Python Security Documentation

Release 0.0

Victor Stinner

Jun 20, 2017

Contents

This page is an attempt to document security vulnerabilities in Python and the versions including the fix.

CHAPTER 1

Pages

Security vulnerabilities

Total: 44 vulnerabilities.

Vulnerability	Disclosure	Fixed In	Vulnerable
bpo-30500: urllib connects to a wrong host	2017-05-29		2.7 3.3 3.4 3.5 3.6
urllib FTP protocol stream injection	2017-02-20		2.7 3.3 3.4 3.5 3.6
CVE-2016-0718: expat 2.2, bug #537	2017-02-17		2.7 3.3 3.4 3.5 3.6
Issue #28563: gettext.c2py()	2016-10-30	2.7.13 3.4.6 3.5.3 3.6.0	3.3
CVE-2016-2183: Sweet32 attack (DES, 3DES)	2016-08-24	2.7.13 3.5.3 3.6.0	3.4
CVE-2016-1000110: HTTPoxy attack	2016-07-18	2.7.13 3.4.6 3.5.3 3.6.0	3.3
CVE-2016-0772: smtplib TLS stripping	2016-06-11	2.7.12 3.4.5 3.5.2 3.6.0	3.3
Issue #26657: HTTP directory traversal	2016-03-28	2.7.12 3.5.2 3.6.0	3.3 3.4
Issue #26556: Expat 2.1.1	2016-03-14	2.7.12 3.4.5 3.5.2 3.6.0	3.3
CVE-2016-5636: zipimporter overflow	2016-01-21	2.7.12 3.4.5 3.5.2 3.6.0	3.3
CVE-2016-5699: HTTP header injection	2014-11-24	2.7.10 3.4.4 3.5.0	3.3
CVE-2014-9365: Validate TLS certificate	2014-08-28	2.7.9 3.4.3 3.5.0	3.3
CVE-2014-7185: buffer() integer overflows	2014-06-24	2.7.8	–
CVE-2014-4616: JSONDecoder.raw_decode	2014-04-13	2.7.7 3.2.6 3.3.6 3.4.1 3.5.0	–
CVE-2014-2667: os.makedirs() not thread-safe	2014-03-28	3.2.6 3.3.6 3.4.1 3.5.0	–
CVE-2014-1912: socket.recvfrom_into() overflow	2014-01-14	2.7.7 3.2.6 3.3.4 3.4.0	–
CVE-2013-7338: zipfile DoS using malformed file	2013-12-27	3.3.4 3.4.0	–
Issue #19435: CGI directory traversal	2013-10-29	2.7.6 3.2.6 3.3.4 3.4.0	–
CVE-2013-4238: ssl: NUL in subjectAltNames	2013-06-27	2.6.9 2.7.6 3.2.6 3.3.3 3.4.0	–
CVE-2013-7440: ssl.match_hostname() IDNA issue	2013-05-17	3.3.3 3.4.0	–
CVE-2013-2099: ssl.match_hostname() wildcard DoS	2013-05-15	3.2.6 3.3.3 3.4.0	–
CVE-2013-1752: ftplib unlimited read	2012-09-25	2.7.6 3.2.6 3.3.3 3.4.0	–
CVE-2013-1752: nntplib unlimited read	2012-09-25	2.6.9 2.7.6 3.2.6 3.4.3 3.5.0	–
CVE-2013-1752: poplib unlimited read	2012-09-25	2.7.9 3.2.6 3.4.3 3.5.0	–

Continued on next page

Table 1.1 – continued from previous page

Vulnerability	Disclosure	Fixed In	Vulnerable
CVE-2013-1752: smtplib unlimited read	2012-09-25	2.7.9 3.2.6 3.4.3 3.5.0	–
CVE-2013-1753: xmlrpc gzip unlimited read	2012-09-25	2.7.9 3.4.3 3.5.0	3.3
CVE-2013-7040: Hash not properly randomized	2012-04-19	3.4.0	2.7 3.3
CVE-2012-2135: UTF-16 decoder	2012-04-14	2.7.4 3.2.4 3.3.0	–
CVE-2012-0845: XML-RPC DoS	2012-02-13	2.6.8 2.7.3 3.1.5 3.2.3 3.3.0	–
CVE-2011-3389: ssl CBC IV attack	2012-01-27	2.6.8 2.7.3 3.1.5 3.2.3 3.3.0	–
CVE-2012-1150: Hash DoS	2011-12-28	2.6.8 2.7.3 3.1.5 3.2.3 3.3.0	–
CVE-2011-4944: pypirc created insecurely	2011-11-30	2.7.4 3.2.4 3.3.1 3.4.0	–
CVE-2011-1521: urllib redirect	2011-03-24	2.5.6 2.6.7 2.7.2 3.1.4 3.2.1 3.3.0	–
CVE-2011-4940: SimpleHTTPServer UTF-7	2011-03-08	2.5.6 2.6.7 2.7.2 3.2.4 3.3.1 3.4.0	–
CVE-2010-1634: audioop integer overflows	2010-05-10	2.6.6 2.7.0 3.1.3 3.2.0	–
CVE-2010-2089: audioop input validation	2010-01-11	2.6.6 2.7.2 3.1.3 3.2.0	–
CVE-2013-1752: httplib unlimited read	2009-08-28	2.7.2 3.1.4 3.2.0	–
CVE-2010-3492: smtpd accept bug	2009-08-14	2.7.4 3.2.0	–
CVE-2010-3493: smtpd race conditions	2009-08-14	2.7.1 3.1.3 3.2.1 3.3.0	–
CVE-2008-2315: Multiple integer overflows (Apple)	2008-07-31	2.6.0 3.0.0	–
CVE-2008-3143: Multiple integer overflows (Google)	2008-04-11	2.5.3 2.6.0 3.0.0	–
CVE-2008-5031: expandtab() integer overflow	2008-03-11	2.5.3 2.6.0 3.0.0	–
CVE-2011-1015: CGI directory traversal	2008-03-07	2.7.0 3.2.4 3.3.1 3.4.0	–
CVE-2007-4965: rgbimg and imageop overflows	2007-09-16	2.5.3 2.6.0	–

Table of Contents:

bpo-30500: urllib connects to a wrong host

The urllib module doesn't parse correctly password containing the # character.

- Disclosure date: **2017-05-29** (Python issue #30500 reported)
- Reported at: 2017-03-04 (Orange Tsai on the PSRT list)

Vulnerable Versions

- Python **2.7**
- Python **3.3**
- Python **3.4**
- Python **3.5**
- Python **3.6**

Python issue

urllib connects to a wrong host.

- Python issue: issue #30500
- Creation date: 2017-05-29
- Reporter: Nam Nguyen

Timeline

Timeline using the disclosure date **2017-05-29** as reference:

- 2017-03-04 (**-86 days**): Reported (Orange Tsai on the PSRT list)
- 2017-05-29: Python issue #30500 reported by Nam Nguyen

urllib FTP protocol stream injection

FTP protocol stream injection via malicious URLs.

- Disclosure date: **2017-02-20** (blog post, mail to oss-security)
- Reported at: 2016-01-15 (email sent to the PSRT list)
- Reported by: Timothy D. Morgan (Blindspot)

Vulnerable Versions

- Python **2.7**
- Python **3.3**
- Python **3.4**
- Python **3.5**
- Python **3.6**

Python issue

urllib FTP protocol stream injection.

- Python issue: issue #29606
- Creation date: 2017-02-20
- Reporter: ecbftw

Timeline

Timeline using the disclosure date **2017-02-20** as reference:

- 2016-01-15 (**-402 days**): Reported (email sent to the PSRT list)
- 2017-02-20: Disclosure date (blog post, mail to oss-security)
- 2017-02-20 (**+0 days**): Python issue #29606 reported by ecbftw

Links

- http://blog.blindspotsecurity.com/2017/02/advisory-javapython-ftp-injections.html
- http://www.openwall.com/lists/oss-security/2017/02/20/1

CVE-2016-0718: expat 2.2, bug #537

The Expat XML parser mishandles certain kinds of malformed input documents, resulting in buffer overflows during processing and error reporting. The overflows can manifest as a segmentation fault or as memory corruption during a parse operation. The bugs allow for a denial of service attack in many applications by an unauthenticated attacker, and could conceivably result in remote code execution.

CVE-ID:

- CVE-2016-0718
- CVE-2016-4472
- Disclosure date: **2017-02-17** (Python issue #29591 reported)
- Reported by: 2016-05-27 (expat bug #537 reported)

Vulnerable Versions

- Python **2.7**
- Python **3.3**
- Python **3.4**
- Python **3.5**
- Python **3.6**

Python issue

Various security vulnerabilities in bundled expat (CVE-2016-0718 and CVE-2016-4472).

- Python issue: issue #29591
- Creation date: 2017-02-17
- Reporter: Natanael Copa

CVE-2016-0718

Expat allows context-dependent attackers to cause a denial of service (crash) or possibly execute arbitrary code via a malformed input document, which triggers a buffer overflow.

- CVE ID: CVE-2016-0718
- Published: 2016-05-26
- CVSS Score: 7.5

Timeline

Timeline using the disclosure date **2017-02-17** as reference:

- 2016-05-26 (**-267 days**): CVE-2016-0718 published
- 2017-02-17: Python issue #29591 reported by Natanael Copa

Links

- https://sourceforge.net/p/expat/bugs/537/
- https://bugs.python.org/issue30610

Issue #28563: gettext.c2py()

Arbitrary code execution in `gettext.c2py()`.

- Disclosure date: **2016-10-30** (Python issue #28563 reported)

Fixed In

- Python **2.7.13** (2016-12-17) fixed by commit a876027 (2016-11-08)
- Python **3.4.6** (2017-01-17) fixed by commit 07bcf05 (2016-11-08)
- Python **3.5.3** (2017-01-17) fixed by commit 07bcf05 (2016-11-08)
- Python **3.6.0** (2016-12-23) fixed by commit 07bcf05 (2016-11-08)

Vulnerable Versions

- Python **3.3**

Python issue

Arbitrary code execution in gettext.c2py.

- Python issue: issue #28563
- Creation date: 2016-10-30
- Reporter: Carl Ekerot

Timeline

Timeline using the disclosure date **2016-10-30** as reference:

- 2016-10-30: Python issue #28563 reported by Carl Ekerot
- 2016-11-08 (**+9 days**): commit 07bcf05
- 2016-11-08 (**+9 days**): commit a876027
- 2016-12-17 (**+48 days**): Python 2.7.13 released
- 2016-12-23: Python 3.6.0 released
- 2017-01-17 (**+79 days**): Python 3.4.6 released
- 2017-01-17 (**+79 days**): Python 3.5.3 released

CVE-2016-2183: Sweet32 attack (DES, 3DES)

Remove 3DES from ssl default cipher list.

Sweet32 vulnerability found by Karthik Bhargavan and Gaetan Leurent from the INRIA.

- Disclosure date: **2016-08-24** (end of the Sweet32 embargo)
- Reported by: Karthik Bhargavan and Gaetan Leurent (Sweet32)

Fixed In

- Python **2.7.13** (2016-12-17) fixed by commit d988f42 (2016-09-06)
- Python **3.5.3** (2017-01-17) fixed by commit 03d13c0 (2016-09-06)
- Python **3.6.0** (2016-12-23) fixed by commit 03d13c0 (2016-09-06)

Vulnerable Versions

- Python **3.4**

Python issue

Remove 3DES from cipher list (sweet32 CVE-2016-2183).

- Python issue: issue #27850
- Creation date: 2016-08-24
- Reporter: Christian Heimes

CVE-2016-2183

The DES and Triple DES ciphers, as used in the TLS, SSH, and IPSec protocols and other protocols and products, have a birthday bound of approximately four billion blocks, which makes it easier for remote attackers to obtain cleartext data via a birthday attack against a long-duration encrypted session, as demonstrated by an HTTPS session using Triple DES in CBC mode, aka a "Sweet32" attack.

- CVE ID: CVE-2016-2183
- Published: 2016-09-01
- CVSS Score: 5.0

Timeline

Timeline using the disclosure date **2016-08-24** as reference:

- 2016-08-24: Disclosure date (end of the Sweet32 embargo)
- 2016-08-24 (**+0 days**): Python issue #27850 reported by Christian Heimes
- 2016-09-01 (**+8 days**): CVE-2016-2183 published
- 2016-09-06 (**+13 days**): commit 03d13c0
- 2016-09-06 (**+13 days**): commit d988f42

- 2016-12-17 (**+115 days**): Python 2.7.13 released
- 2016-12-23: Python 3.6.0 released
- 2017-01-17 (**+146 days**): Python 3.5.3 released

Links

- https://sweet32.info/
- https://www.openssl.org/blog/blog/2016/08/24/sweet32/

CVE-2016-1000110: HTTPoxy attack

It was discovered that the Python `CGIHandler` class did not properly protect against the `HTTP_PROXY` variable name clash in a CGI context.

A remote attacker could possibly use this flaw to redirect HTTP requests performed by a Python CGI script to an attacker-controlled proxy via a malicious HTTP request.

Ignore the `HTTP_PROXY` variable when `REQUEST_METHOD` environment is set, which indicates that the script is in CGI mode.

CVSS score: 5.0 (CVSS v3).

- Disclosure date: **2016-07-18** (Python issue #27568 reported)
- Reported by: Scott Geary (HTTPoxy)

Fixed In

- Python **2.7.13** (2016-12-17) fixed by commit 75d7b61 (2016-07-30)
- Python **3.4.6** (2017-01-17) fixed by commit 4cbb23f (2016-07-31)
- Python **3.5.3** (2017-01-17) fixed by commit 4cbb23f (2016-07-31)
- Python **3.6.0** (2016-12-23) fixed by commit 4cbb23f (2016-07-31)

Vulnerable Versions

- Python **3.3**

Python issue

"HTTPoxy", use of HTTP_PROXY flag supplied by attacker in CGI scripts.

- Python issue: issue #27568
- Creation date: 2016-07-18
- Reporter: Rémi Rampin

Timeline

Timeline using the disclosure date **2016-07-18** as reference:

- 2016-07-18: Python issue #27568 reported by Rémi Rampin
- 2016-07-30 (**+12 days**): commit 75d7b61
- 2016-07-31 (**+13 days**): commit 4cbb23f
- 2016-12-17 (**+152 days**): Python 2.7.13 released
- 2016-12-23: Python 3.6.0 released
- 2017-01-17 (**+183 days**): Python 3.4.6 released
- 2017-01-17 (**+183 days**): Python 3.5.3 released

Links

- https://httpoxy.org/
- https://access.redhat.com/security/cve/cve-2016-1000110
- https://www.cvedetails.com/cve/CVE-2016-1000110/

CVE-2016-0772: smtplib TLS stripping

A vulnerability in smtplib allowing MITM attacker to perform a startTLS stripping attack. smtplib does not seem to raise an exception when the remote end (SMTP server) is capable of negotiating starttls but fails to respond with 220 (ok) to an explicit call of SMTP.starttls(). This may allow a malicious MITM to perform a startTLS stripping attack if the client code does not explicitly check the response code for startTLS.

- Disclosure date: **2016-06-11** (commit date)
- Reported at: 2016-02-01 (Red Hat issue reported)
- Reported by: Tin (Team Oststrom)

Fixed In

- Python **2.7.12** (2016-06-28) fixed by commit 2e1b7fc (2016-06-11)
- Python **3.4.5** (2016-06-27) fixed by commit 46b32f3 (2016-06-11)
- Python **3.5.2** (2016-06-27) fixed by commit 46b32f3 (2016-06-11)
- Python **3.6.0** (2016-12-23) fixed by commit 46b32f3 (2016-06-11)

Vulnerable Versions

- Python **3.3**

CVE-2016-0772

The smtplib library in CPython (aka Python) before 2.7.12, 3.x before 3.4.5, and 3.5.x before 3.5.2 does not return an error when StartTLS fails, which might allow man-in-the-middle attackers to bypass the TLS protections by leveraging a network position between the client and the registry to block the StartTLS command, aka a "StartTLS stripping attack."

- CVE ID: CVE-2016-0772
- Published: 2016-09-02
- CVSS Score: 5.8

Timeline

Timeline using the disclosure date **2016-06-11** as reference:

- 2016-02-01 (**-131 days**): Reported (Red Hat issue reported)
- 2016-06-11: Disclosure date (commit date)
- 2016-06-11 (**+0 days**): commit 2e1b7fc
- 2016-06-11 (**+0 days**): commit 46b32f3
- 2016-06-27 (**+16 days**): Python 3.4.5 released
- 2016-06-27 (**+16 days**): Python 3.5.2 released
- 2016-06-28 (**+17 days**): Python 2.7.12 released
- 2016-09-02 (**+83 days**): CVE-2016-0772 published
- 2016-12-23: Python 3.6.0 released

Links

- http://seclists.org/oss-sec/2016/q2/541
- https://bugzilla.redhat.com/show_bug.cgi?id=CVE-2016-0772

Issue #26657: HTTP directory traversal

Fix directory traversal vulnerability with `http.server` and `SimpleHTTPServer` on Windows.

Regression of Python 3.3.5.

Python issue reported at 2016-03-14.

- Disclosure date: **2016-03-28** (Python issue #26657 reported)

Fixed In

- Python **2.7.12** (2016-06-28) fixed by commit 0cf2cf2 (2016-04-18)
- Python **3.5.2** (2016-06-27) fixed by commit d274b3f (2016-04-18)
- Python **3.6.0** (2016-12-23) fixed by commit d274b3f (2016-04-18)

Vulnerable Versions

- Python **3.3**
- Python **3.4**

Python issue

Directory traversal with http.server and SimpleHTTPServer on windows.

- Python issue: issue #26657
- Creation date: 2016-03-28
- Reporter: Thomas

Timeline

Timeline using the disclosure date **2016-03-28** as reference:

- 2016-03-28: Python issue #26657 reported by Thomas
- 2016-04-18 (**+21 days**): commit 0cf2cf2
- 2016-04-18 (**+21 days**): commit d274b3f
- 2016-06-27 (**+91 days**): Python 3.5.2 released
- 2016-06-28 (**+92 days**): Python 2.7.12 released
- 2016-12-23: Python 3.6.0 released

Issue #26556: Expat 2.1.1

Multiple integer overflows have been discovered in Expat, an XML parsing C library, which may result in denial of service or the execution of arbitrary code if a malformed XML file is processed.

Update bundled copy of Expat library to version 2.1.1 to get CVE-2015-1283 fixes.

- Disclosure date: **2016-03-14** (Python issue #26556 reported)
- Reported at: 2015-07-24 (Expat issue #528 reported)
- Reported by: David Dillard (Expat issue)

Fixed In

- Python **2.7.12** (2016-06-28) fixed by commit d244a8f (2016-06-11)
- Python **3.4.5** (2016-06-27) fixed by commit 196d7db (2016-06-11)
- Python **3.5.2** (2016-06-27) fixed by commit 196d7db (2016-06-11)
- Python **3.6.0** (2016-12-23) fixed by commit 196d7db (2016-06-11)

Vulnerable Versions

- Python **3.3**

Python issue

Update expat to 2.1.1.

- Python issue: issue #26556
- Creation date: 2016-03-14
- Reporter: Christian Heimes

Timeline

Timeline using the disclosure date **2016-03-14** as reference:

- 2015-07-24 (**-234 days**): Reported (Expat issue #528 reported)
- 2016-03-14: Python issue #26556 reported by Christian Heimes
- 2016-06-11 (**+89 days**): commit 196d7db
- 2016-06-11 (**+89 days**): commit d244a8f
- 2016-06-27 (**+105 days**): Python 3.4.5 released
- 2016-06-27 (**+105 days**): Python 3.5.2 released
- 2016-06-28 (**+106 days**): Python 2.7.12 released
- 2016-12-23: Python 3.6.0 released

Links

- https://sourceforge.net/p/expat/bugs/528/

CVE-2016-5636: zipimporter overflow

Heap overflow in `zipimporter` module.

- Disclosure date: **2016-01-21** (Python issue #26171 reported)

Fixed In

- Python **2.7.12** (2016-06-28) fixed by commit 64ea192 (2016-01-21)
- Python **3.4.5** (2016-06-27) fixed by commit c4032da (2016-01-21)
- Python **3.5.2** (2016-06-27) fixed by commit c4032da (2016-01-21)
- Python **3.6.0** (2016-12-23) fixed by commit d751040 (2016-09-14)

Vulnerable Versions

- Python **3.3**

Python issue

heap overflow in zipimporter module.

- Python issue: issue #26171
- Creation date: 2016-01-21
- Reporter: Insu Yun

CVE-2016-5636

Integer overflow in the get_data function in zipimport.c in CPython (aka Python) before 2.7.12, 3.x before 3.4.5, and 3.5.x before 3.5.2 allows remote attackers to have unspecified impact via a negative data size value, which triggers a heap-based buffer overflow.

- CVE ID: CVE-2016-5636
- Published: 2016-09-02
- CVSS Score: 10.0

Timeline

Timeline using the disclosure date **2016-01-21** as reference:

- 2016-01-21: Python issue #26171 reported by Insu Yun
- 2016-01-21 (**+0 days**): commit 64ea192
- 2016-01-21 (**+0 days**): commit c4032da
- 2016-06-27 (**+158 days**): Python 3.4.5 released
- 2016-06-27 (**+158 days**): Python 3.5.2 released
- 2016-06-28 (**+159 days**): Python 2.7.12 released
- 2016-09-02 (**+225 days**): CVE-2016-5636 published
- 2016-09-14 (**+237 days**): commit d751040
- 2016-12-23: Python 3.6.0 released

CVE-2016-5699: HTTP header injection

HTTP header injection in `urllib`, `urrlib2`, `httplib` and `http.client` modules.

CRLF injection vulnerability in the `HTTPConnection.putheader()` function in `urllib2` and `urllib` in CPython before 2.7.10 and 3.x before 3.4.4 allows remote attackers to inject arbitrary HTTP headers via CRLF sequences in a URL.

Reported again in January 2016 by Timothy D. Morgan (Blindspot Security), with a full disclosed at 2016-06-15.

- Disclosure date: **2014-11-24** (Python issue #22928 reported)
- Red Hat impact: Moderate

Fixed In

- Python **2.7.10** (2015-05-23) fixed by commit 59bdf63 (2015-03-12)
- Python **3.4.4** (2015-12-21) fixed by commit a112a8a (2015-03-12)
- Python **3.5.0** (2015-09-09) fixed by commit a112a8a (2015-03-12)

Vulnerable Versions

- Python **3.3**

Python issue

HTTP header injection in urrlib2/urllib/httplib/http.client (CVE-2016-5699).

- Python issue: issue #22928
- Creation date: 2014-11-24
- Reporter: Guido Vranken

CVE-2016-5699

CRLF injection vulnerability in the HTTPConnection.putheader function in urllib2 and urllib in CPython (aka Python) before 2.7.10 and 3.x before 3.4.4 allows remote attackers to inject arbitrary HTTP headers via CRLF sequences in a URL.

- CVE ID: CVE-2016-5699
- Published: 2016-09-02
- CVSS Score: 4.3

Timeline

Timeline using the disclosure date **2014-11-24** as reference:

- 2014-11-24: Python issue #22928 reported by Guido Vranken
- 2015-03-12 (**+108 days**): commit 59bdf63
- 2015-03-12 (**+108 days**): commit a112a8a
- 2015-05-23 (**+180 days**): Python 2.7.10 released
- 2015-09-09: Python 3.5.0 released
- 2015-12-21 (**+392 days**): Python 3.4.4 released
- 2016-09-02 (**+648 days**): CVE-2016-5699 published

Links

- http://blog.blindspotsecurity.com/2016/06/advisory-http-header-injection-in.html

CVE-2014-9365: Validate TLS certificate

The HTTP clients in the (1) httplib, (2) urllib, (3) urllib2, and (4) xmlrpclib libraries in CPython (aka Python) 2.x before 2.7.9 and 3.x before 3.4.3, when accessing an HTTPS URL, do not (a) check the certificate against a trust store or verify that the server hostname matches a domain name in the subject's (b) Common Name or (c) subjectAltName field of the X.509 certificate, which allows man-in-the-middle attackers to spoof SSL servers via an arbitrary valid certificate.

See also the PEP 466: Network Security Enhancements for Python 2.7.x.

- Disclosure date: **2014-08-28** (PEP 476 created)
- Reported by: Alex Gaynor (PEP 476 author)

Fixed In

- Python **2.7.9** (2014-12-10) fixed by commit e3e7d40 (2014-11-24)
- Python **3.4.3** (2015-02-23) fixed by commit 4ffb075 (2014-11-03)
- Python **3.5.0** (2015-09-09) fixed by commit 4ffb075 (2014-11-03)

Vulnerable Versions

- Python **3.3**

Python issue

PEP 476: verify HTTPS certificates by default.

- Python issue: issue #22417
- Creation date: 2014-09-15
- Reporter: Nick Coghlan

CVE-2014-9365

The HTTP clients in the (1) httplib, (2) urllib, (3) urllib2, and (4) xmlrpclib libraries in CPython (aka Python) 2.x before 2.7.9 and 3.x before 3.4.3, when accessing an HTTPS URL, do not (a) check the certificate against a trust store or verify that the server hostname matches a domain name in the subject's (b) Common Name or (c) subjectAltName field of the X.509 certificate, which allows man-in-the-middle attackers to spoof SSL servers via an arbitrary valid certificate.

- CVE ID: CVE-2014-9365
- Published: 2014-12-12
- CVSS Score: 5.8

Timeline

Timeline using the disclosure date **2014-08-28** as reference:

- 2014-08-28: Disclosure date (PEP 476 created)
- 2014-09-15 (**+18 days**): Python issue #22417 reported by Nick Coghlan

- 2014-11-03 (**+67 days**): commit 4ffb075
- 2014-11-24 (**+88 days**): commit e3e7d40
- 2014-12-10 (**+104 days**): Python 2.7.9 released
- 2014-12-12 (**+106 days**): CVE-2014-9365 published
- 2015-02-23 (**+179 days**): Python 3.4.3 released
- 2015-09-09: Python 3.5.0 released

Links

- PEP 476: Enabling certificate verification by default for stdlib http clients

CVE-2014-7185: buffer() integer overflows

Integer overflow in `bufferobject.c` in Python before 2.7.8 allows context-dependent attackers to obtain sensitive information from process memory via a large size and offset in a `buffer` type.

- Disclosure date: **2014-06-24** (Python issue #21831 reported)
- Reported by: Chris Foster (on the Python security list)

Fixed In

- Python **2.7.8** (2014-06-29) fixed by commit 550b945 (2014-06-24)

Python issue

integer overflow in 'buffer' type allows reading memory.

- Python issue: issue #21831
- Creation date: 2014-06-24
- Reporter: Benjamin Peterson

CVE-2014-7185

Integer overflow in bufferobject.c in Python before 2.7.8 allows context-dependent attackers to obtain sensitive information from process memory via a large size and offset in a "buffer" function.

- CVE ID: CVE-2014-7185
- Published: 2014-10-08
- CVSS Score: 6.4

Timeline

Timeline using the disclosure date **2014-06-24** as reference:

- 2014-06-24: Python issue #21831 reported by Benjamin Peterson
- 2014-06-24 (**+0 days**): commit 550b945

- 2014-06-29 (**+5 days**): Python 2.7.8 released
- 2014-10-08 (**+106 days**): CVE-2014-7185 published

CVE-2014-4616: JSONDecoder.raw_decode

Fix arbitrary memory access in `JSONDecoder.raw_decode()` with a negative second parameter.

Note: The issue #21529 was created at 2014-05-19, after the commit.

- Disclosure date: **2014-04-13** (commit)
- Reported by: Guido Vranken
- Red Hat impact: Moderate

Fixed In

- Python **2.7.7** (2014-05-31) fixed by commit 6c939cb (2014-04-14)
- Python **3.2.6** (2014-10-11) fixed by commit 99b5afa (2014-04-14)
- Python **3.3.6** (2014-10-11) fixed by commit 99b5afa (2014-04-14)
- Python **3.4.1** (2014-05-18) fixed by commit 99b5afa (2014-04-14)
- Python **3.5.0** (2015-09-09) fixed by commit 99b5afa (2014-04-14)

Python issue

JSON module: reading arbitrary process memory.

- Python issue: issue #21529
- Creation date: 2014-05-19
- Reporter: Benjamin Peterson

Timeline

Timeline using the disclosure date **2014-04-13** as reference:

- 2014-04-13: Disclosure date (commit)
- 2014-04-14 (**+1 days**): commit 6c939cb
- 2014-04-14 (**+1 days**): commit 99b5afa
- 2014-05-18 (**+35 days**): Python 3.4.1 released
- 2014-05-19 (**+36 days**): Python issue #21529 reported by Benjamin Peterson
- 2014-05-31 (**+48 days**): Python 2.7.7 released
- 2014-10-11 (**+181 days**): Python 3.2.6 released
- 2014-10-11 (**+181 days**): Python 3.3.6 released
- 2015-09-09: Python 3.5.0 released

Links

- https://access.redhat.com/security/cve/cve-2014-4616
- https://www.cvedetails.com/cve/CVE-2014-4616/

CVE-2014-2667: os.makedirs() not thread-safe

`os.makedirs(exist_ok=True)` is not thread-safe: umask is set temporary to `0`, serious security problem.

The fix removes the directory mode check from `os.makedirs()`.

The `exist_ok` parameter was added to Python 3.2.0 (commit 5a22b651173f142a600625a036fcf36484ade237).

- Disclosure date: **2014-03-28** (Python issue #21082 reported)

Fixed In

- Python **3.2.6** (2014-10-11) fixed by commit ee5f1c1 (2014-04-01)
- Python **3.3.6** (2014-10-11) fixed by commit ee5f1c1 (2014-04-01)
- Python **3.4.1** (2014-05-18) fixed by commit ee5f1c1 (2014-04-01)
- Python **3.5.0** (2015-09-09) fixed by commit ee5f1c1 (2014-04-01)

Python issue

os.makedirs(exist_ok=True) is not thread-safe: umask is set temporary to 0, serious security problem.

- Python issue: issue #21082
- Creation date: 2014-03-28
- Reporter: Ryan Lortie

CVE-2014-2667

Race condition in the _get_masked_mode function in Lib/os.py in Python 3.2 through 3.5, when exist_ok is set to true and multiple threads are used, might allow local users to bypass intended file permissions by leveraging a separate application vulnerability before the umask has been set to the expected value.

- CVE ID: CVE-2014-2667
- Published: 2014-11-16
- CVSS Score: 3.3

Timeline

Timeline using the disclosure date **2014-03-28** as reference:

- 2014-03-28: Python issue #21082 reported by Ryan Lortie
- 2014-04-01 (**+4 days**): commit ee5f1c1
- 2014-05-18 (**+51 days**): Python 3.4.1 released
- 2014-10-11 (**+197 days**): Python 3.2.6 released

- 2014-10-11 (**+197 days**): Python 3.3.6 released
- 2014-11-16 (**+233 days**): CVE-2014-2667 published
- 2015-09-09: Python 3.5.0 released

CVE-2014-1912: socket.recvfrom_into() overflow

`socket.recvfrom_into()` fails to check that the supplied buffer object is big enough for the requested read and so will happily write off the end.

- Disclosure date: **2014-01-14** (Python issue #20246 reported)

Fixed In

- Python **2.7.7** (2014-05-31) fixed by commit 28cf368 (2014-01-14)
- Python **3.2.6** (2014-10-11) fixed by commit fbf648e (2014-01-14)
- Python **3.3.4** (2014-02-09) fixed by commit fbf648e (2014-01-14)
- Python **3.4.0** (2014-03-16) fixed by commit fbf648e (2014-01-14)

Python issue

buffer overflow in socket.recvfrom_into.

- Python issue: issue #20246
- Creation date: 2014-01-14
- Reporter: Ryan Smith-Roberts

CVE-2014-1912

Buffer overflow in the socket.recvfrom_into function in Modules/socketmodule.c in Python 2.5 before 2.7.7, 3.x before 3.3.4, and 3.4.x before 3.4rc1 allows remote attackers to execute arbitrary code via a crafted string.

- CVE ID: CVE-2014-1912
- Published: 2014-03-01
- CVSS Score: 7.5

Timeline

Timeline using the disclosure date **2014-01-14** as reference:

- 2014-01-14: Python issue #20246 reported by Ryan Smith-Roberts
- 2014-01-14 (**+0 days**): commit 28cf368
- 2014-01-14 (**+0 days**): commit fbf648e
- 2014-02-09 (**+26 days**): Python 3.3.4 released
- 2014-03-01 (**+46 days**): CVE-2014-1912 published
- 2014-03-16: Python 3.4.0 released

- 2014-05-31 (**+137 days**): Python 2.7.7 released
- 2014-10-11 (**+270 days**): Python 3.2.6 released

CVE-2013-7338: zipfile DoS using malformed file

Python before 3.3.4 RC1 allows remote attackers to cause a denial of service (infinite loop and CPU consumption) via a file size value larger than the size of the zip file to the functions:

- `ZipExtFile.read()`
- `ZipExtFile.readlines()`
- `ZipFile.extract()`
- `ZipFile.extractall()`

Reading malformed zipfiles no longer hangs with 100% CPU consumption.

Python 2.7 is not affected.

- Disclosure date: **2013-12-27** (Python issue #20078 reported)

Fixed In

- Python **3.3.4** (2014-02-09) fixed by commit 5ce3f10 (2014-01-09)
- Python **3.4.0** (2014-03-16) fixed by commit 5ce3f10 (2014-01-09)

Python issue

zipfile - ZipExtFile.read goes into 100% CPU infinite loop on maliciously binary edited zips.

- Python issue: issue #20078
- Creation date: 2013-12-27
- Reporter: Nandiya

CVE-2013-7338

Python before 3.3.4 RC1 allows remote attackers to cause a denial of service (infinite loop and CPU consumption) via a file size value larger than the size of the zip file to the (1) ZipExtFile.read, (2) ZipExtFile.read(n), (3) ZipExtFile.readlines, (4) ZipFile.extract, or (5) ZipFile.extractall function.

- CVE ID: CVE-2013-7338
- Published: 2014-04-22
- CVSS Score: 7.1

Timeline

Timeline using the disclosure date **2013-12-27** as reference:

- 2013-12-27: Python issue #20078 reported by Nandiya
- 2014-01-09 (**+13 days**): commit 5ce3f10

- 2014-02-09 (**+44 days**): Python 3.3.4 released
- 2014-03-16: Python 3.4.0 released
- 2014-04-22 (**+116 days**): CVE-2013-7338 published

Issue #19435: CGI directory traversal

An error in separating the path and filename of the CGI script to run in `http.server.CGIHTTPRequestHandler` allows running arbitrary executables in the directory under which the server was started.

- Disclosure date: **2013-10-29** (Python issue #19435 reported)

Fixed In

- Python **2.7.6** (2013-11-10) fixed by commit 1ef959a (2013-10-30)
- Python **3.2.6** (2014-10-11) fixed by commit 04e9de4 (2013-10-30)
- Python **3.3.4** (2014-02-09) fixed by commit 04e9de4 (2013-10-30)
- Python **3.4.0** (2014-03-16) fixed by commit 04e9de4 (2013-10-30)

Python issue

Directory traversal attack for CGIHTTPRequestHandler.

- Python issue: issue #19435
- Creation date: 2013-10-29
- Reporter: Alexander Kruppa

Timeline

Timeline using the disclosure date **2013-10-29** as reference:

- 2013-10-29: Python issue #19435 reported by Alexander Kruppa
- 2013-10-30 (**+1 days**): commit 04e9de4
- 2013-10-30 (**+1 days**): commit 1ef959a
- 2013-11-10 (**+12 days**): Python 2.7.6 released
- 2014-02-09 (**+103 days**): Python 3.3.4 released
- 2014-03-16: Python 3.4.0 released
- 2014-10-11 (**+347 days**): Python 3.2.6 released

CVE-2013-4238: ssl: NUL in subjectAltNames

SSL module fails to handle NULL bytes inside subjectAltNames general names.

It's related to Ruby's CVE-2013-4073.

Issue #18709 reported by Christian Heimes at 2013-08-12.

- Disclosure date: **2013-06-27** (Ruby issue)
- Reported by: Ryan Sleevi of the Google Chrome Security Team

Fixed In

- Python **2.6.9** (2013-10-29) fixed by commit 82f8828 (2013-08-23)
- Python **2.7.6** (2013-11-10) fixed by commit 82f8828 (2013-08-23)
- Python **3.2.6** (2014-10-11) fixed by commit ec3c103 (2014-09-30)
- Python **3.3.3** (2013-11-17) fixed by commit 824f7f3 (2013-08-16)
- Python **3.4.0** (2014-03-16) fixed by commit 824f7f3 (2013-08-16)

Python issue

SSL module fails to handle NULL bytes inside subjectAltNames general names (CVE-2013-4238).

- Python issue: issue #18709
- Creation date: 2013-08-12
- Reporter: Christian Heimes

CVE-2013-4238

The ssl.match_hostname function in the SSL module in Python 2.6 through 3.4 does not properly handle a '0' character in a domain name in the Subject Alternative Name field of an X.509 certificate, which allows man-in-the-middle attackers to spoof arbitrary SSL servers via a crafted certificate issued by a legitimate Certification Authority, a related issue to CVE-2009-2408.

- CVE ID: CVE-2013-4238
- Published: 2013-08-18
- CVSS Score: 4.3

Timeline

Timeline using the disclosure date **2013-06-27** as reference:

- 2013-06-27: Disclosure date (Ruby issue)
- 2013-08-12 (**+46 days**): Python issue #18709 reported by Christian Heimes
- 2013-08-16 (**+50 days**): commit 824f7f3
- 2013-08-18 (**+52 days**): CVE-2013-4238 published
- 2013-08-23 (**+57 days**): commit 82f8828
- 2013-10-29 (**+124 days**): Python 2.6.9 released
- 2013-11-10 (**+136 days**): Python 2.7.6 released
- 2013-11-17 (**+143 days**): Python 3.3.3 released
- 2014-03-16: Python 3.4.0 released
- 2014-09-30 (**+460 days**): commit ec3c103

- 2014-10-11 (**+471 days**): Python 3.2.6 released

CVE-2013-7440: ssl.match_hostname() IDNA issue

`ssl.match_hostname()`: sub string wildcard should not match IDNA prefix.

Change behavior of `ssl.match_hostname()` to follow RFC 6125, for security reasons. It now doesn't match multiple wildcards nor wildcards inside IDN fragments.

- Disclosure date: **2013-05-17** (Python issue #17997 reported)

Fixed In

- Python **3.3.3** (2013-11-17) fixed by commit 72c98d3 (2013-10-27)
- Python **3.4.0** (2014-03-16) fixed by commit 72c98d3 (2013-10-27)

Python issue

ssl.match_hostname(): sub string wildcard should not match IDNA prefix.

- Python issue: issue #17997
- Creation date: 2013-05-17
- Reporter: Christian Heimes

CVE-2013-7440

The ssl.match_hostname function in CPython (aka Python) before 2.7.9 and 3.x before 3.3.3 does not properly handle wildcards in hostnames, which might allow man-in-the-middle attackers to spoof servers via a crafted certificate.

- CVE ID: CVE-2013-7440
- Published: 2016-06-07
- CVSS Score: 4.3

Timeline

Timeline using the disclosure date **2013-05-17** as reference:

- 2013-05-17: Python issue #17997 reported by Christian Heimes
- 2013-10-27 (**+163 days**): commit 72c98d3
- 2013-11-17 (**+184 days**): Python 3.3.3 released
- 2014-03-16: Python 3.4.0 released
- 2016-06-07 (**+1117 days**): CVE-2013-7440 published

Links

- https://tools.ietf.org/html/rfc6125

CVE-2013-2099: ssl.match_hostname() wildcard DoS

If the name in the certificate contains many * characters (wildcard), matching the compiled regular expression against the host name can take a very long time.

Certificate validation happens before host name checking, so I think this is a minor issue only because it can only be triggered in cooperation with a CA (which seems unlikely).

- Disclosure date: **2013-05-15** (Python issue #17980 reported)

Fixed In

- Python **3.2.6** (2014-10-11) fixed by commit 86d53ca (2013-05-18)
- Python **3.3.3** (2013-11-17) fixed by commit 86d53ca (2013-05-18)
- Python **3.4.0** (2014-03-16) fixed by commit 86d53ca (2013-05-18)

Python issue

CVE-2013-2099 ssl.match_hostname() trips over crafted wildcard names.

- Python issue: issue #17980
- Creation date: 2013-05-15
- Reporter: Florian Weimer

CVE-2013-2099

Algorithmic complexity vulnerability in the ssl.match_hostname function in Python 3.2.x, 3.3.x, and earlier, and unspecified versions of python-backports-ssl_match_hostname as used for older Python versions, allows remote attackers to cause a denial of service (CPU consumption) via multiple wildcard characters in the common name in a certificate.

- CVE ID: CVE-2013-2099
- Published: 2013-10-09
- CVSS Score: 4.3

Timeline

Timeline using the disclosure date **2013-05-15** as reference:

- 2013-05-15: Python issue #17980 reported by Florian Weimer
- 2013-05-18 (**+3 days**): commit 86d53ca
- 2013-10-09 (**+147 days**): CVE-2013-2099 published
- 2013-11-17 (**+186 days**): Python 3.3.3 released
- 2014-03-16: Python 3.4.0 released
- 2014-10-11 (**+514 days**): Python 3.2.6 released

CVE-2013-1752: ftplib unlimited read

ftplib: unlimited read from connection.

- Disclosure date: **2012-09-25** (Python issue #16038 reported)
- Red Hat impact: Moderate

Fixed In

- Python **2.7.6** (2013-11-10) fixed by commit 2585e1e (2013-10-20)
- Python **3.2.6** (2014-10-11) fixed by commit c9cb18d (2014-09-30)
- Python **3.3.3** (2013-11-17) fixed by commit c30b178 (2013-10-20)
- Python **3.4.0** (2014-03-16) fixed by commit c30b178 (2013-10-20)

Python issue

ftplib: unlimited readline() from connection.

- Python issue: issue #16038
- Creation date: 2012-09-25
- Reporter: Christian Heimes

Timeline

Timeline using the disclosure date **2012-09-25** as reference:

- 2012-09-25: Python issue #16038 reported by Christian Heimes
- 2013-10-20 (**+390 days**): commit 2585e1e
- 2013-10-20 (**+390 days**): commit c30b178
- 2013-11-10 (**+411 days**): Python 2.7.6 released
- 2013-11-17 (**+418 days**): Python 3.3.3 released
- 2014-03-16: Python 3.4.0 released
- 2014-09-30 (**+735 days**): commit c9cb18d
- 2014-10-11 (**+746 days**): Python 3.2.6 released

Links

- https://access.redhat.com/security/cve/cve-2013-1752
- https://www.cvedetails.com/cve/CVE-2013-1752/

CVE-2013-1752: nntplib unlimited read

Unlimited read from connection in nntplib.

- Disclosure date: **2012-09-25** (Python issue #16040 reported)
- Red Hat impact: Moderate

Fixed In

- Python **2.6.9** (2013-10-29) fixed by commit 42faa55 (2013-09-30)
- Python **2.7.6** (2013-11-10) fixed by commit 42faa55 (2013-09-30)
- Python **3.2.6** (2014-10-11) fixed by commit b3ac843 (2014-10-12)
- Python **3.4.3** (2015-02-23) fixed by commit b3ac843 (2014-10-12)
- Python **3.5.0** (2015-09-09) fixed by commit b3ac843 (2014-10-12)

Python issue

nntplib: unlimited readline() from connection.

- Python issue: issue #16040
- Creation date: 2012-09-25
- Reporter: Christian Heimes

Timeline

Timeline using the disclosure date **2012-09-25** as reference:

- 2012-09-25: Python issue #16040 reported by Christian Heimes
- 2013-09-30 (**+370 days**): commit 42faa55
- 2013-10-29 (**+399 days**): Python 2.6.9 released
- 2013-11-10 (**+411 days**): Python 2.7.6 released
- 2014-10-11 (**+746 days**): Python 3.2.6 released
- 2014-10-12 (**+747 days**): commit b3ac843
- 2015-02-23 (**+881 days**): Python 3.4.3 released
- 2015-09-09: Python 3.5.0 released

Links

- https://access.redhat.com/security/cve/cve-2013-1752
- https://www.cvedetails.com/cve/CVE-2013-1752/

CVE-2013-1752: poplib unlimited read

poplib: unlimited read from connection.

- Disclosure date: **2012-09-25** (Python issue #16041 reported)
- Red Hat impact: Moderate

Fixed In

- Python **2.7.9** (2014-12-10) fixed by commit faad6bb (2014-12-06)
- Python **3.2.6** (2014-10-11) fixed by commit eaca861 (2014-09-30)
- Python **3.4.3** (2015-02-23) fixed by commit eaca861 (2014-09-30)
- Python **3.5.0** (2015-09-09) fixed by commit eaca861 (2014-09-30)

Python issue

poplib: unlimited readline() from connection.

- Python issue: issue #16041
- Creation date: 2012-09-25
- Reporter: Christian Heimes

Timeline

Timeline using the disclosure date **2012-09-25** as reference:

- 2012-09-25: Python issue #16041 reported by Christian Heimes
- 2014-09-30 (**+735 days**): commit eaca861
- 2014-10-11 (**+746 days**): Python 3.2.6 released
- 2014-12-06 (**+802 days**): commit faad6bb
- 2014-12-10 (**+806 days**): Python 2.7.9 released
- 2015-02-23 (**+881 days**): Python 3.4.3 released
- 2015-09-09: Python 3.5.0 released

Links

- https://access.redhat.com/security/cve/cve-2013-1752
- https://www.cvedetails.com/cve/CVE-2013-1752/

CVE-2013-1752: smtplib unlimited read

The smtplib module doesn't limit the amount of read data in its call to readline(). An erroneous or malicious SMTP server can trick the smtplib module to consume large amounts of memory.

- Disclosure date: **2012-09-25** (Python issue #16042 reported)

- Red Hat impact: Moderate

Fixed In

- Python **2.7.9** (2014-12-10) fixed by commit dabfc56 (2014-12-06)
- Python **3.2.6** (2014-10-11) fixed by commit 210ee47 (2014-09-30)
- Python **3.4.3** (2015-02-23) fixed by commit 210ee47 (2014-09-30)
- Python **3.5.0** (2015-09-09) fixed by commit 210ee47 (2014-09-30)

Python issue

smtplib: unlimited readline() from connection.

- Python issue: issue #16042
- Creation date: 2012-09-25
- Reporter: Christian Heimes

Timeline

Timeline using the disclosure date **2012-09-25** as reference:

- 2012-09-25: Python issue #16042 reported by Christian Heimes
- 2014-09-30 (**+735 days**): commit 210ee47
- 2014-10-11 (**+746 days**): Python 3.2.6 released
- 2014-12-06 (**+802 days**): commit dabfc56
- 2014-12-10 (**+806 days**): Python 2.7.9 released
- 2015-02-23 (**+881 days**): Python 3.4.3 released
- 2015-09-09: Python 3.5.0 released

Links

- https://access.redhat.com/security/cve/cve-2013-1752
- https://www.cvedetails.com/cve/CVE-2013-1752/

CVE-2013-1753: xmlrpc gzip unlimited read

Add a default limit for the amount of data `xmlrpclib.gzip_decode()` will return.

- Disclosure date: **2012-09-25** (Python issue #16043 reported)
- Red Hat impact: Moderate

Fixed In

- Python **2.7.9** (2014-12-10) fixed by commit 9e8f523 (2014-12-06)
- Python **3.4.3** (2015-02-23) fixed by commit 4e9cefa (2014-12-06)
- Python **3.5.0** (2015-09-09) fixed by commit 4e9cefa (2014-12-06)

Vulnerable Versions

- Python **3.3**

Python issue

xmlrpc: gzip_decode has unlimited read().

- Python issue: issue #16043
- Creation date: 2012-09-25
- Reporter: Christian Heimes

Timeline

Timeline using the disclosure date **2012-09-25** as reference:

- 2012-09-25: Python issue #16043 reported by Christian Heimes
- 2014-12-06 (**+802 days**): commit 4e9cefa
- 2014-12-06 (**+802 days**): commit 9e8f523
- 2014-12-10 (**+806 days**): Python 2.7.9 released
- 2015-02-23 (**+881 days**): Python 3.4.3 released
- 2015-09-09: Python 3.5.0 released

Links

- https://access.redhat.com/security/cve/cve-2013-1753
- https://www.cvedetails.com/cve/CVE-2013-1753/

CVE-2013-7040: Hash not properly randomized

Hash function is not randomized properly.

Python 3.4 now used SipHash (PEP 456).

Python 3.3 and Python 2.7 are still affected.

- Disclosure date: **2012-04-19** (Python issue #14621 reported)

Fixed In

- Python **3.4.0** (2014-03-16) fixed by commit 985ecdc (2013-11-20)

Vulnerable Versions

- Python **2.7**
- Python **3.3**

Python issue

Hash function is not randomized properly.

- Python issue: issue #14621
- Creation date: 2012-04-19
- Reporter: Vlado Boza

CVE-2013-7040

Python 2.7 before 3.4 only uses the last eight bits of the prefix to randomize hash values, which causes it to compute hash values without restricting the ability to trigger hash collisions predictably and makes it easier for context-dependent attackers to cause a denial of service (CPU consumption) via crafted input to an application that maintains a hash table. NOTE: this vulnerability exists because of an incomplete fix for CVE-2012-1150.

- CVE ID: CVE-2013-7040
- Published: 2014-05-19
- CVSS Score: 4.3

Timeline

Timeline using the disclosure date **2012-04-19** as reference:

- 2012-04-19: Python issue #14621 reported by Vlado Boza
- 2013-11-20 (**+580 days**): commit 985ecdc
- 2014-03-16 (**+696 days**): Python 3.4.0 released
- 2014-05-19 (**+760 days**): CVE-2013-7040 published

CVE-2012-2135: UTF-16 decoder

Vulnerability in the UTF-16 decoder after error handling.

- Disclosure date: **2012-04-14**

Fixed In

- Python **2.7.4** (2013-04-06) fixed by commit 715a63b (2012-07-20)
- Python **3.2.4** (2013-04-07) fixed by commit 715a63b (2012-07-20)
- Python **3.3.0** (2012-09-29) fixed by commit b4bbee2 (2012-07-20)

Python issue

CVE-2012-2135: Vulnerability in the utf-16 decoder after error handling.

- Python issue: issue #14579
- Creation date: 2012-04-14
- Reporter: Serhiy Storchaka

CVE-2012-2135

The utf-16 decoder in Python 3.1 through 3.3 does not update the aligned_end variable after calling the unicode_decode_call_errorhandler function, which allows remote attackers to obtain sensitive information (process memory) or cause a denial of service (memory corruption and crash) via unspecified vectors.

- CVE ID: CVE-2012-2135
- Published: 2012-08-14
- CVSS Score: 6.4

Timeline

Timeline using the disclosure date **2012-04-14** as reference:

- 2012-04-14: Disclosure date
- 2012-04-14 (**+0 days**): Python issue #14579 reported by Serhiy Storchaka
- 2012-07-20 (**+97 days**): commit 715a63b
- 2012-07-20 (**+97 days**): commit b4bbee2
- 2012-08-14 (**+122 days**): CVE-2012-2135 published
- 2012-09-29: Python 3.3.0 released
- 2013-04-06 (**+357 days**): Python 2.7.4 released
- 2013-04-07 (**+358 days**): Python 3.2.4 released

CVE-2012-0845: XML-RPC DoS

A denial of service flaw was found in the way Simple XML-RPC Server module of Python processed client connections, that were closed prior the complete request body has been received. A remote attacker could use this flaw to cause Python Simple XML-RPC based server process to consume excessive amount of CPU.

- Disclosure date: **2012-02-13** (Python issue #14001 reported)

Fixed In

- Python **2.6.8** (2012-04-10) fixed by commit 66f3cc6 (2012-02-18)
- Python **2.7.3** (2012-04-09) fixed by commit 66f3cc6 (2012-02-18)
- Python **3.1.5** (2012-04-08) fixed by commit ec1712a (2012-02-18)
- Python **3.2.3** (2012-04-10) fixed by commit ec1712a (2012-02-18)

- Python **3.3.0** (2012-09-29) fixed by commit ec1712a (2012-02-18)

Python issue

CVE-2012-0845 Python v2.7.2 / v3.2.2 (SimpleXMLRPCServer): DoS (excessive CPU usage) by processing malformed XMLRPC / HTTP POST request.

- Python issue: issue #14001
- Creation date: 2012-02-13
- Reporter: Jan Lieskovsky

CVE-2012-0845

SimpleXMLRPCServer.py in SimpleXMLRPCServer in Python before 2.6.8, 2.7.x before 2.7.3, 3.x before 3.1.5, and 3.2.x before 3.2.3 allows remote attackers to cause a denial of service (infinite loop and CPU consumption) via an XML-RPC POST request that contains a smaller amount of data than specified by the Content-Length header.

- CVE ID: CVE-2012-0845
- Published: 2012-10-05
- CVSS Score: 5.0

Timeline

Timeline using the disclosure date **2012-02-13** as reference:

- 2012-02-13: Python issue #14001 reported by Jan Lieskovsky
- 2012-02-18 (**+5 days**): commit 66f3cc6
- 2012-02-18 (**+5 days**): commit ec1712a
- 2012-04-08 (**+55 days**): Python 3.1.5 released
- 2012-04-09 (**+56 days**): Python 2.7.3 released
- 2012-04-10 (**+57 days**): Python 2.6.8 released
- 2012-04-10 (**+57 days**): Python 3.2.3 released
- 2012-09-29: Python 3.3.0 released
- 2012-10-05 (**+235 days**): CVE-2012-0845 published

CVE-2011-3389: ssl CBC IV attack

The ssl module would always disable the CBC IV attack countermeasure. Disable OpenSSL `SSL_OP_DONT_INSERT_EMPTY_FRAGMENTS` option.

- Disclosure date: **2012-01-27** (Python issue #13885 reported)
- Reported by: Apple security team

Fixed In

- Python **2.6.8** (2012-04-10) fixed by commit d358e05 (2012-01-27)
- Python **2.7.3** (2012-04-09) fixed by commit d358e05 (2012-01-27)
- Python **3.1.5** (2012-04-08) fixed by commit f2bf8a6 (2012-01-27)
- Python **3.2.3** (2012-04-10) fixed by commit f2bf8a6 (2012-01-27)
- Python **3.3.0** (2012-09-29) fixed by commit f2bf8a6 (2012-01-27)

Python issue

CVE-2011-3389: _ssl module always disables the CBC IV attack countermeasure.

- Python issue: issue #13885
- Creation date: 2012-01-27
- Reporter: Antoine Pitrou

CVE-2011-3389

The SSL protocol, as used in certain configurations in Microsoft Windows and Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera, and other products, encrypts data by using CBC mode with chained initialization vectors, which allows man-in-the-middle attackers to obtain plaintext HTTP headers via a blockwise chosen-boundary attack (BCBA) on an HTTPS session, in conjunction with JavaScript code that uses (1) the HTML5 WebSocket API, (2) the Java URLConnection API, or (3) the Silverlight WebClient API, aka a "BEAST" attack.

- CVE ID: CVE-2011-3389
- Published: 2011-09-06
- CVSS Score: 4.3

Timeline

Timeline using the disclosure date **2012-01-27** as reference:

- 2011-09-06 (**-143 days**): CVE-2011-3389 published
- 2012-01-27: Python issue #13885 reported by Antoine Pitrou
- 2012-01-27 (**+0 days**): commit d358e05
- 2012-01-27 (**+0 days**): commit f2bf8a6
- 2012-04-08 (**+72 days**): Python 3.1.5 released
- 2012-04-09 (**+73 days**): Python 2.7.3 released
- 2012-04-10 (**+74 days**): Python 2.6.8 released
- 2012-04-10 (**+74 days**): Python 3.2.3 released
- 2012-09-29: Python 3.3.0 released

CVE-2012-1150: Hash DoS

Hash collision denial of service.

Python 2.6 and 2.7 require the `-R` command line option to enable the fix.

"Effective Denial of Service attacks against web application platforms" talk at the CCC: 2011-12-28

See also the PEP 456: Secure and interchangeable hash algorithm: Python 3.4 switched to SipHash.

- Disclosure date: **2011-12-28** (CCC talk)
- Reported by: Alexander "alech" Klink and Julian "zeri" Wälde

Fixed In

- Python **2.6.8** (2012-04-10) fixed by commit 1e13eb0 (2012-02-21)
- Python **2.7.3** (2012-04-09) fixed by commit 1e13eb0 (2012-02-21)
- Python **3.1.5** (2012-04-08) fixed by commit 2daf6ae (2012-02-20)
- Python **3.2.3** (2012-04-10) fixed by commit 2daf6ae (2012-02-20)
- Python **3.3.0** (2012-09-29) fixed by commit 2daf6ae (2012-02-20)

Python issue

Hash collision security issue.

- Python issue: issue #13703
- Creation date: 2012-01-03
- Reporter: Barry A. Warsaw

CVE-2012-1150

Python before 2.6.8, 2.7.x before 2.7.3, 3.x before 3.1.5, and 3.2.x before 3.2.3 computes hash values without restricting the ability to trigger hash collisions predictably, which allows context-dependent attackers to cause a denial of service (CPU consumption) via crafted input to an application that maintains a hash table.

- CVE ID: CVE-2012-1150
- Published: 2012-10-05
- CVSS Score: 5.0

Timeline

Timeline using the disclosure date **2011-12-28** as reference:

- 2011-12-28: Disclosure date (CCC talk)
- 2012-01-03 (**+6 days**): Python issue #13703 reported by Barry A. Warsaw
- 2012-02-20 (**+54 days**): commit 2daf6ae
- 2012-02-21 (**+55 days**): commit 1e13eb0
- 2012-04-08 (**+102 days**): Python 3.1.5 released

- 2012-04-09 (**+103 days**): Python 2.7.3 released
- 2012-04-10 (**+104 days**): Python 2.6.8 released
- 2012-04-10 (**+104 days**): Python 3.2.3 released
- 2012-09-29: Python 3.3.0 released
- 2012-10-05 (**+282 days**): CVE-2012-1150 published

Links

- https://events.ccc.de/congress/2011/Fahrplan/events/4680.en.html
- http://www.ocert.org/advisories/ocert-2011-003.html

CVE-2011-4944: pypirc created insecurely

Python 2.6 through 3.2 creates `~/.pypirc` configuration file with world-readable permissions before changing them after data has been written, which introduces a race condition that allows local users to obtain a username and password by reading this file.

- Disclosure date: **2011-11-30** (Python issue #13512 reported)

Fixed In

- Python **2.7.4** (2013-04-06) fixed by commit e5567cc (2012-07-03)
- Python **3.2.4** (2013-04-07) fixed by commit e5567cc (2012-07-03)
- Python **3.3.1** (2013-04-07) fixed by commit e5567cc (2012-07-03)
- Python **3.4.0** (2014-03-16) fixed by commit e5567cc (2012-07-03)

Python issue

~/.pypirc created insecurely.

- Python issue: issue #13512
- Creation date: 2011-11-30
- Reporter: Vincent Danen

CVE-2011-4944

Python 2.6 through 3.2 creates ~/.pypirc with world-readable permissions before changing them after data has been written, which introduces a race condition that allows local users to obtain a username and password by reading this file.

- CVE ID: CVE-2011-4944
- Published: 2012-08-27
- CVSS Score: 1.9

Timeline

Timeline using the disclosure date **2011-11-30** as reference:

- 2011-11-30: Python issue #13512 reported by Vincent Danen
- 2012-07-03 (**+216 days**): commit e5567cc
- 2012-08-27 (**+271 days**): CVE-2011-4944 published
- 2013-04-06 (**+493 days**): Python 2.7.4 released
- 2013-04-07 (**+494 days**): Python 3.2.4 released
- 2013-04-07 (**+494 days**): Python 3.3.1 released
- 2014-03-16: Python 3.4.0 released

CVE-2011-1521: urllib redirect

The Python urllib and urllib2 modules are typically used to fetch web pages but by default also contains handlers for `ftp://` and `file://` URL schemes.

Now unfortunately it appears that it is possible for a web server to redirect (HTTP 302) a urllib request to any of the supported schemes.

- Disclosure date: **2011-03-24** (Python issue #11662 reported)
- Reported by: email received on the Python security list

Fixed In

- Python **2.5.6** (2011-05-26) fixed by commit 60a4a90 (2011-03-24)
- Python **2.6.7** (2011-06-03) fixed by commit 60a4a90 (2011-03-24)
- Python **2.7.2** (2011-06-11) fixed by commit 60a4a90 (2011-03-24)
- Python **3.1.4** (2011-06-11) fixed by commit a119df9 (2011-03-29)
- Python **3.2.1** (2011-07-10) fixed by commit a119df9 (2011-03-29)
- Python **3.3.0** (2012-09-29) fixed by commit a119df9 (2011-03-29)

Python issue

Redirect vulnerability in urllib/urllib2.

- Python issue: issue #11662
- Creation date: 2011-03-24
- Reporter: Guido van Rossum

CVE-2011-1521

The urllib and urllib2 modules in Python 2.x before 2.7.2 and 3.x before 3.2.1 process Location headers that specify redirection to file: URLs, which makes it easier for remote attackers to obtain sensitive information or cause a denial of service (resource consumption) via a crafted URL, as demonstrated by the file:///etc/passwd and file:///dev/zero URLs.

- CVE ID: CVE-2011-1521
- Published: 2011-05-24
- CVSS Score: 6.4

Timeline

Timeline using the disclosure date **2011-03-24** as reference:

- 2011-03-24: Python issue #11662 reported by Guido van Rossum
- 2011-03-24 (**+0 days**): commit 60a4a90
- 2011-03-29 (**+5 days**): commit a119df9
- 2011-05-24 (**+61 days**): CVE-2011-1521 published
- 2011-05-26 (**+63 days**): Python 2.5.6 released
- 2011-06-03 (**+71 days**): Python 2.6.7 released
- 2011-06-11 (**+79 days**): Python 2.7.2 released
- 2011-06-11 (**+79 days**): Python 3.1.4 released
- 2011-07-10 (**+108 days**): Python 3.2.1 released
- 2012-09-29: Python 3.3.0 released

CVE-2011-4940: SimpleHTTPServer UTF-7

The `list_directory()` function in `Lib/SimpleHTTPServer.py` in `SimpleHTTPServer` in Python before 2.5.6c1, 2.6.x before 2.6.7 rc2, and 2.7.x before 2.7.2 does not place a charset parameter in the Content-Type HTTP header, which makes it easier for remote attackers to conduct cross-site scripting (XSS) attacks against Internet Explorer 7 via UTF-7 encoding.

- Disclosure date: **2011-03-08** (Python issue #11442 reported)
- Reported by: email received on the Python security list

Fixed In

- Python **2.5.6** (2011-05-26) fixed by commit 3853586 (2011-03-17)
- Python **2.6.7** (2011-06-03) fixed by commit 3853586 (2011-03-17)
- Python **2.7.2** (2011-06-11) fixed by commit 3853586 (2011-03-17)
- Python **3.2.4** (2013-04-07) fixed by commit 3853586 (2011-03-17)
- Python **3.3.1** (2013-04-07) fixed by commit 3853586 (2011-03-17)
- Python **3.4.0** (2014-03-16) fixed by commit 3853586 (2011-03-17)

Python issue

list_directory() in SimpleHTTPServer.py should add charset=... to Content-type header.

- Python issue: issue #11442

- Creation date: 2011-03-08
- Reporter: Guido van Rossum

CVE-2011-4940

The list_directory function in Lib/SimpleHTTPServer.py in SimpleHTTPServer in Python before 2.5.6c1, 2.6.x before 2.6.7 rc2, and 2.7.x before 2.7.2 does not place a charset parameter in the Content-Type HTTP header, which makes it easier for remote attackers to conduct cross-site scripting (XSS) attacks against Internet Explorer 7 via UTF-7 encoding.

- CVE ID: CVE-2011-4940
- Published: 2012-06-27
- CVSS Score: 2.6

Timeline

Timeline using the disclosure date **2011-03-08** as reference:

- 2011-03-08: Python issue #11442 reported by Guido van Rossum
- 2011-03-17 (**+9 days**): commit 3853586
- 2011-05-26 (**+79 days**): Python 2.5.6 released
- 2011-06-03 (**+87 days**): Python 2.6.7 released
- 2011-06-11 (**+95 days**): Python 2.7.2 released
- 2012-06-27 (**+477 days**): CVE-2011-4940 published
- 2013-04-07 (**+761 days**): Python 3.2.4 released
- 2013-04-07 (**+761 days**): Python 3.3.1 released
- 2014-03-16: Python 3.4.0 released

CVE-2010-1634: audioop integer overflows

Multiple integer overflows in `audioop.c` in the `audioop` module in Python 2.6, 2.7, 3.1, and 3.2 allow context-dependent attackers to cause a denial of service (application crash) via a large fragment, as demonstrated by a call to audioop.lin2lin with a long string in the first argument, leading to a buffer overflow.

NOTE: this vulnerability exists because of an incorrect fix for CVE-2008-3143.

- Disclosure date: **2010-05-10** (Python issue #8674 reported)

Fixed In

- Python **2.6.6** (2010-08-24) fixed by commit 7ceb497 (2010-05-11)
- Python **2.7.0** (2010-07-03) fixed by commit 11bb2cd (2010-05-11)
- Python **3.1.3** (2010-11-27) fixed by commit ee289e6 (2010-05-11)
- Python **3.2.0** (2011-02-20) fixed by commit 393b97a (2010-05-11)

Python issue

audioop: incorrect integer overflow checks.

- Python issue: issue #8674
- Creation date: 2010-05-10
- Reporter: Tomas Hoger

CVE-2010-1634

Multiple integer overflows in audioop.c in the audioop module in Python 2.6, 2.7, 3.1, and 3.2 allow context-dependent attackers to cause a denial of service (application crash) via a large fragment, as demonstrated by a call to audioop.lin2lin with a long string in the first argument, leading to a buffer overflow. NOTE: this vulnerability exists because of an incorrect fix for CVE-2008-3143.5.

- CVE ID: CVE-2010-1634
- Published: 2010-05-27
- CVSS Score: 5.0

Timeline

Timeline using the disclosure date **2010-05-10** as reference:

- 2010-05-10: Python issue #8674 reported by Tomas Hoger
- 2010-05-11 (**+1 days**): commit 11bb2cd
- 2010-05-11 (**+1 days**): commit 393b97a
- 2010-05-11 (**+1 days**): commit 7ceb497
- 2010-05-11 (**+1 days**): commit ee289e6
- 2010-05-27 (**+17 days**): CVE-2010-1634 published
- 2010-07-03: Python 2.7.0 released
- 2010-08-24 (**+106 days**): Python 2.6.6 released
- 2010-11-27 (**+201 days**): Python 3.1.3 released
- 2011-02-20: Python 3.2.0 released

CVE-2010-2089: audioop input validation

The `audioop` module in Python 2.7 and 3.2 does not verify the relationships between size arguments and byte string lengths, which allows context-dependent attackers to cause a denial of service (memory corruption and application crash) via crafted arguments, as demonstrated by a call to `audioop.reverse()` with a one-byte string, a different vulnerability than CVE-2010-1634.

- Disclosure date: **2010-01-11** (Python issue #7673 reported)

Fixed In

- Python **2.6.6** (2010-08-24) fixed by commit e9123ef (2010-07-03)
- Python **2.7.2** (2011-06-11) fixed by commit e9123ef (2010-07-03)
- Python **3.1.3** (2010-11-27) fixed by commit 8e42fb7 (2010-07-03)
- Python **3.2.0** (2011-02-20) fixed by commit bc5c54b (2010-07-03)

Python issue

audioop: check that length is a multiple of the size.

- Python issue: issue #7673
- Creation date: 2010-01-11
- Reporter: STINNER Victor

CVE-2010-2089

The audioop module in Python 2.7 and 3.2 does not verify the relationships between size arguments and byte string lengths, which allows context-dependent attackers to cause a denial of service (memory corruption and application crash) via crafted arguments, as demonstrated by a call to audioop.reverse with a one-byte string, a different vulnerability than CVE-2010-1634.

- CVE ID: CVE-2010-2089
- Published: 2010-05-27
- CVSS Score: 5.0

Timeline

Timeline using the disclosure date **2010-01-11** as reference:

- 2010-01-11: Python issue #7673 reported by STINNER Victor
- 2010-05-27 (**+136 days**): CVE-2010-2089 published
- 2010-07-03 (**+173 days**): commit 8e42fb7
- 2010-07-03 (**+173 days**): commit bc5c54b
- 2010-07-03 (**+173 days**): commit e9123ef
- 2010-08-24 (**+225 days**): Python 2.6.6 released
- 2010-11-27 (**+320 days**): Python 3.1.3 released
- 2011-02-20: Python 3.2.0 released
- 2011-06-11 (**+516 days**): Python 2.7.2 released

CVE-2013-1752: httplib unlimited read

Limit the HTTP header readline.

- Disclosure date: **2009-08-28** (Python issue #6791 reported)
- Red Hat impact: Moderate

Fixed In

- Python **2.7.2** (2011-06-11) fixed by commit d7b6ac6 (2010-12-18)
- Python **3.1.4** (2011-06-11) fixed by commit ff1bbba (2010-12-18)
- Python **3.2.0** (2011-02-20) fixed by commit 5466bf1 (2010-12-18)

Python issue

httplib read status memory usage.

- Python issue: issue #6791
- Creation date: 2009-08-28
- Reporter: sumar

Timeline

Timeline using the disclosure date **2009-08-28** as reference:

- 2009-08-28: Python issue #6791 reported by sumar
- 2010-12-18 (**+477 days**): commit 5466bf1
- 2010-12-18 (**+477 days**): commit d7b6ac6
- 2010-12-18 (**+477 days**): commit ff1bbba
- 2011-02-20: Python 3.2.0 released
- 2011-06-11 (**+652 days**): Python 2.7.2 released
- 2011-06-11 (**+652 days**): Python 3.1.4 released

Links

- https://www.cvedetails.com/cve/CVE-2013-1752/

CVE-2010-3492: smtpd accept bug

The `asyncore` module in Python before 3.2 does not properly handle unsuccessful calls to the accept function, and does not have accompanying documentation describing how daemon applications should handle unsuccessful calls to the accept function, which makes it easier for remote attackers to conduct denial of service attacks that terminate these applications via network connections.

- Disclosure date: **2009-08-14** (Python issue #6706 reported)

Fixed In

- Python **2.7.4** (2013-04-06) fixed by commit 977c707 (2010-10-04)
- Python **3.2.0** (2011-02-20) fixed by commit 977c707 (2010-10-04)

Python issue

asyncore's accept() is broken.

- Python issue: issue #6706
- Creation date: 2009-08-14
- Reporter: Giampaolo Rodola'

CVE-2010-3492

The asyncore module in Python before 3.2 does not properly handle unsuccessful calls to the accept function, and does not have accompanying documentation describing how daemon applications should handle unsuccessful calls to the accept function, which makes it easier for remote attackers to conduct denial of service attacks that terminate these applications via network connections.

- CVE ID: CVE-2010-3492
- Published: 2010-10-19
- CVSS Score: 5.0

Timeline

Timeline using the disclosure date **2009-08-14** as reference:

- 2009-08-14: Python issue #6706 reported by Giampaolo Rodola'
- 2010-10-04 (**+416 days**): commit 977c707
- 2010-10-19 (**+431 days**): CVE-2010-3492 published
- 2011-02-20: Python 3.2.0 released
- 2013-04-06 (**+1331 days**): Python 2.7.4 released

CVE-2010-3493: smtpd race conditions

Multiple race conditions in `smtpd.py` in the `smtpd` module in Python 2.6, 2.7, 3.1, and 3.2 alpha allow remote attackers to cause a denial of service (daemon outage) by establishing and then immediately closing a TCP connection, leading to the accept function having an unexpected return value of None, an unexpected value of None for the address, or an ECONNABORTED, EAGAIN, or EWOULDBLOCK error, or the getpeername function having an ENOTCONN error, a related issue to CVE-2010-3492.

- Disclosure date: **2009-08-14** (Python issue #6706 reported)

Fixed In

- Python **2.7.1** (2010-11-27) fixed by commit 19e9fef (2010-11-01)
- Python **3.1.3** (2010-11-27) fixed by commit 5ea3d0f (2010-11-01)
- Python **3.2.1** (2011-07-10) fixed by commit 5ea3d0f (2010-11-01)
- Python **3.3.0** (2012-09-29) fixed by commit 5ea3d0f (2010-11-01)

Python issue

asyncore's accept() is broken.

- Python issue: issue #6706
- Creation date: 2009-08-14
- Reporter: Giampaolo Rodola'

CVE-2010-3493

Multiple race conditions in smtpd.py in the smtpd module in Python 2.6, 2.7, 3.1, and 3.2 alpha allow remote attackers to cause a denial of service (daemon outage) by establishing and then immediately closing a TCP connection, leading to the accept function having an unexpected return value of None, an unexpected value of None for the address, or an ECONNABORTED, EAGAIN, or EWOULDBLOCK error, or the getpeername function having an ENOTCONN error, a related issue to CVE-2010-3492.

- CVE ID: CVE-2010-3493
- Published: 2010-10-19
- CVSS Score: 4.3

Timeline

Timeline using the disclosure date **2009-08-14** as reference:

- 2009-08-14: Python issue #6706 reported by Giampaolo Rodola'
- 2010-10-19 (**+431 days**): CVE-2010-3493 published
- 2010-11-01 (**+444 days**): commit 19e9fef
- 2010-11-01 (**+444 days**): commit 5ea3d0f
- 2010-11-27 (**+470 days**): Python 2.7.1 released
- 2010-11-27 (**+470 days**): Python 3.1.3 released
- 2011-07-10 (**+695 days**): Python 3.2.1 released
- 2012-09-29: Python 3.3.0 released

CVE-2008-2315: Multiple integer overflows (Apple)

Security patches from Apple: prevent integer overflows when allocating memory.

CVE-ID:

- CVE-2008-1679 (`imageop`)
- CVE-2008-1721 (`zlib`)
- CVE-2008-1887 (`PyString_FromStringAndSize()`)
- CVE-2008-2315
- CVE-2008-2316 (`hashlib`)
- CVE-2008-3142 (`unicode_resize()`, `PyMem_RESIZE()`)
- CVE-2008-3144 (`PyOS_vsnprintf()`)
- CVE-2008-4864 (`imageop`)
- Disclosure date: **2008-07-31** (commit)
- Reported by: Apple

Fixed In

- Python **2.6.0** (2008-10-01) fixed by commit e7d8be8 (2008-07-31)
- Python **3.0.0** (2008-12-03) fixed by commit 3ce5d92 (2008-08-24)

CVE-2008-2315

Multiple integer overflows in Python 2.5.2 and earlier allow context-dependent attackers to have an unknown impact via vectors related to the (1) stringobject, (2) unicodeobject, (3) bufferobject, (4) longobject, (5) tupleobject, (6) stropmodule, (7) gcmodule, and (8) mmapmodule modules. NOTE: The expandtabs integer overflows in stringobject and unicodeobject in 2.5.2 are covered by CVE-2008-5031.

- CVE ID: CVE-2008-2315
- Published: 2008-08-01
- CVSS Score: 7.5

Timeline

Timeline using the disclosure date **2008-07-31** as reference:

- 2008-07-31: Disclosure date (commit)
- 2008-07-31 (**+0 days**): commit e7d8be8
- 2008-08-01 (**+1 days**): CVE-2008-2315 published
- 2008-08-24 (**+24 days**): commit 3ce5d92
- 2008-10-01 (**+62 days**): Python 2.6.0 released
- 2008-12-03: Python 3.0.0 released

Links

- https://lists.apple.com/archives/security-announce/2009/Feb/msg00000.html

CVE-2008-3143: Multiple integer overflows (Google)

Multiple integer overflows in Python before 2.5.2 might allow context-dependent attackers to have an unknown impact via vectors related to:

- `Include/pymem.h`
- `Modules/`:
  - `_csv.c`
  - `_struct.c`
  - `arraymodule.c`
  - `audioop.c`
  - `binascii.c`
  - `cPickle.c`
  - `cStringIO.c`
  - `datetimemodule.c`
  - `md5.c`
  - `rgbimgmodule.c`
  - `stropmodule.c`
- `Modules/cjkcodecs/multibytecodec.c`
- `Objects/`:
  - `bufferobject.c`
  - `listobject.c`
  - `obmalloc.c`
- `Parser/node.c`
- `Python/`:
  - `asdl.c`
  - `ast.c`
  - `bltinmodule.c`
  - `compile`

as addressed by "checks for integer overflows, contributed by Google."

- Disclosure date: **2008-04-11** (Python issue #2620 reported)

Fixed In

- Python **2.5.3** (2008-12-19) fixed by commit 83ac014 (2008-07-28)
- Python **2.6.0** (2008-10-01) fixed by commit 0470bab (2008-07-22)
- Python **3.0.0** (2008-12-03) fixed by commit d492ad8 (2008-07-23)

Python issue

Multiple buffer overflows in unicode processing.

- Python issue: issue #2620
- Creation date: 2008-04-11
- Reporter: Justin Ferguson

CVE-2008-3143

Multiple integer overflows in Python before 2.5.2 might allow context-dependent attackers to have an unknown impact via vectors related to (1) Include/pymem.h; (2) _csv.c, (3) _struct.c, (4) arraymodule.c, (5) audioop.c, (6) binascii.c, (7) cPickle.c, (8) cStringIO.c, (9) cjkcodecs/multibytecodec.c, (10) datetimemodule.c, (11) md5.c, (12) rgbimgmodule.c, and (13) stropmodule.c in Modules/; (14) bufferobject.c, (15) listobject.c, and (16) obmalloc.c in Objects/; (17) Parser/node.c; and (18) asdl.c, (19) ast.c, (20) bltinmodule.c, and (21) compile.c in Python/, as addressed by "checks for integer overflows, contributed by Google."

- CVE ID: CVE-2008-3143
- Published: 2008-08-01
- CVSS Score: 7.5

Timeline

Timeline using the disclosure date **2008-04-11** as reference:

- 2008-04-11: Python issue #2620 reported by Justin Ferguson
- 2008-07-22 (**+102 days**): commit 0470bab
- 2008-07-23 (**+103 days**): commit d492ad8
- 2008-07-28 (**+108 days**): commit 83ac014
- 2008-08-01 (**+112 days**): CVE-2008-3143 published
- 2008-10-01: Python 2.6.0 released
- 2008-12-03: Python 3.0.0 released
- 2008-12-19 (**+252 days**): Python 2.5.3 released

CVE-2008-5031: expandtab() integer overflow

Multiple integer overflows in Python 2.2.3 through 2.5.1, and 2.6, allow context-dependent attackers to have an unknown impact via a large integer value in the tabsize argument to the expandtabs method, as implemented by:

- the `string_expandtabs()` function in `Objects/stringobject.c`
- the `unicode_expandtabs()` function in `Objects/unicodeobject.c`

NOTE: this vulnerability reportedly exists because of an incomplete fix for CVE-2008-2315.

- Disclosure date: **2008-03-11** (commit date)
- Reported by: Chris Evans

Fixed In

- Python **2.5.3** (2008-12-19) fixed by commit 44a93e5 (2008-03-11)
- Python **2.6.0** (2008-10-01) fixed by commit 5bdff60 (2008-03-11)
- Python **3.0.0** (2008-12-03) fixed by commit dd15f6c (2008-03-16)

CVE-2008-5031

Multiple integer overflows in Python 2.2.3 through 2.5.1, and 2.6, allow context-dependent attackers to have an unknown impact via a large integer value in the tabsize argument to the expandtabs method, as implemented by (1) the string_expandtabs function in Objects/stringobject.c and (2) the unicode_expandtabs function in Objects/unicodeobject.c. NOTE: this vulnerability reportedly exists because of an incomplete fix for CVE-2008-2315.

- CVE ID: CVE-2008-5031
- Published: 2008-11-10
- CVSS Score: 10.0

Timeline

Timeline using the disclosure date **2008-03-11** as reference:

- 2008-03-11: Disclosure date (commit date)
- 2008-03-11 (**+0 days**): commit 44a93e5
- 2008-03-11 (**+0 days**): commit 5bdff60
- 2008-03-16 (**+5 days**): commit dd15f6c
- 2008-10-01: Python 2.6.0 released
- 2008-11-10 (**+244 days**): CVE-2008-5031 published
- 2008-12-03: Python 3.0.0 released
- 2008-12-19 (**+283 days**): Python 2.5.3 released

Links

- http://scary.beasts.org/security/CESA-2008-008.html

CVE-2011-1015: CGI directory traversal

The `is_cgi()` method in `CGIHTTPServer.py` in the `CGIHTTPServer` module in Python 2.5, 2.6, and 3.0 allows remote attackers to read script source code via an HTTP GET request that lacks a / (slash) character at the beginning of the URI.

- Disclosure date: **2008-03-07** (Python issue #2254 reported)

Fixed In

- Python **2.7.0** (2010-07-03) fixed by commit 923ba36 (2009-04-06)
- Python **3.2.4** (2013-04-07) fixed by commit 923ba36 (2009-04-06)
- Python **3.3.1** (2013-04-07) fixed by commit 923ba36 (2009-04-06)
- Python **3.4.0** (2014-03-16) fixed by commit 923ba36 (2009-04-06)

Python issue

Python CGIHTTPServer information disclosure.

- Python issue: issue #2254
- Creation date: 2008-03-07
- Reporter: sumar

CVE-2011-1015

The is_cgi method in CGIHTTPServer.py in the CGIHTTPServer module in Python 2.5, 2.6, and 3.0 allows remote attackers to read script source code via an HTTP GET request that lacks a / (slash) character at the beginning of the URI.

- CVE ID: CVE-2011-1015
- Published: 2011-05-09
- CVSS Score: 5.0

Timeline

Timeline using the disclosure date **2008-03-07** as reference:

- 2008-03-07: Python issue #2254 reported by sumar
- 2009-04-06 (**+395 days**): commit 923ba36
- 2010-07-03 (**+848 days**): Python 2.7.0 released
- 2011-05-09 (**+1158 days**): CVE-2011-1015 published
- 2013-04-07 (**+1857 days**): Python 3.2.4 released
- 2013-04-07 (**+1857 days**): Python 3.3.1 released
- 2014-03-16: Python 3.4.0 released

CVE-2007-4965: rgbimg and imageop overflows

Multiple integer overflows in the `imageop` module in Python 2.5.1 and earlier allow context-dependent attackers to cause a denial of service (application crash) and possibly obtain sensitive information (memory contents) via crafted arguments to (1) the `tovideo()` method, and unspecified other vectors related to (2) `imageop.c`, (3) `rbgimgmodule.c`, and other files, which trigger heap-based buffer overflows.

CVE-ID:

- CVE-2007-4965

- CVE-2009-4134
- CVE-2010-1449
- CVE-2010-1450

Reported again by Marc Schoenefeld in the Red Hat bugzilla at 2009-11-26.

- Disclosure date: **2007-09-16** (full-disclosure email)
- Reported by: Slythers Bro (on the full-disclosure mailing list)

Fixed In

- Python **2.5.3** (2008-12-19) fixed by commit 4df1b6d (2008-08-19)
- Python **2.6.0** (2008-10-01) fixed by commit 93ebfb1 (2008-08-19)

Python issue

[CVE-2007-4965] Integer overflow in imageop module.

- Python issue: issue #1179
- Creation date: 2007-09-19
- Reporter: Ismail Donmez

CVE-2007-4965

Multiple integer overflows in the imageop module in Python 2.5.1 and earlier allow context-dependent attackers to cause a denial of service (application crash) and possibly obtain sensitive information (memory contents) via crafted arguments to (1) the tovideo method, and unspecified other vectors related to (2) imageop.c, (3) rbgimgmodule.c, and other files, which trigger heap-based buffer overflows.

- CVE ID: CVE-2007-4965
- Published: 2007-09-18
- CVSS Score: 5.8

Timeline

Timeline using the disclosure date **2007-09-16** as reference:

- 2007-09-16: Disclosure date (full-disclosure email)
- 2007-09-18 (**+2 days**): CVE-2007-4965 published
- 2007-09-19 (**+3 days**): Python issue #1179 reported by Ismail Donmez
- 2008-08-19 (**+338 days**): commit 4df1b6d
- 2008-08-19 (**+338 days**): commit 93ebfb1
- 2008-10-01: Python 2.6.0 released
- 2008-12-19 (**+460 days**): Python 2.5.3 released

Links

- http://seclists.org/fulldisclosure/2007/Sep/279
- http://bugs.python.org/issue8678
- https://bugzilla.redhat.com/show_bug.cgi?id=541698

Python SSL and TLS security

Evolutions of the `ssl` module.

Cipher suite

Python 2.7 and 3.5-3.7:

```
_DEFAULT_CIPHERS = (
    'ECDH+AESGCM:ECDH+CHACHA20:DH+AESGCM:DH+CHACHA20:ECDH+AES256:DH+AES256:'
    'ECDH+AES128:DH+AES:ECDH+HIGH:DH+HIGH:RSA+AESGCM:RSA+AES:RSA+HIGH:'
    '!aNULL:!eNULL:!MD5:!3DES'
    )
```

Pytohn 3.4:

```
_DEFAULT_CIPHERS = (
    'ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:ECDH+HIGH:'
    'DH+HIGH:ECDH+3DES:DH+3DES:RSA+AESGCM:RSA+AES:RSA+HIGH:RSA+3DES:!aNULL:'
    '!eNULL:!MD5'
)
```

Python 3.3:

```
_DEFAULT_CIPHERS = 'DEFAULT:!aNULL:!eNULL:!LOW:!EXPORT:!SSLv2'
```

Options

- `SSL_OP_DONT_INSERT_EMPTY_FRAGMENTS`: CBC IV attack countermeasure (CVE-2011-3389)
- `SSL_OP_NO_SSLv2`: SSLv2 is unsafe
- `SSL_OP_NO_SSLv3`: SSLv3 is unsafe
- `SSL_OP_NO_COMPRESSION`: CRIME countermeasure
- `SSL_OP_CIPHER_SERVER_PREFERENCE`
- `SSL_OP_SINGLE_DH_USE`
- `SSL_OP_SINGLE_ECDH_USE`

Python 3.7:

```
/* Defaults */
    options = SSL_OP_ALL & ~SSL_OP_DONT_INSERT_EMPTY_FRAGMENTS;
    if (proto_version != PY_SSL_VERSION_SSL2)
        options |= SSL_OP_NO_SSLv2;
```

```
    if (proto_version != PY_SSL_VERSION_SSL3)
        options |= SSL_OP_NO_SSLv3;
    /* Minimal security flags for server and client side context.
     * Client sockets ignore server-side parameters. */
#ifdef SSL_OP_NO_COMPRESSION
    options |= SSL_OP_NO_COMPRESSION;
#endif
#ifdef SSL_OP_CIPHER_SERVER_PREFERENCE
    options |= SSL_OP_CIPHER_SERVER_PREFERENCE;
#endif
#ifdef SSL_OP_SINGLE_DH_USE
    options |= SSL_OP_SINGLE_DH_USE;
#endif
#ifdef SSL_OP_SINGLE_ECDH_USE
    options |= SSL_OP_SINGLE_ECDH_USE;
#endif
    SSL_CTX_set_options(self->ctx, options);
```

CA store

`SSLContext.load_default_certs()` new in Python 3.4.

- Windows: `ssl.enum_certificates(store_name)`, new in Python 3.4. Use CertOpenStore() and `CertEnumCertificatesInStore()` functions.
- Linux: xxx
- macOS: xxx

See also

- certifi: "Python package for providing Mozilla's CA Bundle".
- [Python-Dev] SSL certificates recommendations for downstream python packagers

SSLContext

New in Python 3.2.

CRLs

- `SSLContext.verify_flags`: New in Python 3.4
- `SSLContext.load_verify_locations()`: This method can also load certification revocation lists (CRLs) in PEM or DER format. New in Python 3.5.
- `ssl.enum_crls(store_name)`: new in Python 3.4, specific to Windows

Validate TLS certificates

- Python decides for certificate validation (September, 2014)
- CVE-2014-9365
- Python 2.7.9 (2014-12-10)

- Python 3.4.3 (2015-02-23)
- PEP 476: Enabling certificate verification by default for stdlib http clients: Python 3.4.3, 3.5
- PEP 466: Python 2.7.9
- Version matrix?
  - HTTP
  - SMTP
  - FTP
  - IMAP
  - POP3
  - XML-RPC
  - NNTP

TLS versions

- SSLv2 now black listed
- SSLv3 now black listed

OpenSSL versions

Python bundled OpenSSL in Windows and macOS installers.

OpenSSL versions (read from the Windows installer):

- Python 3.6.1: OpenSSL 1.0.2k
- Python 2.7.13, 3.5.3 and 3.6.0: OpenSSL 1.0.2j
- Python 2.7.12, 3.5.2: OpenSSL 1.0.2h
- Python 2.7.11, 3.4.4, 3.5.0, 3.5.1: OpenSSL 1.0.2d
- Python 2.7.10: OpenSSL 1.0.2a
- Python 2.7.9: OpenSSL 1.0.1j
- Python 3.3.5: OpenSSL 1.0.1e

Windows: see PCbuild/get_externals.bat (or PCbuild/readme.txt in older versions).

macOS: see Mac/BuildScript/build-installer.py.

macOS:

```
# Since Apple removed the header files for the deprecated system
# OpenSSL as of the Xcode 7 release (for OS X 10.10+), we do not
# have much choice but to build our own copy here, too.
```

Example of OpenSSL update: Upgrade installers to OpenSSL 1.0.2k (March 2017).

Links

- The future of the Python ssl module (June, 2016)
- cryptography (cryptography.io): Python library which exposes cryptographic recipes and primitives
- pyOpenSSL
- M2Crypto
- *urllib3 <https://urllib3.readthedocs.io/>_*
- LibreSSL
- borringssl

Python releases

- 2.7.12: 2016-06-28
- 2.7.10: 2015-05-23
- 3.4.5, 3.5.2: 2016-06-27
- 3.4.4: 2015-12-21
- 2.6.8: 2012-04-10
- 3.2.3: 2012-04-10
- 2.7.3: 2012-04-09
- 3.1.5: 2012-04-08

See Misc/NEWS for release dates.

TODO list

TODO list for this python-security documentation.

- Get Red Hat impact from a Red Hat URL?

cookielib

Add https://hackerone.com/reports/26647 vulnerability.

https://bugs.python.org/issue16611 #16611: BaseCookie now parses 'secure' and 'httponly' flags.

https://bugs.python.org/issue22796 Regression in Python 3.2 cookie parsing

https://bugs.python.org/issue25228 Support for httponly/secure cookies reintroduced lax parsing behavior

https://code.djangoproject.com/ticket/26158 cookie parsing fails with python 3.x if request contains unnamed cookie

YAML template:

```
- name: "Issue #22796"
  summary: >
    hardened HTTP cookie parsing
  links:
    - http://bugs.python.org/issue22796
  disclosure: "2014-11-04 (issue #22796 created)"
  fixed-in:
   - b1e36073cdde71468efa27e88016aa6dd46f3ec7 # 3.x
  description: >
    HTTP cookie parsing is now stricter, in order to protect against potential
    injection attacks.

    Reported by Tim Graham.
```

CHAPTER 2

Python branches

- (Latest update: 2017-03-28) Python 2.6, 3.0, 3.1, 3.2 don't get security fixes anymore and so should be considered as vulnerable
- Branches getting security fixes: 2.7, 3.3, 3.4 and 3.5
- See Status of Python branches

CHAPTER 3

Dangerous functions and modules

- Python 2 input()
- Python 2 execfile()
- eval()
- subprocess.Popen(shell=True)
- str.format(), Python 3 str.format_map, and Python 2 unicode.format() all allow arbitrary attribute access on formatted values, and hence access to Python's introspection features: Be Careful with Python's New-Style String Format (Armin Ronacher, December 2016)
- The `pickle` module executes arbitrary Python code: never use it with untrusted data.
- archives:
  - tarfile: Never extract archives from untrusted sources without prior inspection. It is possible that files are created outside of path, e.g. members that have absolute filenames starting with "/" or filenames with two dots "..".
  - zipfile: Never extract archives from untrusted sources without prior inspection. It is possible that files are created outside of path, e.g. members that have absolute filenames starting with "/" or filenames with two dots "..". zipfile attempts to prevent that.

CHAPTER 4

Security model

Bytecode

CPython doesn't verify that bytecode is safe. If an attacker is able to execute arbitrary bytecode, we consider that the security of the bytecode is the least important issue: using bytecode, sensitive code can be imported and executed.

For example, the `marshal` doesn't validate inputs.

Sandbox

Don't try to build a sandbox inside CPython. The attack surface is too large. Python has many introspection features, see for example the `inspect` module. Python also many convenient features which executes code on demand. Examples:

- the literal string `'\N{Snowman}'` imports the `unicodedata` module
- the code to log a warning might be abused to execute code

The good design is to put CPython into a sandbox, not the opposite.

Ok, understood, but I want a sandbox in Python. Well...

- Eval really is dangerous (Ned Batchelder, June 2012)
- PyPy sandboxing
- For Linux, search for SECCOMP

CHAPTER 5

RNG

- CSPRNG:
  - `os.urandom()`
  - `random.SystemRandom`
  - secrets module (Python 3.6)
- `os.urandom()` uses:
  - Python 3.6: `CryptGenRandom()`, `getentropy()`, `getrandom(0)` (blocking) or `/dev/urandom`
  - Python 3.5: `CryptGenRandom()`, `getentropy()`, `getrandom(GRND_NONBLOCK)` (non-blocking) or `/dev/urandom`
  - Python 2.7: `CryptGenRandom()`, `getentropy()` or `/dev/urandom`
  - PEP 524: Make os.urandom() blocking on Linux: Python 3.6
- `ssl.RAND_bytes()` fork issue:
  - Python issue: Re-seed OpenSSL's PRNG after fork
  - OpenSSL Random fork-safety

The `random` module must not be used in security sensitive code, except of the `random.SystemRandom` class.

CHAPTER 6

CPython Security Experts

- Alex Gaynor
- Antoine Pitrou
- Christian Heimes
- Donald Stufft

CHAPTER 7

Misc

- python3 -E: ignore `PYTHON*` environment variables like `PYTHONPATH`
- python3 -I: isolated mode, also implies `-E` and `-s`
- Python 3.7 adds a `is_safe` attribute to uuid.UUID objects: http://bugs.python.org/issue22807
- XML: defusedxml, XML bomb protection for Python stdlib modules
- Coverity:
  - Coverity Scan: Python
  - devguide info about Coverity
  - analysis of 2012 by Coverity Software resulted in CPython receiving their highest quality rating.
- Windows: ASLR and DEP protections enabled since Python 3.4 (and Python 2.7.11 if built using `PCbuild/` directory)
- sys.path:
  - CVE-2008-5983: http://bugs.python.org/issue5753 added `PySys_SetArgvEx()`
  - CVE-2015-5652: Untrusted search path vulnerability in python.exe in Python through 3.5.0 on Windows allows local users to gain privileges via a Trojan horse readline.pyd file in the current working directory. NOTE: the vendor says "It was determined that this is a longtime behavior of Python that cannot really be altered at this point."
  - `python -E`, `python -I`
- Python at HackerOne
- humans.txt of python.org with the list of "people who found security bugs in the website". For the rationale, see humanstxt.org.

CHAPTER 8

Python Security Response Team (PSRT)

- Handle security@python.org incoming emails
- PSRT issues (private)
- LWN: The Python security response team (June, 2016)

CHAPTER 9

Links

- Reporting security issues in Python
- OWASP Python Security Project (pythonsecurity.org)
- bandit: Python AST-based static analyzer from OpenStack Security Group
- Python CVEs (cvedetails.com)
- https://github.com/pyupio/safety-db and https://pyup.io/
- owasp-pysec: OWASP Python Security Project