

python / cpython Public

<> Code Issues 5k+ Pull requests 2.2k Actions Projects Security and q

Commit 39258d3



tiran authored on Apr 17, 2021 Verified

[bpo-43669](#): PEP 644: Require OpenSSL 1.1.1 or newer ([GH-23014](#))

- Remove HAVE_X509_VERIFY_PARAM_SET1_HOST check
- Update hashopenssl to require OpenSSL 1.1.1
- multissltests only OpenSSL > 1.1.0
- ALPN is always supported
- SNI is always supported
- Remove deprecated NPN code. Python wrappers are no-op.
- ECDH is always supported
- Remove OPENSSL_VERSION_1_1 macro
- Remove locking callbacks
- Drop PY_OPENSSL_1_1_API macro
- Drop HAVE_SSL_CTX_CLEAR_OPTIONS macro
- SSL_CTRL_GET_MAX_PROTO_VERSION is always defined now
- security level is always available now
- get_num_tickets is available with TLS 1.3
- X509_V_ERR_MISMATCH is always available now
- Always set SSL_MODE_RELEASE_BUFFERS
- X509_V_FLAG_TRUSTED_FIRST is always available
- get_ciphers is always supported
- SSL_CTX_set_keylog_callback is always available
- Update Modules/Setup with static link example
- Mention PEP in whatsnew
- Drop 1.0.2 and 1.1.0 from GHA tests

[main](#) (#23014) · v3.15.0a8 ... v3.10.0b1

1 parent [b467d9a](#) commit 39258d3

17 files changed

+5,310 -8,440

Top

Filter files...



▼ .github/workflows

- ├─ build.yml
- └─ Doc
 - ├─ using
 - ├─ unix.rst
 - └─ whatsnew
 - ├─ 3.10.rst
 - └─ Lib
 - ├─ ssl.py
 - └─ test
 - ├─ test_ssl.py
 - └─ Misc/NEWS.d/next/Build
 - ├─ 2021-03-30-14-19-39.bpo-43669.IWMUYx.rst
 - └─ Modules
 - ├─ Setup
 - ├─ _hashopenssl.c
 - ├─ _ssl.c
 - └─ _ssl
 - ├─ debughelpers.c
 - └─ clinic
 - ├─ _hashopenssl.c.h
 - └─ _ssl.c.h
 - └─ Tools/ssl
 - ├─ multissltests.py
 - ├─ configure.ac
 - ├─ configure
 - ├─ pyconfig.h.in
 - └─ setup.py

1

Search within code



```

  ▾ .github/workflows/build.yml
  ↑... @@ -177,7 +177,7 @@ jobs:
177 177     strategy:
178 178         fail-fast: false
179 179     matrix:
180 180 -     openssl_ver: [1.0.2u, 1.1.0l, 1.1.1k, 3.0.0-alpha14]
180 180 +     openssl_ver: [1.1.1k, 3.0.0-alpha14]
181 181     env:
182 182         OPENSSL_VER: ${ matrix.openssl_ver }
183 183         MULTISSL_DIR: ${ github.workspace }}/multissl
  ↓...

```

```

  ▾ Doc/using/unix.rst
  <> 📄 ...
  ↑... @@ -135,6 +135,7 @@ some Unices may not have the :program:`env` command, so
  you may need to hardcode
135 135
136 136     To use shell commands in your Python scripts, look at the :mod:`subprocess`
  module.
137 137
138 138 + .. _unix_custom_openssl:
138 139
139 140     Custom OpenSSL
140 141     =====
  ↓...

```

```

  ▾ Doc/whatsnew/3.10.rst
  <> 📄 ...
  ↑... @@ -65,6 +65,7 @@ Summary -- Release highlights
65 65
66 66     .. PEP-sized items next.
67 67
68 68 + * :pep:`644`, require OpenSSL 1.1.1 or newer
68 69
69 70
70 71     New Features
  ↓...
  ↑... @@ -1438,6 +1439,10 @@ CPython bytecode changes
1438 1439     Build Changes
1439 1440     =====

```

1440	1441	
1442	1443	+ * :pep:`644`: Python now requires OpenSSL 1.1.1 or newer. OpenSSL 1.0.2 is no longer supported.
1444	1445	+ (Contributed by Christian Heimes in :issue:`43669` .)
1441	1442	+ * The C99 functions <code>:c:func:`snprintf`</code> and <code>:c:func:`vsprintf`</code> are now required
1442	1443	to build Python.
1443	1444	(Contributed by Victor Stinner in :issue:`36020` .)
		@@ -1483,7 +1488,6 @@ Build Changes
1483	1484	(Contributed by Christian Heimes in :issue:`43466` .)
1484	1485	
1485	1486	
1486	1487	-
1487	1488	C API Changes
1488	1489	=====
1489	1490	

Lib/ssl.py		...
		@@ -909,15 +909,12 @@ def selected_npn_protocol(self):
909	909	"""Return the currently selected NPN protocol as a string, or
		``None``
910	910	if a next protocol was not negotiated or if NPN is not supported by
		one
911	911	of the peers."""
912	912	- if _ssl.HAS_NPN:
913	913	- return self._sslobj.selected_npn_protocol()
914	914	
915	915	def selected_alpn_protocol(self):
916	916	"""Return the currently selected ALPN protocol as a string, or
		``None``
917	917	if a next protocol was not negotiated or if ALPN is not supported by
		one
918	918	of the peers."""
919	919	- if _ssl.HAS_ALPN:
920	920	- return self._sslobj.selected_alpn_protocol()
	917	+ return self._sslobj.selected_alpn_protocol()
921	921	

```

922     919         def cipher(self):
923     920             """Return the currently selected cipher as a 3-tuple ``(name,
@@ -1126,10 +1123,7 @@ def getpeercert(self, binary_form=False):
    ↓
    ↑
1126    1123         @_sslcopydoc
1127    1124         def selected_npn_protocol(self):
1128    1125             self._checkClosed()
1129    -         if self._sslobj is None or not _ssl.HAS_NPN:
1130    -             return None
1131    -         else:
1132    -             return self._sslobj.selected_npn_protocol()
+         return None
1133    1127
1134    1128         @_sslcopydoc
1135    1129         def selected_alpn_protocol(self):
    ↓

```

Lib/test/test_ssl.py



```

    ↑
@@ -40,7 +40,6 @@
40     40     PROTOCOLS = sorted(_ssl._PROTOCOL_NAMES)
41     41     HOST = socket_helper.HOST
42     42     IS_LIBRESSL = _ssl.OPENSSL_VERSION.startswith('LibreSSL')
43     - IS_OPENSSL_1_1_0 = not IS_LIBRESSL and _ssl.OPENSSL_VERSION_INFO >= (1, 1, 0)
44     43     IS_OPENSSL_1_1_1 = not IS_LIBRESSL and _ssl.OPENSSL_VERSION_INFO >= (1, 1, 1)
45     44     IS_OPENSSL_3_0_0 = not IS_LIBRESSL and _ssl.OPENSSL_VERSION_INFO >= (3, 0, 0)
46     45     PY_SSL_DEFAULT_CIPHERS = sysconfig.get_config_var('PY_SSL_DEFAULT_CIPHERS')
    ↓
    ↑
@@ -270,18 +269,6 @@ def handle_error(prefix):
270    269         if support.verbose:
271    270             sys.stdout.write(prefix + exc_format)
272    271
273    - def can_clear_options():
274    -     # 0.9.8m or higher
275    -     return _ssl.OPENSSL_API_VERSION >= (0, 9, 8, 13, 15)
276    -
277    - def no_sslv2_implies_sslv3_hello():
278    -     # 0.9.7h or higher
279    -     return _ssl.OPENSSL_VERSION_INFO >= (0, 9, 7, 8, 15)
280    -
281    - def have_verify_flags():

```

282	-	# 0.9.8 or higher
283	-	return ssl.OPENSSL_VERSION_INFO >= (0, 9, 8, 0, 15)
284	-	
285	272	def _have_secp_curves():
286	273	if not ssl.HAS_ECDH:
287	274	return False
⋮ ↓ ↑ ⋮		@@ -372,17 +359,15 @@ def test_constants(self):
372	359	ssl.OP_SINGLE_DH_USE
373	360	if ssl.HAS_ECDH:
374	361	ssl.OP_SINGLE_ECDH_USE
375	-	if ssl.OPENSSL_VERSION_INFO >= (1, 0):
376	-	ssl.OP_NO_COMPRESSION
362	+	ssl.OP_NO_COMPRESSION
377	363	self.assertIn(ssl.HAS_SNI, {True, False})
378	364	self.assertIn(ssl.HAS_ECDH, {True, False})
379	365	ssl.OP_NO_SSLv2
380	366	ssl.OP_NO_SSLv3
381	367	ssl.OP_NO_TLSv1
382	368	ssl.OP_NO_TLSv1_3
383	-	if ssl.OPENSSL_VERSION_INFO >= (1, 0, 1):
384	-	ssl.OP_NO_TLSv1_1
385	-	ssl.OP_NO_TLSv1_2
369	+	ssl.OP_NO_TLSv1_1
370	+	ssl.OP_NO_TLSv1_2
386	371	self.assertEqual(ssl.PROTOCOL_TLS, ssl.PROTOCOL_SSLv23)
387	372	
388	373	def test_private_init(self):
⋮ ↓ ↑ ⋮		@@ -1161,7 +1146,6 @@ def test_python_ciphers(self):
1161	1146	self.assertNotIn("RC4", name)
1162	1147	self.assertNotIn("3DES", name)
1163	1148	
1164	-	@unittest.skipIf(ssl.OPENSSL_VERSION_INFO < (1, 0, 2, 0, 0), 'OpenSSL too old')
1165	1149	def test_get_ciphers(self):
1166	1150	ctx = ssl.SSLContext(ssl.PROTOCOL_TLS_CLIENT)
1167	1151	ctx.set_ciphers('AESGCM')
⋮ ↑ ⋮		@@ -1181,15 +1165,11 @@ def test_options(self):
1181	1165	self.assertEqual(default, ctx.options)

```

1182 1166         ctx.options |= ssl.OP_NO_TLSv1
1183 1167         self.assertEqual(default | ssl.OP_NO_TLSv1, ctx.options)
1184 -         if can_clear_options():
1185 -             ctx.options = (ctx.options & ~ssl.OP_NO_TLSv1)
1186 -             self.assertEqual(default, ctx.options)
1187 -             ctx.options = 0
1188 -             # Ubuntu has OP_NO_SSLv3 forced on by default
1189 -             self.assertEqual(0, ctx.options & ~ssl.OP_NO_SSLv3)
1190 -         else:
1191 -             with self.assertRaises(ValueError):
1192 -                 ctx.options = 0
1168 +         ctx.options = (ctx.options & ~ssl.OP_NO_TLSv1)
1169 +         self.assertEqual(default, ctx.options)
1170 +         ctx.options = 0
1171 +         # Ubuntu has OP_NO_SSLv3 forced on by default
1172 +         self.assertEqual(0, ctx.options & ~ssl.OP_NO_SSLv3)
1193 1173
1194 1174         def test_verify_mode_protocol(self):
1195 1175             ctx = ssl.SSLContext(ssl.PROTOCOL_TLS)
1196 1176
1197 1177
1198 1178
1199 1179
1200 1180
1201 1181
1202 1182
1203 1183
1204 1184
1205 1185
1206 1186
1207 1187
1208 1188
1209 1189
1210 1190
1211 1191
1212 1192
1213 1193
1214 1194
1215 1195
1216 1196
1217 1197
1218 1198
1219 1199
1220 1200
1221 1201
1222 1202
1223 1203
1224 1204
1225 1205
1226 1206
1227 1207
1228 1208
1229 1209
1230 1210
1231 1211
1232 1212
1233 1213
1234 1214
1235 1215
1236 1216
1237 1217
1238 1218
1239 1219
1240 1220
1241 1221
1242 1222
1243 1223
1244 1224
1245 1225
1246 1226
1247 1227
1248 1228
1249 1229
1250 1230
1251 1231
1252 1232
1253 1233
1254 1234
1255 1235
1256 1236
1257 1237
1258 1238
1259 1239
1260 1240
1261 1241
1262 1242
1263 1243
1264 1244
1265 1245
1266 1246
1267 1247
1268 1248
1269 1249
1270 1250
1271 1251
1272 1252
1273 1253
1274 1254
1275 1255
1276 1256
1277 1257
1278 1258
1279 1259
1280 1260
1281 1261
1282 1262
1283 1263
1284 1264
1285 1265
1286 1266
1287 1267
1288 1268
1289 1269
1290 1270
1291 1271
1292 1272
1293 1273
1294 1274
1295 1275
1296 1276
1297 1277
1298 1278
1299 1279
1300 1280
1301 1281
1302 1282
1303 1283
1304 1284
1305 1285
1306 1286
1307 1287
1308 1288
1309 1289
1310 1290
1311 1291
1312 1292
1313 1293
1314 1294
1315 1295
1316 1296
1317 1297
1318 1298
1319 1299
1320 1300
1321 1301
1322 1302
1323 1303
1324 1304
1325 1305
1326 1306
1327 1307
1328 1308
1329 1309
1330 1310
1331 1311
1332 1312
1333 1313
1334 1314
1335 1315
1336 1316
1337 1317
1338 1318
1339 1319
1340 1320
1341 1321
1342 1322
1343 1323
1344 1324
1345 1325
1346 1326
1347 1327
1348 1328
1349 1329
1350 1330
1351 1331
1352 1332
1353 1333
1354 1334
1355 1335
1356 1336
1357 1337
1358 1338
1359 1339
1360 1340
1361 1341
1362 1342
1363 1343
1364 1344
1365 1345
1366 1346
1367 1347
1368 1348
1369 1349
1370 1350
1371 1351
1372 1352
1373 1353
1374 1354
1375 1355
1376 1356
1377 1357
1378 1358
1379 1359
1380 1360
1381 1361
1382 1362
1383 1363
1384 1364
1385 1365
1386 1366
1387 1367
1388 1368
1389 1369
1390 1370
1391 1371
1392 1372
1393 1373
1394 1374
1395 1375
1396 1376
1397 1377
1398 1378
1399 1379
1400 1380
1401 1381
1402 1382
1403 1383
1404 1384
1405 1385
1406 1386
1407 1387
1408 1388
1409 1389
1410 1390
1411 1391
1412 1392
1413 1393
1414 1394
1415 1395
1416 1396
1417 1397
1418 1398
1419 1399
1420 1400
1421 1401
1422 1402
1423 1403
1424 1404
1425 1405
1426 1406
1427 1407
1428 1408
1429 1409
1430 1410
1431 1411
1432 1412
1433 1413
1434 1414
1435 1415
1436 1416
1437 1417
1438 1418
1439 1419
1440 1420
1441 1421
1442 1422
1443 1423
1444 1424
1445 1425
1446 1426
1447 1427
1448 1428
1449 1429
1450 1430
1451 1431
1452 1432
1453 1433
1454 1434
1455 1435
1456 1436
1457 1437
1458 1438
1459 1439
1460 1440
1461 1441
1462 1442
1463 1443
1464 1444
1465 1445
1466 1446
1467 1447
1468 1448
1469 1449
1470 1450
1471 1451
1472 1452
1473 1453
1474 1454
1475 1455
1476 1456
1477 1457
1478 1458
1479 1459
1480 1460
1481 1461
1482 1462
1483 1463
1484 1464
1485 1465
1486 1466
1487 1467
1488 1468
1489 1469
1490 1470
1491 1471
1492 1472
1493 1473
1494 1474
1495 1475
1496 1476
1497 1477
1498 1478
1499 1479
1500 1480
1501 1481
1502 1482
1503 1483
1504 1484
1505 1485
1506 1486
1507 1487
1508 1488
1509 1489
1510 1490
1511 1491
1512 1492
1513 1493
1514 1494
1515 1495
1516 1496
1517 1497
1518 1498
1519 1499
1520 1500
1521 1501
1522 1502
1523 1503
1524 1504
1525 1505
1526 1506
1527 1507
1528 1508
1529 1509
1530 1510
1531 1511
1532 1512
1533 1513
1534 1514
1535 1515
1536 1516
1537 1517
1538 1518
1539 1519
1540 1520
1541 1521
1542 1522
1543 1523
1544 1524
1545 1525
1546 1526
1547 1527
1548 1528
1549 1529
1550 1530
1551 1531
1552 1532
1553 1533
1554 1534
1555 1535
1556 1536
1557 1537
1558 1538
1559 1539
1560 1540
1561 1541
1562 1542
1563 1543
1564 1544
1565 1545
1566 1546
1567 1547
1568 1548
1569 1549
1570 1550
1571 1551
1572 1552
1573 1553
1574 1554
1575 1555
1576 1556
1577 1557
1578 1558
1579 1559
1580 1560
1581 1561
1582 1562
1583 1563
1584 1564
1585 1565
1586 1566
1587 1567
1588 1568
1589 1569
1590 1570
1591 1571
1592 1572
1593 1573
1594 1574
1595 1575
1596 1576
1597 1577
1598 1578
1599 1579
1600 1580
1601 1581
1602 1582
1603 1583
1604 1584
1605 1585
1606 1586
1607 1587
1608 1588
1609 1589
1610 1590
1611 1591
1612 1592
1613 1593
1614 1594
1615 1595
1616 1596
1617 1597
1618 1598
1619 1599
1620 1600
1621 1601
1622 1602
1623 1603
1624 1604
1625 1605
1626 1606
1627 1607
1628 1608
1629 1609
1630 1610
1631 1611
1632 1612
1633 1613
1634 1614
1635 1615
1636 1616
1637 1617
1638 1618
1639 1619
1640 1620
1641 1621
1642 1622
1643 1623
1644 1624
1645 1625
1646 1626
1647 1627
1648 1628
1649 1629
1650 1630
1651 1631
1652 1632
1653 1633
1654 1634
1655 1635
1656 1636
1657 1637
1658 1638
1659 1639
1660 1640
1661 1641
1662 1642
1663 1643
1664 1644
1665 1645
1666 1646
1667 1647
1668 1648
1669 1649
1670 1650
1671 1651
1672 1652
1673 1653
1674 1654
1675 1655
1676 1656
1677 1657
1678 1658
1679 1659
1680 1660
1681 1661
1682 1662
1683 1663
1684 1664
1685 1665
1686 1666
1687 1667
1688 1668
1689 1669
1690 1670
1691 1671
1692 1672
1693 1673
1694 1674
1695 1675
1696 1676
1697 1677
1698 1678
1699 1679
1700 1680
1701 1681
1702 1682
1703 1683
1704 1684
1705 1685
1706 1686
1707 1687
1708 1688
1709 1689
1710 1690
1711 1691
1712 1692
1713 1693
1714 1694
1715 1695
1716 1696
1717 1697
1718 1698
1719 1699
1720 1700
1721 1701
1722 1702
1723 1703
1724 1704
1725 1705
1726 1706
1727 1707
1728 1708
1729 1709
1730 1710
1731 1711
1732 1712
1733 1713
1734 1714
1735 1715
1736 1716
1737 1717
1738 1718
1739 1719
1740 1720
1741 1721
1742 1722
1743 1723
1744 1724
1745 1725
1746 1726
1747 1727
1748 1728
1749 1729
1750 1730
1751 1731
1752 1732
1753 1733
1754 1734
1755 1735
1756 1736
1757 1737
1758 1738
1759 1739
1760 1740
1761 1741
1762 1742
1763 1743
1764 1744
1765 1745
1766 1746
1767 1747
1768 1748
1769 1749
1770 1750
1771 1751
1772 1752
1773 1753
1774 1754
1775 1755
1776 1756
1777 1757
1778 1758
1779 1759
1780 1760
1781 1761
1782 1762
1783 1763
1784 1764
1785 1765
1786 1766
1787 1767
1788 1768
1789 1769
1790 1770
1791 1771
1792 1772
1793 1773
1794 1774
1795 1775
1796 1776
1797 1777
1798 1778
1799 1779
1800 1780
1801 1781
1802 1782
1803 1783
1804 1784
1805 1785
1806 1786
1807 1787
1808 1788
1809 1789
1810 1790
1811 1791
1812 1792
1813 1793
1814 1794
1815 1795
1816 1796
1817 1797
1818 1798
1819 1799
1820 1800
1821 1801
1822 1802
1823 1803
1824 1804
1825 1805
1826 1806
1827 1807
1828 1808
1829 1809
1830 1810
1831 1811
1832 1812
1833 1813
1834 1814
1835 1815
1836 1816
1837 1817
1838 1818
1839 1819
1840 1820
1841 1821
1842 1822
1843 1823
1844 1824
1845 1825
1846 1826
1847 1827
1848 1828
1849 1829
1850 1830
1851 1831
1852 1832
1853 1833
1854 1834
1855 1835
1856 1836
1857 1837
1858 1838
1859 1839
1860 1840
1861 1841
1862 1842
1863 1843
1864 1844
1865 1845
1866 1846
1867 1847
1868 1848
1869 1849
1870 1850
1871 1851
1872 1852
1873 1853
1874 1854
1875 1855
1876 1856
1877 1857
1878 1858
1879 1859
1880 1860
1881 1861
1882 1862
1883 1863
1884 1864
1885 1865
1886 1866
1887 1867
1888 1868
1889 1869
1890 1870
1891 1871
1892 1872
1893 1873
1894 1874
1895 1875
1896 1876
1897 1877
1898 1878
1899 1879
1900 1880
1901 1881
1902 1882
1903 1883
1904 1884
1905 1885
1906 1886
1907 1887
1908 1888
1909 1889
1910 1890
1911 1891
1912 1892
1913 1893
1914 1894
1915 1895
1916 1896
1917 1897
1918 1898
1919 1899
1920 1900
1921 1901
1922 1902
1923 1903
1924 1904
1925 1905
1926 1906
1927 1907
1928 1908
1929 1909
1930 1910
1931 1911
1932 1912
1933 1913
1934 1914
1935 1915
1936 1916
1937 1917
1938 1918
1939 1919
1940 1920
1941 1921
1942 1922
1943 1923
1944 1924
1945 1925
1946 1926
1947 1927
1948 1928
1949 1929
1950 1930
1951 1931
1952 1932
1953 1933
1954 1934
1955 1935
1956 1936
1957 1937
1958 1938
1959 1939
1960 1940
1961 1941
1962 1942
1963 1943
1964 1944
1965 1945
1966 1946
1967 1947
1968 1948
1969 1949
1970 1950
1971 1951
1972 1952
1973 1953
1974 1954
1975 1955
1976 1956
1977 1957
1978 1958
1979 1959
1980 1960
1981 1961
1982 1962
1983 1963
1984 1964
1985 1965
1986 1966
1987 1967
1988 1968
1989 1969
1990 1970
1991 1971
1992 1972
1993 1973
1994 1974
1995 1975
1996 1976
1997 1977
1998 1978
1999 1979
2000 1980
2001 1981
2002 1982
2003 1983
2004 1984
2005 1985
2006 1986
2007 1987
2008 1988
2009 1989
2010 1990
2011 1991
2012 1992
2013 1993
2014 1994
2015 1995
2016 1996
2017 1997
2018 1998
2019 1999
2020 2000
2021 2001
2022 2002
2023 2003
2024 2004
2025 2005
2026 2006
2027 2007
2028 2008
2029 2009
2030 2010
2031 2011
2032 2012
2033 2013
2034 2014
2035 2015
2036 2016
2037 2017
2038 2018
2039 2019
2040 2020
2041 2021
2042 2022
2043 2023
2044 2024
2045 2025
2046 2026
2047 2027
2048 2028
2049 2029
2050 2030
2051 2031
2052 2032
2053 2033
2054 2034
2055 2035
2056 2036
2057 2037
2058 2038
2059 2039
2060 2040
2061 2041
2062 2042
2063 2043
2064 2044
2065 2045
2066 2046
2067 2047
2068 2048
2069 2049
2070 2050
2071 2051
2072 2052
2073 2053
2074 2054
2075 2055
2076 2056
2077 2057
2078 2058
2079 2059
2080 2060
2081 2061
2082 2062
2083 2063
2084 2064
2085 2065
2086 2066
2087 2067
2088 2068
2089 2069
2090 2070
2091 2071
2092 2072
2093 2073
2094 2074
2095 2075
2096 2076
2097 2077
2098 2078
2099 2079
2100 2080
2101 2081
2102 2082
2103 2083
2104 2084
2105 2085
2106 2086
2107 2087
2108 2088
2109 2089
2110 2090
2111 2091
2112 2092
2113 2093
2114 2094
2115 2095
2116 2096
2117 2097
2118 2098
2119 2099
2120 2100
2121 2101
2122 2102
2123 2103
2124 2104
2125 2105
2126 2106
2127 2107
2128 2108
2129 2109
2130 2110
2131 2111
2132 2112
2133 2113
2134 2114
2135 2115
2136 2116
2137 2117
2138 2118
2139 2119
2140 2120
2141 2121
2142 2122
2143 2123
2144 2124
2145 2125
2146 2126
2147 2127
2148 2128
2149 2129
2150 2130
2151 2131
2152 2132
2153 2133
2154 2134
2155 2135
2156 2136
2157 2137
2158 2138
2159 2139
2160 2140
2161 2141
2162 2142
2163 2143
2164 2144
2165 2145
2166 2146
2167 2147
2168 2148
2169 2149
2170 2150
2171 2151
2172 2152
2173 2153
2174 2154
2175 2155
2176 2156
2177 2157
2178 2158
2179 2159
2180 2160
2181 2161
2182 2162
2183 2163
2184 2164
2185 2165
2186 2166
2187 2167
2188 2168
2189 2169
2190 2170
2191 2171
2192 2172
2193 2173
2194 2174
2195 2175
2196 2176
2197 2177
2198 2178
2199 2179
2200 2180
2201 2181
2202 2182
2203 2183
2204 2184
2205 2185
2206 2186
2207 2187
2208 2188
2209 2189
2210 2190
2211 2191
2212 2192
2213 2193
2214 2194
2215 2195
2216 2196
2217 2197
2218 2198
2219 2199
2220 2200
2221 2201
2222 2202
2223 2203
2224 2204
2225 2205
2226 2206
2227 2207
2228 2208
2229 2209
2230 2210
2231 2211
2232 2212
2233 2213
2234 2214
2235 2215
2236 2216
2237 2217
2238 2218
2239 2219
2240 2220
2241 2221
2242 2222
2243 2223
2244 2224
2245 2225
2246 2226
2247 2227
2248 2228
2249 2229
2250 2230
2251 2231
2252 2232
2253 2233
2254 2234
2255 2235
2256 2236
2257 2237
2258 2238
2259 2239
2260 2240
2261 2241
2262 2242
2263 2243
2264 2244
2265 2245
2266 2246
2267 2247
2268 2248
2269 2249
2270 2250
2271 2251
2272 2252
2273 2253
2274 2254
2275 2255
2276 2256
2277 2257
2278 2258
2279 2259
2280 2260
2281 2261
2282 2262
2283 2263
2284 2264
2285 2265
2286 2266
2287 2267
2288 2268
2289 2269
2290 2270
2291 2271
2292 2272
2293 2273
2294 2274
2295 2275
2296 2276
2297 2277
2298 2278
2299 2279
2300 2280
2301 2281
2302 2282
2303 2283
2304 2284
2305 2285
2306 2286
2307 2287
2308 2288
2309 2289
2310 2290
2311 2291
2312 2292
2313 2293
2314 2294
2315 2295
2316 2296
2317 2297
2318 2298
2319 2299
2320 2300
2321 2301
2322 2302
2323 2303
2324 2304
2325 2305
2326 2306
2327 2307
2328 2308
2329 2309
2330 2310
2331 2311
2332 2312
2333 2313
2334 2314
2335 2315
2336 2316
2337 2317
2338 2318
2339 2319
2340 2320
2341 2321
2342 2322
2343 2323
2344 2324
2345 2325
2346 2326
2347 2327
2348 2328
2349 2329
2350 2330
2351 2331
2352 2332
2353 2333
2354 2334
2355 2335
2356 2336
2357 2337
2358 2338
2359 2339
2360 2340
2361 2341
2362 2342
2363 2343
2364 2344
2365 2345
2366 2346
2367 2347
2368 2348
2369 2349
2370 2350
2371 2351
2372 2352
2373 2353
2374 2354
2375 2355
2376 2356
2377 2357
2378 2358
2379 2359
2380 2360
2381 2361
2382 2362
2383 2363
2384 2364
2385 2365

```

2956	2933	after = ssl.cert_time_to_seconds(cert['notAfter'])
2957	2934	self.assertLess(before, after)
2958	2935	
2959	-	@unittest.skipUnless(have_verify_flags(),
2960	-	"verify_flags need OpenSSL > 0.9.8")
2961	2936	def test_crl_check(self):
2962	2937	if support.verbose:
2963	2938	sys.stdout.write("\n")
		@@ -3859,12 +3834,7 @@ def test_version_basic(self):
3859	3834	self.assertIs(s.version(), None)
3860	3835	self.assertIs(s._sslobj, None)
3861	3836	s.connect((HOST, server.port))
3862	-	if IS_OPENSSL_1_1_1 and has_tls_version('TLSv1.3'):
3863	-	self.assertEqual(s.version(), 'TLSv1.3')
3864	-	elif ssl.OPENSSL_VERSION_INFO >= (1, 0, 2):
3865	-	self.assertEqual(s.version(), 'TLSv1.2')
3866	-	else: # 0.9.8 to 1.0.1
3867	-	self.assertIn(s.version(), ('TLSv1', 'TLSv1.2'))
	3837	+ self.assertEqual(s.version(), 'TLSv1.3')
3868	3838	self.assertIs(s._sslobj, None)
3869	3839	self.assertIs(s.version(), None)
3870	3840	
		@@ -3966,8 +3936,6 @@ def test_default_ecdh_curve(self):
3966	3936	# explicitly using the 'ECCdraft' cipher alias. Otherwise,
3967	3937	# our default cipher list should prefer ECDH-based ciphers
3968	3938	# automatically.
3969	-	if ssl.OPENSSL_VERSION_INFO < (1, 0, 0):
3970	-	context.set_ciphers("ECCdraft:ECDH")
3971	3939	with ThreadedEchoServer(context=context) as server:
3972	3940	with context.wrap_socket(socket.socket()) as s:
3973	3941	s.connect((HOST, server.port))
		@@ -4099,15 +4067,11 @@ def test_ecdh_curve(self):
4099	4067	server_context.set_ciphers("ECDHE:!eNULL:!aNULL")
4100	4068	server_context.options = ssl.OP_NO_TLSv1 ssl.OP_NO_TLSv1_1
4101	4069	try:
4102	-	stats = server_params_test(client_context, server_context,
4103	-	chatty=True, connectionchatty=True,

```

4104 -             sni_name=hostname)
4105 +
4106 +         server_params_test(client_context, server_context,
4107 +                             chatty=True, connectionchatty=True,
4108 +                             sni_name=hostname)
4109 -
4110 -     except ssl.SSLError:
4111 -         pass
4112 -     else:
4113 -         # OpenSSL 1.0.2 does not fail although it should.
4114 -         if IS_OPENSSL_1_1_0:
4115 -             self.fail("mismatch curve did not fail")
4116 +         self.fail("mismatch curve did not fail")
4117 +
4118 +
4119 +
4120 +
4121 +
4122 +
4123 +
4124 +
4125 +
4126 +
4127 +
4128 +
4129 +
4130 +
4131 +
4132 +
4133 +
4134 +
4135 +
4136 +
4137 +
4138 +
4139 +
4140 +
4141 +
4142 +
4143 +
4144 +
4145 +
4146 +
4147 +
4148 +
4149 +
4150 +
4151 +
4152 +
4153 +
4154 +
4155 +
4156 +
4157 +

```

```

4158 -         msg = "failed trying %s (s) and %s (c).\n" \
4159 -             "was expecting %s, but got %s from the %s" \
4160 -             % (str(server_protocols), str(client_protocols),
4161 -               str(expected))
4162 -         client_result = stats['client_alpn_protocol']
4163 -         self.assertEqual(client_result, expected,
4164 -                          msg % (client_result, "client"))
4165 -         server_result = stats['server_alpn_protocols'][-1] \
4166 -             if len(stats['server_alpn_protocols']) else 'nothing'
4167 -         self.assertEqual(server_result, expected,
4168 -                          msg % (server_result, "server"))

```

```

4115 +         msg = "failed trying %s (s) and %s (c).\n" \
4116 +             "was expecting %s, but got %s from the %s" \
4117 +             % (str(server_protocols), str(client_protocols),
4118 +               str(expected))
4119 +         client_result = stats['client_alpn_protocol']
4120 +         self.assertEqual(client_result, expected,
4121 +                          msg % (client_result, "client"))
4122 +         server_result = stats['server_alpn_protocols'][-1] \
4123 +             if len(stats['server_alpn_protocols']) else 'nothing'
4124 +         self.assertEqual(server_result, expected,
4125 +                          msg % (server_result, "server"))

```

4169 4126

4170 4127 `def test_selected_npn_protocol(self):`4171 4128 `# selected_npn_protocol() is None unless NPN is used`@@ -4175,31 +4132,8 @@ `def test_selected_npn_protocol(self):`4175 4132 `sni_name=hostname)`4176 4133 `self.assertIs(stats['client_npn_protocol'], None)`

4177 4134

4178 - `@unittest.skipUnless(ssl.HAS_NPN, "NPN support needed for this test")`4179 4135 `def test_npn_protocols(self):`4180 - `server_protocols = ['http/1.1', 'spdy/2']`4181 - `protocol_tests = [`4182 - `(['http/1.1', 'spdy/2'], 'http/1.1'),`4183 - `(['spdy/2', 'http/1.1'], 'http/1.1'),`4184 - `(['spdy/2', 'test'], 'spdy/2'),`4185 - `(['abc', 'def'], 'abc')`4186 - `]`4187 - `for client_protocols, expected in protocol_tests:`4188 - `client_context, server_context, hostname = testing_context()`

```

4189 -         server_context.set_npn_protocols(server_protocols)
4190 -         client_context.set_npn_protocols(client_protocols)
4191 -         stats = server_params_test(client_context, server_context,
4192 -                                 chatty=True, connectionchatty=True,
4193 -                                 sni_name=hostname)
4194 -         msg = "failed trying %s (s) and %s (c).\n" \
4195 -              "was expecting %s, but got %s from the %s" \
4196 -              % (str(server_protocols), str(client_protocols),
4197 -                 str(expected))
4198 -         client_result = stats['client_npn_protocol']
4199 -         self.assertEqual(client_result, expected, msg % (client_result,
4200 - "client"))
4201 -         server_result = stats['server_npn_protocols'][-1] \
4202 -             if len(stats['server_npn_protocols']) else 'nothing'
4203 -         self.assertEqual(server_result, expected, msg % (server_result,
4204 - "server"))

```

```
4136 +         assert not ssl.HAS_NPN
```

```
4203 4137
```

```
4204 4138         def sni_contexts(self):
```

```
4205 4139             server_context = ssl.SSLContext(ssl.PROTOCOL_TLS_SERVER)
```



```
@@ -4369,8 +4303,7 @@ def test_session(self):
```

```
4369 4303         self.assertGreater(session.time, 0)
```

```
4370 4304         self.assertGreater(session.timeout, 0)
```

```
4371 4305         self.assertTrue(session.has_ticket)
```

```
4372 -         if ssl.OPENSSL_VERSION_INFO > (1, 0, 1):
```

```
4373 -             self.assertGreater(session.ticket_lifetime_hint, 0)
```

```
4306 +         self.assertGreater(session.ticket_lifetime_hint, 0)
```

```
4374 4307         self.assertFalse(stats['session_reused'])
```

```
4375 4308         sess_stat = server_context.session_stats()
```

```
4376 4309         self.assertEqual(sess_stat['accept'], 1)
```



...ld/2021-03-30-14-19-39.bpo-43669.lWMUYx.rst



```
... @@ -0,0 +1 @@
```

```
1 + Implement :pep:`644`. Python now requires OpenSSL 1.1.1 or newer.
```

Modules/Setup



```
@@ -207,11 +207,23 @@ _symtable symtablemodule.c
```

```
207 207 #_socket socketmodule.c
208 208
209 209 # Socket module helper for SSL support; you must comment out the other
210 - # socket line above, and possibly edit the SSL variable:
211 - #SSL=/usr/local/ssl
212 - #_ssl _ssl.c \
213 - # -DUSE_SSL -I$(SSL)/include -I$(SSL)/include/openssl \
214 - # -L$(SSL)/lib -lssl -lcrypto

210 + # socket line above, and edit the OPENSSL variable:
211 + # OPENSSL=/path/to/openssl/directory
212 + # _ssl _ssl.c \
213 + # -I$(OPENSSL)/include -L$(OPENSSL)/lib \
214 + # -lssl -lcrypto
215 + #_hashlib _hashopenssl.c \
216 + # -I$(OPENSSL)/include -L$(OPENSSL)/lib \
217 + # -lcrypto
218 +
219 + # To statically link OpenSSL:
220 + # _ssl _ssl.c \
221 + # -I$(OPENSSL)/include -L$(OPENSSL)/lib \
222 + # -l:libssl.a -Wl,--exclude-libs,libssl.a \
223 + # -l:libcrypto.a -Wl,--exclude-libs,libcrypto.a
224 + #_hashlib _hashopenssl.c \
225 + # -I$(OPENSSL)/include -L$(OPENSSL)/lib \
226 + # -l:libcrypto.a -Wl,--exclude-libs,libcrypto.a

215 227
216 228 # The crypt module is now disabled by default because it breaks builds
217 229 # on many systems (where -lcrypt is needed), e.g. Linux (I believe).
```



Comments 0