization) is assigned a unique two-part encryption key. One part of the key is public, and is called the *public key*; the other part is kept secret, and is called the *private key*. The two parts are related, in that if you encrypt a message with one of the parts, you can decrypt it with the other part, and **only** with the other part.

A certificate contains information about two principals. It contains the name of a *subject*, and the subject's public key. It also contains a statement by a second principal, the *issuer*, that the subject is who they claim to be, and that this is indeed the subject's public key. The issuer's statement is signed with the issuer's private key, which only the issuer knows. However, anyone can verify the issuer's statement by finding the issuer's public key, decrypting the statement with it, and comparing it to the other information in the certificate. The certificate also contains information about the time period over which it is valid. This is expressed as two fields, called "notBefore" and "notAfter".

In the Python use of certificates, a client or server can use a certificate to prove who they are. The other side of a network connection can also be required to produce a certificate, and that certificate can be validated to the satisfaction of the client or server that requires such validation. The connection attempt can be set to raise an exception if the validation fails. Validation is done automatically, by the underlying OpenSSL framework; the application need not concern itself with its mechanics. But the application

does usually need to provide sets of certificates to allow this process to take place.

Python uses files to contain certificates. They should be formatted as "PEM" (see RFC 1422), which is a base-64 encoded form wrapped with a header line and a footer line:

```
-----BEGIN CERTIFICATE-----
... (certificate in base64 PEM encoding) ...
-----END CERTIFICATE-----
```

Certificate chains

The Python files which contain certificates can contain a sequence of certificates, sometimes called a *certificate chain*. This chain should start with the specific certificate for the principal who "is" the client or server, and then the certificate for the issuer of that certificate, and then the certificate for the issuer of *that* certificate, and so on up the chain till you get to a certificate which is *self-signed*, that is, a certificate which has the same subject and issuer, sometimes called a *root certificate*. The certificates should just be concatenated together in the certificate file. For example, suppose we had a three certificate chain, from our server certificate to the certificate of the certification authority that signed our server certificate, to the root certificate of the agency which issued the certification authority's certificate:

```
-----BEGIN CERTIFICATE-----
... (certificate for your server)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the certificate for the CA)...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
... (the root certificate for the CA's issuer)...
-----END CERTIFICATE-----
```

CA certificates

If you are going to require validation of the other side of the connection's certificate, you need to provide a "CA certs" file, filled with the certificate chains for each issuer you are willing to trust. Again, this file just contains these chains concatenated together. For validation, Python will use the first chain it finds in the file which matches. The platform's certificates file can be used by calling :meth:`SSLContext.load_default_certs`, this is done automatically with :func:`.create_default_context`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2161); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2161); *backlink*

Unknown interpreted text role "func".

Combined key and certificate

Often the private key is stored in the same file as the certificate; in this case, only the `certfile` parameter to :meth:`SSLContext.load_cert_chain` and :func:`wrap_socket` needs to be passed. If the private key is stored with the certificate, it should come before the first certificate in the certificate chain:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2172); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2172); *backlink*

Unknown interpreted text role "func".

```
-----BEGIN RSA PRIVATE KEY-----
... (private key in base64 encoding) ...
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
... (certificate in base64 PEM encoding) ...
-----END CERTIFICATE-----
```

Self-signed certificates

If you are going to create a server that provides SSL-encrypted connection services, you will need to acquire a certificate for that service. There are many ways of acquiring appropriate certificates, such as buying one from a certification authority. Another common practice is to generate a self-signed certificate. The simplest way to do this is with the OpenSSL package, using something like the following:

```
% openssl req -new -x509 -days 365 -nodes -out cert.pem -keyout cert.pem
Generating a 1024 bit RSA private key
.......++++++
..............................++++++
writing new private key to 'cert.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:MyState
Locality Name (eg, city) []:Some City
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Organization, Inc.
Organizational Unit Name (eg, section) []:My Group
Common Name (eg, YOUR name) []:myserver.mygroup.myorganization.com
Email Address []:ops@myserver.mygroup.myorganization.com
%
```

The disadvantage of a self-signed certificate is that it is its own root certificate, and no one else will have it in their cache of known (and trusted) root certificates.

Examples

Testing for SSL support

To test for the presence of SSL support in a Python installation, user code should use the following idiom:

```
try:
    import ssl
except ImportError:
    pass
else:
    ...  # do something that requires SSL support
```

Client-side operation

This example creates a SSL context with the recommended security settings for client sockets, including automatic certificate verification:

```
>>> context = ssl.create_default_context()
```

If you prefer to tune security settings yourself, you might create a context from scratch (but beware that you might not get the settings right):

```
>>> context = ssl.SSLContext(ssl.PROTOCOL_TLS_CLIENT)
>>> context.load_verify_locations("/etc/ssl/certs/ca-bundle.crt")
```

(this snippet assumes your operating system places a bundle of all CA certificates in `/etc/ssl/certs/ca-bundle.crt`; if not, you'll get an error and have to adjust the location)

The :data:`PROTOCOL_TLS_CLIENT` protocol configures the context for cert validation and hostname verification. :attr:`~SSLContext.verify_mode` is set to :data:`CERT_REQUIRED` and :attr:`~SSLContext.check_hostname` is set to `True`. All other protocols create SSL contexts with insecure defaults.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2257); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2257); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2257); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2257); *backlink*

Unknown interpreted text role "attr".

When you use the context to connect to a server, :const:`CERT_REQUIRED` and :attr:`~SSLContext.check_hostname` validate the server certificate: it ensures that the server certificate was signed with one of the CA certificates, checks the signature for correctness, and verifies other properties like validity and identity of the hostname:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2262); *backlink*

Unknown interpreted text role "const".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2262); *backlink*

Unknown interpreted text role "attr".

```
>>> conn = context.wrap_socket(socket.socket(socket.AF_INET),
...                            server_hostname="www.python.org")
>>> conn.connect(("www.python.org", 443))
```

You may then fetch the certificate:

```
>>> cert = conn.getpeercert()
```

Visual inspection shows that the certificate does identify the desired service (that is, the HTTPS host `www.python.org`):

```
>>> pprint.pprint(cert)
{'OCSP': ('http://ocsp.digicert.com',),
 'caIssuers': ('http://cacerts.digicert.com/DigiCertSHA2ExtendedValidationServerCA.crt',),
 'crlDistributionPoints': ('http://crl3.digicert.com/sha2-ev-server-g1.crl',
                           'http://crl4.digicert.com/sha2-ev-server-g1.crl'),
 'issuer': ((('countryName', 'US'),),
            (('organizationName', 'DigiCert Inc'),),
            (('organizationalUnitName', 'www.digicert.com'),),
            (('commonName', 'DigiCert SHA2 Extended Validation Server CA'),)),
 'notAfter': 'Sep  9 12:00:00 2016 GMT',
 'notBefore': 'Sep  5 00:00:00 2014 GMT',
 'serialNumber': '01BB6F00122B177F36CAB49CEA8B6B26',
 'subject': ((('businessCategory', 'Private Organization'),),
             (('1.3.6.1.4.1.311.60.2.1.3', 'US'),),
             (('1.3.6.1.4.1.311.60.2.1.2', 'Delaware'),),
             (('serialNumber', '3359300'),),
             (('streetAddress', '16 Allen Rd'),),
             (('postalCode', '03894-4801'),),
             (('countryName', 'US'),),
             (('stateOrProvinceName', 'NH'),),
             (('localityName', 'Wolfeboro'),),
             (('organizationName', 'Python Software Foundation'),),
             (('commonName', 'www.python.org'),)),
 'subjectAltName': (('DNS', 'www.python.org'),
                    ('DNS', 'python.org'),
                    ('DNS', 'pypi.org'),
                    ('DNS', 'docs.python.org'),
                    ('DNS', 'testpypi.org'),
                    ('DNS', 'bugs.python.org'),
                    ('DNS', 'wiki.python.org'),
                    ('DNS', 'hg.python.org'),
                    ('DNS', 'mail.python.org'),
                    ('DNS', 'packaging.python.org'),
                    ('DNS', 'pythonhosted.org'),
                    ('DNS', 'www.pythonhosted.org'),
                    ('DNS', 'test.pythonhosted.org'),
                    ('DNS', 'us.pycon.org'),
                    ('DNS', 'id.python.org')),
 'version': 3}
```

Now the SSL channel is established and the certificate verified, you can proceed to talk with the server:

```
>>> conn.sendall(b"HEAD / HTTP/1.0\r\nHost: linuxfr.org\r\n\r\n")
>>> pprint.pprint(conn.recv(1024).split(b"\r\n"))
[b'HTTP/1.1 200 OK',
 b'Date: Sat, 18 Oct 2014 18:27:20 GMT',
 b'Server: nginx',
 b'Content-Type: text/html; charset=utf-8',
 b'X-Frame-Options: SAMEORIGIN',
 b'Content-Length: 45679',
 b'Accept-Ranges: bytes',
 b'Via: 1.1 varnish',
```

```
b'Age: 2188',
b'X-Served-By: cache-lcy1134-LCY',
b'X-Cache: HIT',
b'X-Cache-Hits: 11',
b'Vary: Cookie',
b'Strict-Transport-Security: max-age=63072000; includeSubDomains',
b'Connection: close',
b'',
b'']
```

See the discussion of :ref:`ssl-security` below.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 2342); *backlink*

Unknown interpreted text role "ref".

Server-side operation

For server operation, typically you'll need to have a server certificate, and private key, each in a file. You'll first create a context holding the key and the certificate, so that clients can check your authenticity. Then you'll open a socket, bind it to a port, call :meth:`listen` on it, and start waiting for clients to connect:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 2348); *backlink*

Unknown interpreted text role "meth".

```
import socket, ssl

context = ssl.create_default_context(ssl.Purpose.CLIENT_AUTH)
context.load_cert_chain(certfile="mycertfile", keyfile="mykeyfile")

bindsocket = socket.socket()
bindsocket.bind(('myaddr.mydomain.com', 10023))
bindsocket.listen(5)
```

When a client connects, you'll call :meth:`accept` on the socket to get the new socket from the other end, and use the context's :meth:`SSLContext.wrap_socket` method to create a server-side SSL socket for the connection:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 2363); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 2363); *backlink*

Unknown interpreted text role "meth".

```
while True:
    newsocket, fromaddr = bindsocket.accept()
    connstream = context.wrap_socket(newsocket, server_side=True)
    try:
        deal_with_client(connstream)
    finally:
        connstream.shutdown(socket.SHUT_RDWR)
        connstream.close()
```

Then you'll read data from the `connstream` and do something with it till you are finished with the client (or the client is finished with you):

```
def deal_with_client(connstream):
    data = connstream.recv(1024)
    # empty data means the client is finished with us
    while data:
        if not do_something(connstream, data):
            # we'll assume do_something returns False
            # when we're finished with client
            break
        data = connstream.recv(1024)
    # finished with client
```

And go back to listening for new client connections (of course, a real server would probably handle each client connection in a separate thread, or put the sockets in :ref:`non-blocking mode <ssl-nonblocking>` and use an event loop).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 2390); *backlink*

Unknown interpreted text role "ref".

Notes on non-blocking sockets

SSL sockets behave slightly different than regular sockets in non-blocking mode. When working with non-blocking sockets, there are thus several things you need to be aware of:

- Most :class:`SSLSocket` methods will raise either :exc:`SSLWantWriteError` or :exc:`SSLWantReadError` instead of :exc:`BlockingIOError` if an I/O operation would block. :exc:`SSLWantReadError` will be raised if a read operation on the underlying socket is necessary, and :exc:`SSLWantWriteError` for a write operation on the underlying socket. Note that attempts to *write* to an SSL socket may require *reading* from the underlying socket first, and attempts to *read* from the SSL socket may require a prior *write* to the underlying socket.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 2404); *backlink*

  Unknown interpreted text role "class".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 2404); *backlink*

  Unknown interpreted text role "exc".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)ssl.rst, line 2404); *backlink*

  Unknown interpreted text role "exc".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2404); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2404); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2404); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2414)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.5

   In earlier Python versions, the :meth:`!SSLSocket.send` method
   returned zero instead of raising :exc:`SSLWantWriteError` or
   :exc:`SSLWantReadError`.
```

- Calling :func:`~select.select` tells you that the OS-level socket can be read from (or written to), but it does not imply that there is sufficient data at the upper SSL layer. For example, only part of an SSL frame might have arrived. Therefore, you must be ready to handle :meth:`SSLSocket.recv` and :meth:`SSLSocket.send` failures, and retry after another call to :func:`~select.select`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2420); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2420); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2420); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2420); *backlink*

Unknown interpreted text role "func".

- Conversely, since the SSL layer has its own framing, a SSL socket may still have data available for reading without :func:`~select.select` being aware of it. Therefore, you should first call :meth:`SSLSocket.recv` to drain any potentially available data, and then only block on a :func:`~select.select` call if still necessary.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2427); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2427); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2427); *backlink*

Unknown interpreted text role "func".

(of course, similar provisions apply when using other primitives such as :func:`~select.poll`, or those in the :mod:`selectors` module)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2433); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2433); *backlink*

Unknown interpreted text role "mod".

- The SSL handshake itself will be non-blocking: the :meth:`SSLSocket.do_handshake` method has to be retried until it returns successfully. Here is a synopsis using :func:`~select.select` to wait for the socket's readiness:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2436); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2436); *backlink*

Unknown interpreted text role "func".

```
while True:
    try:
        sock.do_handshake()
        break
    except ssl.SSLWantReadError:
        select.select([sock], [], [])
    except ssl.SSLWantWriteError:
        select.select([], [sock], [])
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2450)

Unknown directive type "seealso".

```
.. seealso::

   The :mod:`asyncio` module supports :ref:`non-blocking SSL sockets
   <ssl-nonblocking>` and provides a
   higher level API. It polls for events using the :mod:`selectors` module and
   handles :exc:`SSLWantWriteError`, :exc:`SSLWantReadError` and
   :exc:`BlockingIOError` exceptions. It runs the SSL handshake asynchronously
   as well.
```

Memory BIO Support

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2463)

Unknown directive type "versionadded".

```
.. versionadded:: 3.5
```

Ever since the SSL module was introduced in Python 2.6, the :class:`SSLSocket` class has provided two related but distinct areas of functionality:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2465); *backlink*

Unknown interpreted text role "class".

- SSL protocol handling
- Network IO

The network IO API is identical to that provided by :class:`socket.socket`, from which :class:`SSLSocket` also inherits. This allows an SSL socket to be used as a drop-in replacement for a regular socket, making it very easy to add SSL support to an existing application.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2471); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2471); *backlink*

Unknown interpreted text role "class".

Combining SSL protocol handling and network IO usually works well, but there are some cases where it doesn't. An example is async IO frameworks that want to use a different IO multiplexing model than the "select/poll on a file descriptor" (readiness based) model that is assumed by :class:`socket.socket` and by the internal OpenSSL socket IO routines. This is mostly relevant for platforms like Windows where this model is not efficient. For this purpose, a reduced scope variant of :class:`SSLSocket` called :class:`SSLObject` is provided.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2476); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2476); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2476); *backlink*

Unknown interpreted text role "class".

A reduced-scope variant of :class:`SSLSocket` representing an SSL protocol instance that does not contain any network IO methods. This class is typically used by framework authors that want to implement asynchronous IO for SSL through memory buffers.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2487); *backlink*

Unknown interpreted text role "class".

This class implements an interface on top of a low-level SSL object as implemented by OpenSSL. This object captures the state of an SSL connection but does not provide any network IO itself. IO needs to be performed through separate "BIO" objects which are OpenSSL's IO abstraction layer.

This class has no public constructor. An :class:`SSLObject` instance must be created using the :meth:`~SSLContext.wrap_bio` method. This method will create the :class:`SSLObject` instance and bind it to a pair of BIOs. The *incoming* BIO is used to pass data from Python to the SSL protocol instance, while the *outgoing* BIO is used to pass data the other way around.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2497); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2497); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2497); *backlink*

Unknown interpreted text role "class".

The following methods are available:

- :attr:`~SSLSocket.context`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2506); *backlink*

  Unknown interpreted text role "attr".

- :attr:`~SSLSocket.server_side`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2507); *backlink*

  Unknown interpreted text role "attr".

- :attr:`~SSLSocket.server_hostname`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2508); *backlink*

  Unknown interpreted text role "attr".

- :attr:`~SSLSocket.session`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2509); *backlink*

  Unknown interpreted text role "attr".

- :attr:`~SSLSocket.session_reused`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2510); *backlink*

  Unknown interpreted text role "attr".

- :meth:`~SSLSocket.read`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2511); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~SSLSocket.write`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2512); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~SSLSocket.getpeercert`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2513); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~SSLSocket.selected_alpn_protocol`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2514); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~SSLSocket.selected_npn_protocol`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2515); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~SSLSocket.cipher`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2516); *backlink*

Unknown interpreted text role "meth".

- :meth:`~SSLSocket.shared_ciphers`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2517); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~SSLSocket.compression`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2518); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~SSLSocket.pending`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2519); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~SSLSocket.do_handshake`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2520); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~SSLSocket.verify_client_post_handshake`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2521); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~SSLSocket.unwrap`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2522); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~SSLSocket.get_channel_binding`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2523); *backlink*

  Unknown interpreted text role "meth".

- :meth:`~SSLSocket.version`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2524); *backlink*

  Unknown interpreted text role "meth".

When compared to :class:`SSLSocket`, this object lacks the following features:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2526); *backlink*

Unknown interpreted text role "class".

- Any form of network IO; `recv()` and `send()` read and write only to the underlying :class:`MemoryBIO` buffers.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2529); *backlink*

  Unknown interpreted text role "class".

- There is no *do_handshake_on_connect* machinery. You must always manually call :meth:`~SSLSocket.do_handshake` to start the handshake.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2532); *backlink*

  Unknown interpreted text role "meth".

- There is no handling of *suppress_ragged_eofs*. All end-of-file conditions that are in violation of the protocol are reported via the :exc:`SSLEOFError` exception.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2535); *backlink*

  Unknown interpreted text role "exc".

- The method :meth:`~SSLSocket.unwrap` call does not return anything, unlike for an SSL socket where it returns the underlying socket.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2539); backlink

Unknown interpreted text role "meth".

- The *server_name_callback* callback passed to :meth:`SSLContext.set_servername_callback` will get an :class:`SSLObject` instance instead of a :class:`SSLSocket` instance as its first parameter.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2542); backlink

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2542); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2542); backlink

Unknown interpreted text role "class".

Some notes related to the use of :class:`SSLObject`:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2546); backlink

Unknown interpreted text role "class".

- All IO on an :class:`SSLObject` is :ref:`non-blocking <ssl-nonblocking>`. This means that for example :meth:`~SSLSocket.read` will raise an :exc:`SSLWantReadError` if it needs more data than the incoming BIO has available.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2548); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2548); backlink

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2548); backlink

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2548); backlink

Unknown interpreted text role "exc".

- There is no module-level `wrap_bio()` call like there is for :meth:`~SSLContext.wrap_socket`. An :class:`SSLObject` is always created via an :class:`SSLContext`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2553); backlink

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2553); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2553); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2557)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.7
   :class:`SSLObject` instances must to created with
   :meth:`~SSLContext.wrap_bio`. In earlier versions, it was possible to
   create instances directly. This was never documented or officially
   supported.
```

An SSLObject communicates with the outside world using memory buffers. The class :class:`MemoryBIO` provides a memory buffer that can be used for this purpose. It wraps an OpenSSL memory BIO (Basic IO) object:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2563); backlink

Unknown interpreted text role "class".

A memory buffer that can be used to pass data between Python and an SSL protocol instance.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2572)

Unknown directive type "attribute".

```
.. attribute:: MemoryBIO.pending

   Return the number of bytes currently in the memory buffer.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2576)

Unknown directive type "attribute".

```
.. attribute:: MemoryBIO.eof

   A boolean indicating whether the memory BIO is current at the end-of-file
   position.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2581)

Unknown directive type "method".

```
.. method:: MemoryBIO.read(n=-1)

   Read up to *n* bytes from the memory buffer. If *n* is not specified or
   negative, all bytes are returned.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2586)

Unknown directive type "method".

```
.. method:: MemoryBIO.write(buf)

   Write the bytes from *buf* to the memory BIO. The *buf* argument must be an
   object supporting the buffer protocol.

   The return value is the number of bytes written, which is always equal to
   the length of *buf*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2594)

Unknown directive type "method".

```
.. method:: MemoryBIO.write_eof()

   Write an EOF marker to the memory BIO. After this method has been called, it
   is illegal to call :meth:`~MemoryBIO.write`. The attribute :attr:`eof` will
   become true after all data currently in the buffer has been read.
```

SSL session

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2604)

Unknown directive type "versionadded".

```
.. versionadded:: 3.6
```

Session object used by :attr:`~SSLSocket.session`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2608); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2610)

Unknown directive type "attribute".

```
.. attribute:: id
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2611)

Unknown directive type "attribute".

```
.. attribute:: time
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2612)

Unknown directive type "attribute".

```
.. attribute:: timeout
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2613)

Unknown directive type "attribute".

```
.. attribute:: ticket_lifetime_hint
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2614)

Unknown directive type "attribute".

```
.. attribute:: has_ticket
```

Security considerations

Best defaults

For **client use**, if you don't have any special requirements for your security policy, it is highly recommended that you use the :func:`create_default_context` function to create your SSL context. It will load the system's trusted CA certificates, enable certificate validation and hostname checking, and try to choose reasonably secure protocol and cipher settings.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2625); *backlink*

Unknown interpreted text role "func".

For example, here is how you would use the :class:`smtplib.SMTP` class to create a trusted, secure connection to a SMTP server:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2632); *backlink*

Unknown interpreted text role "class".

```
>>> import ssl, smtplib
>>> smtp = smtplib.SMTP("mail.python.org", port=587)
>>> context = ssl.create_default_context()
>>> smtp.starttls(context=context)
(220, b'2.0.0 Ready to start TLS')
```

If a client certificate is needed for the connection, it can be added with :meth:`SSLContext.load_cert_chain`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2641); *backlink*

Unknown interpreted text role "meth".

By contrast, if you create the SSL context by calling the :class:`SSLContext` constructor yourself, it will not have certificate validation nor hostname checking enabled by default. If you do so, please read the paragraphs below to achieve a good security level.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2644); *backlink*

Unknown interpreted text role "class".

Manual settings

Verifying certificates

When calling the :class:`SSLContext` constructor directly, :const:`CERT_NONE` is the default. Since it does not authenticate the other peer, it can be insecure, especially in client mode where most of time you would like to ensure the authenticity of the server you're talking to. Therefore, when in client mode, it is highly recommended to use :const:`CERT_REQUIRED`. However, it is in itself not sufficient; you also have to check that the server certificate, which can be obtained by calling :meth:`SSLSocket.getpeercert`, matches the desired service. For many protocols and applications, the service can be identified by the hostname; in this case, the :func:`match_hostname` function can be used. This common check is automatically performed when :attr:`SSLContext.check_hostname` is enabled.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2655); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2655); *backlink*

Unknown interpreted text role "const".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2655); *backlink*

Unknown interpreted text role "const".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2655); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2655); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2655); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2668)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.7
   Hostname matchings is now performed by OpenSSL. Python no longer uses
   :func:`match_hostname`.
```

In server mode, if you want to authenticate your clients using the SSL layer (rather than using a higher-level authentication mechanism), you'll also have to specify :const:`CERT_REQUIRED` and similarly check the client certificate.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2672); *backlink*

Unknown interpreted text role "const".

Protocol versions

SSL versions 2 and 3 are considered insecure and are therefore dangerous to use. If you want maximum compatibility between clients and servers, it is recommended to use :const:`PROTOCOL_TLS_CLIENT` or :const:`PROTOCOL_TLS_SERVER` as the protocol version. SSLv2 and SSLv3 are disabled by default.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2680); ***backlink***

Unknown interpreted text role "const".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2680); ***backlink***

Unknown interpreted text role "const".

```
>>> client_context = ssl.SSLContext(ssl.PROTOCOL_TLS_CLIENT)
>>> client_context.minimum_version = ssl.TLSVersion.TLSv1_3
>>> client_context.maximum_version = ssl.TLSVersion.TLSv1_3
```

The SSL context created above will only allow TLSv1.2 and later (if supported by your system) connections to a server. :const:`PROTOCOL_TLS_CLIENT` implies certificate validation and hostname checks by default. You have to load certificates into the context.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2693); ***backlink***

Unknown interpreted text role "const".

Cipher selection

If you have advanced security requirements, fine-tuning of the ciphers enabled when negotiating a SSL session is possible through the :meth:`SSLContext.set_ciphers` method. Starting from Python 3.2.3, the ssl module disables certain weak ciphers by default, but you may want to further restrict the cipher choice. Be sure to read OpenSSL's documentation about the cipher list format. If you want to check which ciphers are enabled by a given cipher list, use :meth:`SSLContext.get_ciphers` or the `openssl ciphers` command on your system.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2702); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2702); ***backlink***

Unknown interpreted text role "meth".

Multi-processing

If using this module as part of a multi-processed application (using, for example the :mod:`multiprocessing` or :mod:`concurrent.futures` modules), be aware that OpenSSL's internal random number generator does not properly handle forked processes. Applications must change the PRNG state of the parent process if they use any SSL feature with :func:`os.fork`. Any successful call of :func:`~ssl.RAND_add`, :func:`~ssl.RAND_bytes` or :func:`~ssl.RAND_pseudo_bytes` is sufficient.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2715); ***backlink***

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2715); ***backlink***

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2715); ***backlink***

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2715); ***backlink***

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2715); ***backlink***

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2715); ***backlink***

Unknown interpreted text role "func".

TLS 1.3

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2729)

Unknown directive type "versionadded".

```
.. versionadded:: 3.7
```

The TLS 1.3 protocol behaves slightly differently than previous version of TLS/SSL. Some new TLS 1.3 features are not yet available.

- TLS 1.3 uses a disjunct set of cipher suites. All AES-GCM and ChaCha20 cipher suites are enabled by default. The method :meth:`SSLContext.set_ciphers` cannot enable or disable any TLS 1.3 ciphers yet, but :meth:`SSLContext.get_ciphers` returns them.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)ssl.rst, line 2734);

backlink

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2734); **backlink**

Unknown interpreted text role "meth".

- Session tickets are no longer sent as part of the initial handshake and are handled differently. :attr:`SSLSocket.session` and :class:`SSLSession` are not compatible with TLS 1.3.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2738); **backlink**

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2738); **backlink**

Unknown interpreted text role "class".

- Client-side certificates are also no longer verified during the initial handshake. A server can request a certificate at any time. Clients process certificate requests while they send or receive application data from the server.
- TLS 1.3 features like early data, deferred TLS client cert request, signature algorithm configuration, and rekeying are not supported yet.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)ssl.rst, line 2749)

Unknown directive type "seealso".

```
.. seealso::

   Class :class:`socket.socket`
       Documentation of underlying :mod:`socket` class

   `SSL/TLS Strong Encryption: An Introduction <https://httpd.apache.org/docs/trunk/en/ssl/ssl_intro.html>`_
       Intro from the Apache HTTP Server documentation

   :rfc:`RFC 1422: Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management <1422>`
       Steve Kent

   :rfc:`RFC 4086: Randomness Requirements for Security <4086>`
       Donald E., Jeffrey I. Schiller

   :rfc:`RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile <5280>`
       D. Cooper

   :rfc:`RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2 <5246>`
       T. Dierks et. al.

   :rfc:`RFC 6066: Transport Layer Security (TLS) Extensions <6066>`
       D. Eastlake

   `IANA TLS: Transport Layer Security (TLS) Parameters <https://www.iana.org/assignments/tls-parameters/tls-parameters.xml>`_
       IANA

   :rfc:`RFC 7525: Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)
       IETF

   `Mozilla's Server Side TLS recommendations <https://wiki.mozilla.org/Security/Server_Side_TLS>`_
       Mozilla
```