
Python Security Documentation

Release 0.0

Victor Stinner

Jan 16, 2019

Contents

1	Pages	3
1.1	Security vulnerabilities	3
1.2	Packages and PyPI	62
1.3	Python SSL and TLS security	66
1.4	Python Security	70
1.5	TODO list	74

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

1.1 Security vulnerabilities

Total: 53 vulnerabilities.

Vulnerability	Disclosure	Fixed In	V
<i>CVE-2018-1000117: Buffer overflow vulnerability in os.symlink on Windows</i>	2018-03-05	3.4.9 3.5.6 3.6.5 3.7.0	2
<i>CVE-2018-1060: difflib and poplib catastrophic backtracking</i>	2018-03-02	2.7.15 3.4.9 3.5.6 3.6.5 3.7.0	-
<i>CVE-2018-1000030: Python 2.7 readahead is not thread safe</i>	2017-09-20	2.7.15	-
<i>Expat 2.2.3</i>	2017-07-17	2.7.14 3.3.7 3.4.8 3.5.5 3.6.3 3.7.0	-
<i>Environment variables injection in subprocess on Windows</i>	2017-06-22	2.7.14 3.3.7 3.4.7 3.5.4 3.6.2	-
<i>CVE-2017-9233: Expat 2.2.1</i>	2017-06-17	2.7.14 3.3.7 3.4.7 3.5.4 3.6.2	-
<i>CVE-2017-1000158: PyString_DecodeEscape integer overflow</i>	2017-06-13	2.7.14 3.4.8 3.5.5	-
<i>bpo-30500: urllib connects to a wrong host</i>	2017-05-29	2.7.14 3.3.7 3.4.7 3.5.4 3.6.2	-
<i>urllib FTP protocol stream injection</i>	2017-02-20	2.7.14 3.3.7 3.4.7 3.5.4 3.6.3 3.7.0	-
<i>CVE-2016-0718: expat 2.2, bug #537</i>	2017-02-17	2.7.14 3.3.7 3.4.7 3.5.4 3.6.2	-
<i>update zlib to 1.2.11</i>	2017-01-05	2.7.14 3.4.8 3.5.4 3.6.1 3.7.0	-
<i>Issue #28563: gettext.c2py()</i>	2016-10-30	2.7.13 3.3.7 3.4.6 3.5.3 3.6.0	-
<i>CVE-2016-2183: Sweet32 attack (DES, 3DES)</i>	2016-08-24	2.7.13 3.4.7 3.5.3 3.6.0	-
<i>CVE-2016-1000110: HTTPoxy attack</i>	2016-07-18	2.7.13 3.3.7 3.4.6 3.5.3 3.6.0	-
<i>CVE-2016-0772: smtplib TLS stripping</i>	2016-06-11	2.7.12 3.3.7 3.4.5 3.5.2 3.6.0	-
<i>Issue #26657: HTTP directory traversal</i>	2016-03-28	2.7.12 3.3.7 3.4.7 3.5.2 3.6.0	-
<i>Issue #26556: Expat 2.1.1</i>	2016-03-14	2.7.12 3.3.7 3.4.5 3.5.2 3.6.0	-
<i>CVE-2016-5636: zipimporter overflow</i>	2016-01-21	2.7.12 3.3.7 3.4.5 3.5.2 3.6.0	-
<i>CVE-2016-5699: HTTP header injection</i>	2014-11-24	2.7.10 3.3.7 3.4.4 3.5.0	-
<i>CVE-2014-9365: Validate TLS certificate</i>	2014-08-28	2.7.9 3.4.3 3.5.0	-
<i>CVE-2014-7185: buffer() integer overflows</i>	2014-06-24	2.7.8	-
<i>CVE-2014-4616: JSONDecoder.raw_decode</i>	2014-04-13	2.7.7 3.2.6 3.3.6 3.4.1 3.5.0	-
<i>CVE-2014-2667: os.makedirs() not thread-safe</i>	2014-03-28	3.2.6 3.3.6 3.4.1 3.5.0	-
<i>CVE-2014-1912: socket.recvfrom_into() overflow</i>	2014-01-14	2.7.7 3.2.6 3.3.4 3.4.0	-

Continued on r

Table 1 – continued from previous page

Vulnerability	Disclosure	Fixed In	V
<i>CVE-2013-7338: zipfile DoS using malformed file</i>	2013-12-27	3.3.4 3.4.0	–
<i>Issue #19435: CGI directory traversal</i>	2013-10-29	2.7.6 3.2.6 3.3.4 3.4.0	–
<i>CVE-2013-4238: ssl: NUL in subjectAltNames</i>	2013-06-27	2.6.9 2.7.6 3.2.6 3.3.3 3.4.0	–
<i>CVE-2013-7440: ssl.match_hostname() IDNA issue</i>	2013-05-17	3.3.3 3.4.0	–
<i>CVE-2013-2099: ssl.match_hostname() wildcard DoS</i>	2013-05-15	3.2.6 3.3.3 3.4.0	–
<i>CVE-2013-1752: CVE-2013-1752: Limit imaplib.IMAP4_SSL.readline()</i>	2012-09-25		2
<i>CVE-2013-1752: ftplib unlimited read</i>	2012-09-25	2.7.6 3.2.6 3.3.3 3.4.0	–
<i>CVE-2013-1752: nntplib unlimited read</i>	2012-09-25	2.6.9 2.7.6 3.2.6 3.3.7 3.4.3 3.5.0	–
<i>CVE-2013-1752: poplib unlimited read</i>	2012-09-25	2.7.9 3.2.6 3.3.7 3.4.3 3.5.0	–
<i>CVE-2013-1752: smtplib unlimited read</i>	2012-09-25	2.7.9 3.2.6 3.3.7 3.4.3 3.5.0	–
<i>CVE-2013-1753: xmlrpc gzip unlimited read</i>	2012-09-25	2.7.9 3.3.7 3.4.3 3.5.0	–
<i>CVE-2013-7040: Hash not properly randomized</i>	2012-04-19	3.4.0	2
<i>CVE-2012-2135: UTF-16 decoder</i>	2012-04-14	2.7.4 3.2.4 3.3.0	–
<i>CVE-2012-0845: XML-RPC DoS</i>	2012-02-13	2.6.8 2.7.3 3.1.5 3.2.3 3.3.0	–
<i>CVE-2011-3389: ssl CBC IV attack</i>	2012-01-27	2.6.8 2.7.3 3.1.5 3.2.3 3.3.0	–
<i>CVE-2012-1150: Hash DoS</i>	2011-12-28	2.6.8 2.7.3 3.1.5 3.2.3 3.3.0	–
<i>CVE-2011-4944: pypirc created insecurely</i>	2011-11-30	2.7.4 3.2.4 3.3.1 3.4.0	–
<i>CVE-2011-1521: urllib redirect</i>	2011-03-24	2.5.6 2.6.7 2.7.2 3.1.4 3.2.1 3.3.0	–
<i>CVE-2011-4940: SimpleHTTPServer UTF-7</i>	2011-03-08	2.5.6 2.6.7 2.7.2 3.2.4 3.3.1 3.4.0	–
<i>CVE-2010-1634: audioop integer overflows</i>	2010-05-10	2.6.6 2.7.0 3.1.3 3.2.0	–
<i>CVE-2010-2089: audioop input validation</i>	2010-01-11	2.6.6 2.7.2 3.1.3 3.2.0	–
<i>CVE-2013-1752: httplib unlimited read</i>	2009-08-28	2.7.2 3.1.4 3.2.0	–
<i>CVE-2010-3492: smtpd accept bug</i>	2009-08-14	2.7.4 3.2.0	–
<i>CVE-2010-3493: smtpd race conditions</i>	2009-08-14	2.7.1 3.1.3 3.2.1 3.3.0	–
<i>CVE-2008-2315: Multiple integer overflows (Apple)</i>	2008-07-31	2.6.0 3.0.0	–
<i>CVE-2008-3143: Multiple integer overflows (Google)</i>	2008-04-11	2.5.3 2.6.0 3.0.0	–
<i>CVE-2008-5031: expandtab() integer overflow</i>	2008-03-11	2.5.3 2.6.0 3.0.0	–
<i>CVE-2011-1015: CGI directory traversal</i>	2008-03-07	2.7.0 3.2.4 3.3.1 3.4.0	–
<i>CVE-2007-4965: rgbimg and imageop overflows</i>	2007-09-16	2.5.3 2.6.0	–

Table of Contents:

1.1.1 CVE-2018-1000117: Buffer overflow vulnerability in os.symlink on Windows

On February 27th, 2018, the Python Security Response team was notified of a buffer overflow issue in the `os.symlink()` method on Windows. The issue affects all versions of Python between 3.2 and 3.6.4, including the 3.7 beta releases. It has been patched for the next releases of 3.4, 3.5, 3.6 and 3.7.

Scripts may be vulnerable if they use `os.symlink()` on Windows and an attacker is able to influence the location where links are created. As `os.symlink` requires additional privileges, exploits using this vulnerability are more likely to result in escalation of privilege.

Besides applying the fix to CPython, scripts can also ensure that the length of each path argument is less than 260, and if the source is a relative path, that its combination with the destination is also shorter than 260 characters. That is:

```
assert (len(src) < 260 and
        len(dest) < 260 and
        len(os.path.join(os.path.dirname(dest), src)) < 260)
os.symlink(src, dest)
```

Scripts that explicitly pass the `target_is_directory` argument as `True` are not vulnerable. Scripts on Python 3.5 that use

bytes for paths are not vulnerable, because of a combination of stack layout and added parameter validation, but will still not behave correctly for long paths.

This vulnerability has been registered as CVE-2018-1000117, and patched in the commits listed below. This patch prevents the buffer overflow, but does not raise any new errors or enable the use of long paths when creating symlinks.

Many thanks to **Alexey Izbyshev** for the report, and helping us work through developing the patch.

- Disclosure date: **2018-03-05** (Python issue [bpo-33001](#) reported)
- Reported at: 2018-02-27 (email to the PSRT)
- Reported by: Alexey Izbyshev

Fixed In

- Python **3.4.9** (2018-08-02) fixed by [commit 77c02cd](#) (2018-05-14)
- Python **3.5.6** (2018-08-02) fixed by [commit f381cfe](#) (2018-05-14)
- Python **3.6.5** (2018-03-28) fixed by [commit baa4507](#) (2018-03-05)
- Python **3.7.0** (2018-06-28) fixed by [commit 96fdbac](#) (2018-03-05)

Vulnerable Versions

- Python **2.7**

Python issue

Buffer overflow vulnerability in `os.symlink` on Windows (CVE-2018-1000117).

- Python issue: [bpo-33001](#)
- Creation date: 2018-03-05
- Reporter: Steve Dower

CVE-2018-1000117

Python Software Foundation CPython version From 3.2 until 3.6.4 on Windows contains a Buffer Overflow vulnerability in `os.symlink()` function on Windows that can result in Arbitrary code execution, likely escalation of privilege. This attack appears to be exploitable via a python script that creates a symlink with an attacker controlled name or location. This vulnerability appears to have been fixed in 3.7.0 and 3.6.5.

- CVE ID: [CVE-2018-1000117](#)
- Published: 2018-03-07

Timeline

Timeline using the disclosure date **2018-03-05** as reference:

- 2018-02-27 (**-6 days**): Reported (email to the PSRT)
- 2018-03-05: [Python issue bpo-33001](#) reported by Steve Dower
- 2018-03-05 (**+0 days**): [commit 96fdbac](#)

- 2018-03-05 (+0 days): `commit baa4507`
- 2018-03-07 (+2 days): CVE-2018-1000117 published
- 2018-03-28 (+23 days): Python 3.6.5 released
- 2018-05-14 (+70 days): `commit 77c02cd`
- 2018-05-14 (+70 days): `commit f381cfe`
- 2018-06-28: Python 3.7.0 released
- 2018-08-02 (+150 days): Python 3.4.9 released
- 2018-08-02 (+150 days): Python 3.5.6 released

Links

- <https://mail.python.org/mm3/archives/list/security-announce@python.org/thread/PVSURQ2YCNZODILA3QE7ZF3GCD25EVVT/>

1.1.2 CVE-2018-1060: `difflib` and `poplib` catastrophic backtracking

Regexes in `difflib` and `poplib` were vulnerable to catastrophic backtracking. These regexes formed potential DOS vectors (REDOS). They have been refactored.

This resolves CVE-2018-1060 and CVE-2018-1061.

Patch by **Jamie Davis**.

- Disclosure date: **2018-03-02** (Python issue `bpo-32981` reported)

Fixed In

- Python **2.7.15** (2018-04-29) fixed by `commit e052d40` (2018-03-04)
- Python **3.4.9** (2018-08-02) fixed by `commit 942cc04` (2018-03-11)
- Python **3.5.6** (2018-08-02) fixed by `commit 937ac1f` (2018-03-11)
- Python **3.6.5** (2018-03-28) fixed by `commit c951675` (2018-03-04)
- Python **3.7.0** (2018-06-28) fixed by `commit 0902a2d` (2018-03-04)

Python issue

Catastrophic backtracking in `poplib` (CVE-2018-1060) and `difflib` (CVE-2018-1061).

- Python issue: `bpo-32981`
- Creation date: 2018-03-02
- Reporter: James Davis

Timeline

Timeline using the disclosure date **2018-03-02** as reference:

- 2018-03-02: Python issue [bpo-32981](#) reported by James Davis
- 2018-03-04 (+2 days): [commit 0902a2d](#)
- 2018-03-04 (+2 days): [commit c951675](#)
- 2018-03-04 (+2 days): [commit e052d40](#)
- 2018-03-11 (+9 days): [commit 937ac1f](#)
- 2018-03-11 (+9 days): [commit 942cc04](#)
- 2018-03-28 (+26 days): Python 3.6.5 released
- 2018-04-29 (+58 days): Python 2.7.15 released
- 2018-06-28: Python 3.7.0 released
- 2018-08-02 (+153 days): Python 3.4.9 released
- 2018-08-02 (+153 days): Python 3.5.6 released

Links

- <https://www.cvedetails.com/cve/CVE-2018-1060/>

1.1.3 CVE-2018-100030: Python 2.7 readahead is not thread safe

Reading from the same file object in different threads does crash Python 2.7. The readahead feature of `Objects/fileobject.c` is not thread safe.

The PSRT decided that it's a regular bug and doesn't need to be categorized as a vulnerability, since the attacker has to be able to run arbitrary code in practice.

The PSRT considers that no Python 2.7 application currently rely on reading from the same file object "at the same time" from different thread, since it currently crashes.

- Disclosure date: **2017-09-20** (Python issue [bpo-31530](#) reported)
- Reported by: email to PSRT

Fixed In

- Python **2.7.15** (2018-04-29) fixed by [commit dbf52e0](#) (2018-01-02)

Python issue

CVE-2018-100030: Python 2.7 readahead feature of file objects is not thread safe.

- Python issue: [bpo-31530](#)
- Creation date: 2017-09-20
- Reporter: STINNER Victor

CVE-2018-100030

Python 2.7.14 is vulnerable to a Heap-Buffer-Overflow as well as a Heap-Use-After-Free. Python versions prior to 2.7.14 may also be vulnerable and it appears that Python 2.7.17 and prior may also be vulnerable however this has not been confirmed. The vulnerability lies when multiply threads are handling large amounts of data. In both cases there is essentially a race condition that occurs. For the Heap-Buffer-Overflow, Thread 2 is creating the size for a buffer, but Thread1 is already writing to the buffer without knowing how much to write. So when a large amount of data is being processed, it is very easy to cause memory corruption using a Heap-Buffer-Overflow. As for the Use-After-Free, Thread3->Malloc->Thread1->Free's->Thread2-Re-uses-Free'd Memory. The PSRT has stated that this is not a security vulnerability due to the fact that the attacker must be able to run code, however in some situations, such as function as a service, this vulnerability can potentially be used by an attacker to violate a trust boundary, as such the DWF feels this issue deserves a CVE.

- CVE ID: [CVE-2018-100030](#)
- Published: 2018-02-08
- CVSS Score: 6.8

Timeline

Timeline using the disclosure date **2017-09-20** as reference:

- 2017-09-20: Python issue [bpo-31530](#) reported by STINNER Victor
- 2018-01-02 (+104 days): [commit dbf52e0](#)
- 2018-02-08 (+141 days): CVE-2018-100030 published
- 2018-04-29 (+221 days): Python 2.7.15 released

Links

- <https://access.redhat.com/security/cve/cve-2018-100030>

1.1.4 Expat 2.2.3

Expat 2.2.2 was released with multiple security fixes:

- #43: Protect against compilation without any source of high quality entropy enabled, e.g. with CMake build system
- #60: Windows with `_UNICODE`: Unintended use of `LoadLibraryW` with a non-wide string resulted in failure to load `advapi32.dll` and degradation in quality of used entropy when compiled with `_UNICODE` for Windows; you can launch existing binaries with `EXPAT_ENTROPY_DEBUG=1` in the environment to inspect the quality of entropy used during runtime
- [MOX-006]: Fix non-NULL parser parameter validation in `XML_Parse`; resulted in NULL dereference, previously

Expat 2.2.3 contains an additional security fix: #82: CVE-2017-11742 – Windows: Fix DLL hijacking vulnerability using Steve Holme's `LoadLibrary` wrapper for/of `cURL`

- Disclosure date: **2017-07-17** (Python issue [bpo-30947](#) reported)

Fixed In

- Python **2.7.14** (2017-09-17) fixed by [commit ec4ab09](#) (2017-08-18)
- Python **3.3.7** (2017-09-19) fixed by [commit 297516e](#) (2017-09-06)
- Python **3.4.8** (2018-02-04) fixed by [commit 86a713c](#) (2017-09-24)
- Python **3.5.5** (2018-02-04) fixed by [commit f2492bb](#) (2017-09-25)
- Python **3.6.3** (2017-10-03) fixed by [commit 83e37e1](#) (2017-08-18)
- Python **3.7.0** (2018-06-28) fixed by [commit 93d0cb5](#) (2017-08-18)

Python issue

Update embeded copy of libexpat from 2.2.1 to 2.2.3.

- Python issue: [bpo-30947](#)
- Creation date: 2017-07-17
- Reporter: STINNER Victor

Timeline

Timeline using the disclosure date **2017-07-17** as reference:

- 2017-07-17: Python issue [bpo-30947](#) reported by STINNER Victor
- 2017-08-18 (+**32 days**): [commit 83e37e1](#)
- 2017-08-18 (+**32 days**): [commit 93d0cb5](#)
- 2017-08-18 (+**32 days**): [commit ec4ab09](#)
- 2017-09-06 (+**51 days**): [commit 297516e](#)
- 2017-09-17 (+**62 days**): Python 2.7.14 released
- 2017-09-19 (+**64 days**): Python 3.3.7 released
- 2017-09-24 (+**69 days**): [commit 86a713c](#)
- 2017-09-25 (+**70 days**): [commit f2492bb](#)
- 2017-10-03 (+**78 days**): Python 3.6.3 released
- 2018-02-04 (+**202 days**): Python 3.4.8 released
- 2018-02-04 (+**202 days**): Python 3.5.5 released
- 2018-06-28: Python 3.7.0 released

1.1.5 Environment variables injection in subprocess on Windows

On Windows, prevent passing invalid environment variables and command arguments to subprocess.Popen.

It is possible to inject an environment variable in subprocess on Windows if a user data is passed to a subprocess via environment variable.

Check for invalid environment (variable names containing '=') and command arguments (containing '0').

- Disclosure date: **2017-06-22** (Python issue [bpo-30730](#) reported)

Fixed In

- Python **2.7.14** (2017-09-17) fixed by [commit 9dda2ca](#) (2017-06-24)
- Python **3.3.7** (2017-09-19) fixed by [commit e46f1c1](#) (2017-07-19)
- Python **3.4.7** (2017-08-09) fixed by [commit fe82c46](#) (2017-07-11)
- Python **3.5.4** (2017-08-08) fixed by [commit a7c0264](#) (2017-06-23)
- Python **3.6.2** (2017-07-17) fixed by [commit a9b16cf](#) (2017-06-23)

Python issue

[security] Injecting environment variable in subprocess on Windows.

- Python issue: [bpo-30730](#)
- Creation date: 2017-06-22
- Reporter: Serhiy Storchaka

Timeline

Timeline using the disclosure date **2017-06-22** as reference:

- 2017-06-22: Python issue [bpo-30730](#) reported by Serhiy Storchaka
- 2017-06-23 (+1 days): [commit a7c0264](#)
- 2017-06-23 (+1 days): [commit a9b16cf](#)
- 2017-06-24 (+2 days): [commit 9dda2ca](#)
- 2017-07-11 (+19 days): [commit fe82c46](#)
- 2017-07-17 (+25 days): Python 3.6.2 released
- 2017-07-19 (+27 days): [commit e46f1c1](#)
- 2017-08-08 (+47 days): Python 3.5.4 released
- 2017-08-09 (+48 days): Python 3.4.7 released
- 2017-09-17 (+87 days): Python 2.7.14 released
- 2017-09-19 (+89 days): Python 3.3.7 released

1.1.6 CVE-2017-9233: Expat 2.2.1

Upgrade expat copy from 2.2.0 to 2.2.1 to get fixes of multiple security vulnerabilities including:

- CVE-2017-9233 (External entity infinite loop DoS),
- CVE-2016-9063 (Integer overflow, re-fix),
- CVE-2016-0718 (Fix regression bugs from 2.2.0's fix to CVE-2016-0718)
- CVE-2012-0876 (Counter hash flooding with SipHash).

Note: the CVE-2016-5300 (Use os-specific entropy sources like `getrandom`) doesn't impact Python, since Python already gets entropy from the OS to set the expat secret using `XML_SetHashSalt()`.

- Disclosure date: **2017-06-17** (Expat 2.2.1 release)

Fixed In

- Python **2.7.14** (2017-09-17) fixed by [commit 2ada64d](#) (2017-06-21)
- Python **3.3.7** (2017-09-19) fixed by [commit ab90986](#) (2017-07-16)
- Python **3.4.7** (2017-08-09) fixed by [commit 71572bb](#) (2017-07-12)
- Python **3.5.4** (2017-08-08) fixed by [commit 91d171b](#) (2017-06-21)
- Python **3.6.2** (2017-07-17) fixed by [commit ea1ab80](#) (2017-06-21)

Python issue

Update embedded copy of expat to 2.2.1.

- Python issue: [bpo-30694](#)
- Creation date: 2017-06-18
- Reporter: Ned Deily

Timeline

Timeline using the disclosure date **2017-06-17** as reference:

- 2017-06-17: Disclosure date (Expat 2.2.1 release)
- 2017-06-18 (**+1 days**): [Python issue bpo-30694](#) reported by Ned Deily
- 2017-06-21 (**+4 days**): [commit 2ada64d](#)
- 2017-06-21 (**+4 days**): [commit 91d171b](#)
- 2017-06-21 (**+4 days**): [commit ea1ab80](#)
- 2017-07-12 (**+25 days**): [commit 71572bb](#)
- 2017-07-16 (**+29 days**): [commit ab90986](#)
- 2017-07-17 (**+30 days**): Python 3.6.2 released
- 2017-08-08 (**+52 days**): Python 3.5.4 released
- 2017-08-09 (**+53 days**): Python 3.4.7 released
- 2017-09-17 (**+92 days**): Python 2.7.14 released
- 2017-09-19 (**+94 days**): Python 3.3.7 released

Links

- <https://libexpat.github.io/doc/cve-2017-9233/>
- https://github.com/libexpat/libexpat/blob/R_2_2_1/expat/Changes
- <https://www.cvedetails.com/cve/CVE-2017-9233/>

1.1.7 CVE-2017-1000158: PyString_DecodeEscape integer overflow

Check & prevent integer overflow in PyString_DecodeEscape.

You need to compile a 1 GiB Python file on 32-bit system for reproducing it. It is very unlikely that this can happen by accident, and it is hard to use it in security attack. If you can make the attacked program compiling a 1 GiB Python file, you perhaps have easier ways to make a harm.

- Disclosure date: **2017-06-13** (Python issue bpo-30657 reported)

Fixed In

- Python **2.7.14** (2017-09-17) fixed by [commit c3c9db8](#) (2017-06-18)
- Python **3.4.8** (2018-02-04) fixed by [commit 6c004b4](#) (2017-12-08)
- Python **3.5.5** (2018-02-04) fixed by [commit fd8614c](#) (2017-12-08)

Python issue

[security] CVE-2017-1000158: Unsafe arithmetic in PyString_DecodeEscape.

- Python issue: [bpo-30657](#)
- Creation date: 2017-06-13
- Reporter: Jay Bosamiya

CVE-2017-1000158

CPython (aka Python) up to 2.7.13 is vulnerable to an integer overflow in the PyString_DecodeEscape function in stringobject.c, resulting in heap-based buffer overflow (and possible arbitrary code execution)

- CVE ID: [CVE-2017-1000158](#)
- Published: 2017-11-17

Timeline

Timeline using the disclosure date **2017-06-13** as reference:

- 2017-06-13: [Python issue bpo-30657](#) reported by Jay Bosamiya
- 2017-06-18 (+5 days): [commit c3c9db8](#)
- 2017-09-17 (+96 days): Python 2.7.14 released
- 2017-11-17 (+157 days): CVE-2017-1000158 published
- 2017-12-08 (+178 days): [commit 6c004b4](#)
- 2017-12-08 (+178 days): [commit fd8614c](#)
- 2018-02-04 (+236 days): Python 3.4.8 released
- 2018-02-04 (+236 days): Python 3.5.5 released

1.1.8 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 bpo-30500 reported)
- Reported at: 2017-03-04 (Orange Tsai on the PSRT list)

Fixed In

- Python **2.7.14** (2017-09-17) fixed by [commit d4324ba](#) (2017-06-20)
- Python **3.3.7** (2017-09-19) fixed by [commit 052f9d6](#) (2017-07-26)
- Python **3.4.7** (2017-08-09) fixed by [commit cc54c1c](#) (2017-07-12)
- Python **3.5.4** (2017-08-08) fixed by [commit 4899d84](#) (2017-06-20)
- Python **3.6.2** (2017-07-17) fixed by [commit b0fba88](#) (2017-06-20)

Python issue

[security] urllib connects to a wrong host.

- Python issue: [bpo-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 bpo-30500](#) reported by Nam Nguyen
- 2017-06-20 (**+22 days**): [commit 4899d84](#)
- 2017-06-20 (**+22 days**): [commit b0fba88](#)
- 2017-06-20 (**+22 days**): [commit d4324ba](#)
- 2017-07-12 (**+44 days**): [commit cc54c1c](#)
- 2017-07-17 (**+49 days**): Python 3.6.2 released
- 2017-07-26 (**+58 days**): [commit 052f9d6](#)
- 2017-08-08 (**+71 days**): Python 3.5.4 released
- 2017-08-09 (**+72 days**): Python 3.4.7 released
- 2017-09-17 (**+111 days**): Python 2.7.14 released
- 2017-09-19 (**+113 days**): Python 3.3.7 released

1.1.9 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)

Fixed In

- Python **2.7.14** (2017-09-17) fixed by [commit e5eae47](#) (2017-07-26)
- Python **3.3.7** (2017-09-19) fixed by [commit a4e774f](#) (2017-07-26)
- Python **3.4.7** (2017-08-09) fixed by [commit 2a5a26c](#) (2017-07-27)
- Python **3.5.4** (2017-08-08) fixed by [commit 19b2890](#) (2017-07-26)
- Python **3.6.3** (2017-10-03) fixed by [commit 8c2d4cf](#) (2017-07-26)
- Python **3.7.0** (2018-06-28) fixed by [commit 2b1e6e9](#) (2017-07-22)

Python issue

(ftplib) A remote attacker could possibly attack by containing the newline characters.

- Python issue: [bpo-30119](#)
- Creation date: 2017-04-20
- Reporter: Dong-hee Na

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)
- 2017-04-20 (**+59 days**): Python issue [bpo-30119](#) reported by Dong-hee Na
- 2017-07-22 (**+152 days**): [commit 2b1e6e9](#)
- 2017-07-26 (**+156 days**): [commit 19b2890](#)
- 2017-07-26 (**+156 days**): [commit 8c2d4cf](#)
- 2017-07-26 (**+156 days**): [commit a4e774f](#)
- 2017-07-26 (**+156 days**): [commit e5eae47](#)
- 2017-07-27 (**+157 days**): [commit 2a5a26c](#)
- 2017-08-08 (**+169 days**): Python 3.5.4 released
- 2017-08-09 (**+170 days**): Python 3.4.7 released
- 2017-09-17 (**+209 days**): Python 2.7.14 released
- 2017-09-19 (**+211 days**): Python 3.3.7 released

- 2017-10-03 (+225 days): Python 3.6.3 released
- 2018-06-28: Python 3.7.0 released

Links

- <http://blog.blindspotsecurity.com/2017/02/advisory-javapython-ftp-injections.html>
- <http://www.openwall.com/lists/oss-security/2017/02/20/1>
- https://bugzilla.redhat.com/show_bug.cgi?id=1478916

1.1.10 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 bpo-29591 reported)
- Reported by: 2016-05-27 (expat bug #537 reported)

Fixed In

- Python **2.7.14** (2017-09-17) fixed by [commit 0e4571a](#) (2017-06-15)
- Python **3.3.7** (2017-09-19) fixed by [commit ab90986](#) (2017-07-16)
- Python **3.4.7** (2017-08-09) fixed by [commit 71572bb](#) (2017-07-12)
- Python **3.5.4** (2017-08-08) fixed by [commit 8c797ed](#) (2017-06-15)
- Python **3.6.2** (2017-07-17) fixed by [commit 86b9537](#) (2017-06-14)

Python issue

expat 2.2.0: Various security vulnerabilities in bundled expat (CVE-2016-0718 and CVE-2016-4472).

- Python issue: [bpo-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 [bpo-29591](#) reported by Natanael Copa
- 2017-06-14 (+**117 days**): [commit 86b9537](#)
- 2017-06-15 (+**118 days**): [commit 0e4571a](#)
- 2017-06-15 (+**118 days**): [commit 8c797ed](#)
- 2017-07-12 (+**145 days**): [commit 71572bb](#)
- 2017-07-16 (+**149 days**): [commit ab90986](#)
- 2017-07-17 (+**150 days**): Python 3.6.2 released
- 2017-08-08 (+**172 days**): Python 3.5.4 released
- 2017-08-09 (+**173 days**): Python 3.4.7 released
- 2017-09-17 (+**212 days**): Python 2.7.14 released
- 2017-09-19 (+**214 days**): Python 3.3.7 released

Links

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

1.1.11 update zlib to 1.2.11

These are the changes updating zlib from 1.2.8 to 1.2.10. It is only used when building without a system zlib.

The new release includes fixes for security issues CVE-2016-9840, CVE-2016-9841, CVE-2016-9842, CVE-2016-9843.

Note: Only Windows and macOS are affected by this issue. Linux packages use the system zlib.

- Disclosure date: **2017-01-05** (Python issue [bpo-29169](#) reported)
- Reported at: 2017-01-02 (zlib 1.2.10 released)

Fixed In

- Python **2.7.14** (2017-09-17) fixed by [commit 80b24a9](#) (2017-01-31)
- Python **3.4.8** (2018-02-04) fixed by [commit d0e61bd](#) (2017-08-16)
- Python **3.5.4** (2017-08-08) fixed by [commit 34e7e2e](#) (2017-01-31)
- Python **3.6.1** (2017-03-21) fixed by [commit 34e7e2e](#) (2017-01-31)
- Python **3.7.0** (2018-06-28) fixed by [commit 34e7e2e](#) (2017-01-31)

Python issue

update zlib to 1.2.11.

- Python issue: [bpo-29169](#)
- Creation date: 2017-01-05
- Reporter: Matthias Klose

Timeline

Timeline using the disclosure date **2017-01-05** as reference:

- 2017-01-02 (**-3 days**): Reported (zlib 1.2.10 released)
- 2017-01-05: Python issue [bpo-29169](#) reported by Matthias Klose
- 2017-01-31 (**+26 days**): [commit 34e7e2e](#)
- 2017-01-31 (**+26 days**): [commit 80b24a9](#)
- 2017-03-21 (**+75 days**): Python 3.6.1 released
- 2017-08-08 (**+215 days**): Python 3.5.4 released
- 2017-08-16 (**+223 days**): [commit d0e61bd](#)
- 2017-09-17 (**+255 days**): Python 2.7.14 released
- 2018-02-04 (**+395 days**): Python 3.4.8 released
- 2018-06-28: Python 3.7.0 released

1.1.12 Issue #28563: `gettext.c2py()`

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

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

Fixed In

- Python **2.7.13** (2016-12-17) fixed by [commit a876027](#) (2016-11-08)
- Python **3.3.7** (2017-09-19) fixed by [commit 07bcf05](#) (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)

Python issue

Arbitrary code execution in `gettext.c2py`.

- Python issue: [bpo-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 [bpo-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
- 2017-09-19 (+324 days): Python 3.3.7 released

Links

- <https://www.xil.se/post/is-eval-safe-yet-rspkt/>

1.1.13 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 **3.4.7** (2017-08-09) fixed by [commit fa53dbd](#) (2017-03-10)
- 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)

Python issue

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

- Python issue: [bpo-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 bpo-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
- 2017-03-10 (+198 days): [commit fa53dbd](#)
- 2017-08-09 (+350 days): Python 3.4.7 released

Links

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

1.1.14 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 [bpo-27568](#) reported)
- Reported by: Scott Geary (HTTPoxy)

Fixed In

- Python **2.7.13** (2016-12-17) fixed by [commit 75d7b61](#) (2016-07-30)
- Python **3.3.7** (2017-09-19) fixed by [commit 4cbb23f](#) (2016-07-31)
- 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)

Python issue

“HTTPOxy”, use of HTTP_PROXY flag supplied by attacker in CGI scripts.

- Python issue: [bpo-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 bpo-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
- 2017-09-19 (+428 days): Python 3.3.7 released

Links

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

1.1.15 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 **3.3.7** (2017-09-19) fixed by [commit 3625f7f](#) (2017-07-19)
- 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)

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
- 2017-07-19 (**+403 days**): [commit 3625f7f](#)
- 2017-09-19 (**+465 days**): Python 3.3.7 released

Links

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

1.1.16 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 bpo-26657 reported)

Fixed In

- Python **2.7.12** (2016-06-28) fixed by [commit 0cf2cf2](#) (2016-04-18)
- Python **3.3.7** (2017-09-19) fixed by [commit 7b92f9f](#) (2017-07-26)
- Python **3.4.7** (2017-08-09) fixed by [commit 6f6bc1d](#) (2017-07-12)
- 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)

Python issue

Directory traversal with `http.server` and `SimpleHTTPServer` on windows.

- Python issue: [bpo-26657](#)
- Creation date: 2016-03-28
- Reporter: Thomas

Timeline

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

- 2016-03-28: [Python issue bpo-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
- 2017-07-12 (+471 days): [commit 6f6bc1d](#)
- 2017-07-26 (+485 days): [commit 7b92f9f](#)
- 2017-08-09 (+499 days): Python 3.4.7 released
- 2017-09-19 (+540 days): Python 3.3.7 released

1.1.17 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 bpo-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 **3.3.7** (2017-09-19) fixed by [commit ab90986](#) (2017-07-16)
- 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)

Python issue

Update expat to 2.1.1.

- Python issue: [bpo-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 bpo-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
- 2017-07-16 (**+489 days**): [commit ab90986](#)
- 2017-09-19 (**+554 days**): Python 3.3.7 released

Links

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

1.1.18 CVE-2016-5636: zipimporter overflow

Heap overflow in `zipimporter` module.

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

Fixed In

- Python **2.7.12** (2016-06-28) fixed by [commit 64ea192](#) (2016-01-21)
- Python **3.3.7** (2017-09-19) fixed by [commit d751040](#) (2016-09-14)
- 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)

Python issue

heap overflow in `zipimporter` module.

- Python issue: [bpo-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 bpo-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

- 2017-09-19 (+607 days): Python 3.3.7 released

1.1.19 CVE-2016-5699: HTTP header injection

HTTP header injection in `urllib`, `urllib2`, `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 [bpo-22928](#) reported)
- Red Hat impact: Moderate

Fixed In

- Python **2.7.10** (2015-05-23) fixed by [commit 59bdf63](#) (2015-03-12)
- Python **3.3.7** (2017-09-19) fixed by [commit 8e88f6b](#) (2017-07-26)
- 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)

Python issue

HTTP header injection in `urllib2/urllib/httplib/http.client` (CVE-2016-5699).

- Python issue: [bpo-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 [bpo-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
- 2017-07-26 (+**975 days**): [commit 8e88f6b](#)
- 2017-09-19 (+**1030 days**): Python 3.3.7 released

Links

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

1.1.20 CVE-2014-9365: Validate TLS certificate

The HTTP clients in the (1) `httplib`, (2) `urllib`, (3) `urllib2`, and (4) `xmllib` 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 476 – Enabling certificate verification by default for stdlib http clients](#) and [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)

Python issue

PEP 476: verify HTTPS certificates by default.

- Python issue: [bpo-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) `xmllib` 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 [bpo-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](#)

1.1.21 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 [bpo-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: [bpo-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 bpo-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

1.1.22 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: [bpo-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 bpo-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/>

1.1.23 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 [bpo-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: [bpo-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 bpo-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

1.1.24 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 [bpo-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: [bpo-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 bpo-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

1.1.25 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 [bpo-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: [bpo-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 bpo-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

1.1.26 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 [bpo-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: [bpo-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 [bpo-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

1.1.27 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: [bpo-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 [bpo-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

1.1.28 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. Note that this function was only added to Python 2.7 in a backport to 2.7.9, and was added in its fixed form, so no releases of Python 2.7 have this vulnerability.

- Disclosure date: **2013-05-17** (Python issue [bpo-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: [bpo-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 bpo-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>

1.1.29 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 [bpo-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: [bpo-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 bpo-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

1.1.30 CVE-2013-1752: CVE-2013-1752: Limit `imaplib.IMAP4_SSL.readline()`

The `imaplib` module doesn't limit the amount of read data in its call to `IMAP4_SSL.readline()`. An erroneous or malicious IMAP server can trick the `imaplib` module to consume large amounts of memory.

- Disclosure date: **2012-09-25** (Python issue [bpo-16039](#) reported)

Vulnerable Versions

- Python **2.7**

Python issue

`imaplib`: unlimited `readline()` from connection.

- Python issue: [bpo-16039](#)
- Creation date: 2012-09-25
- Reporter: Christian Heimes

Timeline

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

- 2012-09-25: Python issue [bpo-16039](#) reported by Christian Heimes

Links

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

1.1.31 CVE-2013-1752: ftplib unlimited read

ftplib: unlimited read from connection.

- Disclosure date: **2012-09-25** (Python issue [bpo-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: [bpo-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 [bpo-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/>

1.1.32 CVE-2013-1752: nntplib unlimited read

Unlimited read from connection in nntplib.

- Disclosure date: **2012-09-25** (Python issue bpo-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 **3.3.7** (2017-09-19) 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: [bpo-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 bpo-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
- 2017-09-19 (**+1820 days**): Python 3.3.7 released

Links

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

1.1.33 CVE-2013-1752: poplib unlimited read

poplib: unlimited read from connection.

- Disclosure date: **2012-09-25** (Python issue bpo-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 **3.3.7** (2017-09-19) 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: [bpo-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 bpo-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
- 2017-09-19 (**+1820 days**): Python 3.3.7 released

Links

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

1.1.34 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 bpo-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 **3.3.7** (2017-09-19) 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: [bpo-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 bpo-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
- 2017-09-19 (**+1820 days**): Python 3.3.7 released

Links

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

1.1.35 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 bpo-16043 reported)
- Red Hat impact: Moderate

Fixed In

- Python **2.7.9** (2014-12-10) fixed by [commit 9e8f523](#) (2014-12-06)
- Python **3.3.7** (2017-09-19) fixed by [commit 4e9cefa](#) (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)

Python issue

xmlrpc: gzip_decode has unlimited read().

- Python issue: [bpo-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 [bpo-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
- 2017-09-19 (**+1820 days**): Python 3.3.7 released

Links

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

1.1.36 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 bpo-14621 reported)

Fixed In

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

Vulnerable Versions

- Python **2.7**

Python issue

Hash function is not randomized properly.

- Python issue: [bpo-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 bpo-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

1.1.37 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: [bpo-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 bpo-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

1.1.38 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 bpo-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: [bpo-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 bpo-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

1.1.39 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 [bpo-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: [bpo-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 bpo-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

1.1.40 CVE-2012-1150: Hash DoS

Hash collision denial of service.

Python 2.7 and older and Python 3.2 and older require the `-R` command line option to enable the enable hash function randomization. Randomization is enabled by default since Python 3.3 (the `-R` option is ignored).

“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.

- Ruby: CRuby 1.9 fixed the vulnerability in 2008 with randomized hash function; JRuby has also been fixed.
- Perl: Perl 5.8.1 fixed the vulnerability in 2003 using a random “`PERL_HASH_SEED`”.
- 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: [bpo-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 [bpo-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>

1.1.41 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 [bpo-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: [bpo-13512](#)
- Creation date: 2011-11-30

- Reporter: Vincent Danen

CVE-2011-4944

Python 2.6 through 3.2 creates `~/.pyirc` 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 bpo-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

1.1.42 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 [bpo-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: [bpo-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 [bpo-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

1.1.43 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 [bpo-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: [bpo-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 bpo-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

1.1.44 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 [bpo-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: [bpo-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 [bpo-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

1.1.45 CVE-2010-2089: audiopop input validation

The `audiopop` 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 `audiopop.reverse()` with a one-byte string, a different vulnerability than CVE-2010-1634.

- Disclosure date: **2010-01-11** (Python issue [bpo-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

`audiopop`: check that length is a multiple of the size.

- Python issue: [bpo-7673](#)
- Creation date: 2010-01-11
- Reporter: STINNER Victor

CVE-2010-2089

The `audiopop` 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 `audiopop.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 bpo-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

1.1.46 CVE-2013-1752: httplib unlimited read

Limit the HTTP header readline.

- Disclosure date: **2009-08-28** (Python issue [bpo-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: [bpo-6791](#)
- Creation date: 2009-08-28
- Reporter: [sumar](#)

Timeline

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

- 2009-08-28: [Python issue bpo-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/>

1.1.47 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 [bpo-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: [bpo-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 bpo-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

1.1.48 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 [bpo-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: [bpo-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 bpo-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

1.1.49 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>

1.1.50 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 bpo-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: [bpo-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 bpo-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

1.1.51 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 5bdf60](#) (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 5bdf60](#)
- 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>

1.1.52 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 [bpo-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: [bpo-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 bpo-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

1.1.53 CVE-2007-4965: rbgimg 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: [bpo-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 bpo-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

1.2 Packages and PyPI

1.2.1 Check for known vulnerabilities

- <https://github.com/pyupio/safety-db> and <https://pyup.io/>
- safety package: Safety checks your installed dependencies for known security vulnerabilities.

1.2.2 GPG

- Verifying PyPI and Conda Packages by Stuart Mumford (2016-06-21)
- Sign a package using GPG and Twine

1.2.3 pip security

- pip: Implement “hook” support for package signature verification

1.2.4 PyPI

- PEP 458 – Surviving a Compromise of PyPI (27-Sep-2013)
- PEP 480 – Surviving a Compromise of PyPI: The Maximum Security Model (8-Oct-2014)
- Making PyPI security independent of SSL/TLS by Nick Coghlan

1.2.5 Vulnerabilities in the Package Index

Index Vulnerability: Unchecked File Deletion

Improper checking of ACLs would have allowed any authenticated user to delete any release file hosted on the Package Index by supplying its md5 to the `files` action in the `pypi-legacy` code base.

- Disclosure date: **2017-10-12** (Reported via security policy on pypi.org)
- Disclosed by: [Max Justicz](#)

Fixed In

- PyPI “Legacy Codebase” (2017-10-12) fixed by [commit 18200fa](#) (2017-10-12)

Audit

After mitigating the attack vector and deploying it, the responding Package Index maintainer worked to verify that no release files had been improperly removed using this exploit.

The Package Index maintains an audit log in the form of a “Journal” for all actions initiated. It was determined that exploitation of this attack vector would still remove files via the [existing interface](#) an audit log would still be [written](#).

Using this information, we were able to reconstruct the users with access to legitimately remove release files at point in time of each file removal [using the audit log](#).

The output of this script were used to determine that no malicious actors exploited this vulnerability. All flagged journal entries were related to one of the following scenarios:

- Username updates that were not properly updated in the Journal
- Administrator intervention to remove packages

Timeline

Timeline using the disclosure date **2017-10-12** as reference:

- 2017-10-12: Issue reported by [Max Justicz](#) following guidelines in security policy on [pypi.org](#)
- 2017-10-12 (**+0days**): Report investigated by [Ernest W. Durbin III](#) and determined to be exploitable
- 2017-10-12 (**+0days**): Fix implemented and deployed in [commit 18200fa](#)
- 2017-10-12 (**+0days**): The audit journals maintained by PyPI were used to reconstruct the full history of file removals to determine that no malicious deletions were performed.

PyPI credential exposure on GitHub

Introduction

A common mistake made by users is committing and publishing “dotfiles” containing private material such as passwords, API keys, or cryptographic keys to public repositories on services such as GitHub.

Compounding this issue, the Python packaging ecosystem historically and currently encourages—albeit with some level of caution—the use of a `.pypirc` file for storage of passwords consumption by packaging tools. For a summary of the dangers of this methodology, see [this article on securing PyPI credentials](#).

With ever strengthening search tools on GitHub attackers are able to formulate queries which quickly identify and obtain credentials from such hosting sites.

- Disclosure date: **2017-11-05** (Reported via security policy on [pypi.org](#))
- Disclosed by: Joachim Jablon

Report

The PyPI security team was notified by Joachim Jablon that `.pypirc` files containing valid PyPI credentials were obtainable with a straightforward search and scrape of GitHub.

Using tools developed by the reporter the PyPI security team was able to identify 77 valid PyPI logins in 85 public files published to GitHub. These 77 logins had maintainer or administrator access to 146 unique projects on PyPI.

Audit

Action Taken by PyPI team

The PyPI security team followed up by auditing and extending the Proof of Concept tools supplied by the reporter to verify the report.

After running the tooling against the full result set of the GitHub code search the PyPI administrators unset the passphrases for all valid logins found and issued an administrative password reset for exposed users.

Additionally an audit of PyPI's journals showed no signs of malicious access for the exposed accounts.

The email sent to affected users took the form

```
From: admin@mail.pypi.python.org
To: {user['email']}
Subject: [Urgent] Your PyPI password has been reset

{username},

A security report recently identified that your PyPI login credentials were
exposed in a public code repository on github.com.

Please see the following links where your credentials were found:

{pypirc_links}

An initial audit of our journals found that {package_count} projects your
account has access to were potentially exposed but did not indicate any
malicious activity.

Packages:

{packages}

Please double check the audit logs at https://pypi.python.org after you have
reset your password and notify us if you identify any suspicious activity.

Also please reset your passwords anywhere else you may have used the password
exposed in the above links.

To reset your password, please visit {password_reset_link}.

Thanks,
PyPI Security Team
```

Recommendations

All users of PyPI should ensure that their PyPI login credentials are safe and have not been inadvertently exposed in a public repository of dotfiles, in the root of a project directory, or in some other public or shared medium.

The PyPI team does not have the resources to search or scrape all such services and may not have identified all forms of this exposure.

Additionally, reviewing the Audit Journal for your projects on pypi.python.org for suspicious activity is a good idea. If you identify any such activity, please report it per [our published security policy](#).

Timeline

Timeline using the disclosure date **2017-11-05** as reference:

- 2017-11-05 Issue reported by Joachim Jablon to a single member of the security team listed in our security policy on [pypi.org](#)
- 2017-11-08 (+3days): Issue reported by Joachim Jablon to an additional member of the security team listed in our security policy on [pypi.org](#)
- 2017-11-08 (+3days): Issue reported by Joachim Jablon to all members of the security team listed in our security policy on [pypi.org](#)
- 2017-11-08 (+3days): Report investigated by [Ernest W. Durbin III](#) and determined to be valid.
- 2017-11-09 (+4days): Administrative password resets issued.

1.2.6 PyPI typo squatting

- Typosquatting programming language package managers by Nikolai Tschacher (8 June, 2016)
- LWN: Typosquatting in package repositories (July 20, 2016)
- Building a botnet on PyPi by Steve Stagg (May 19, 2017)
- warehouse bug ([pypi.org](#)): Block package names that conflict with core libraries (reported at June 28, 2017)
- 2017-09-09: [skcsirt-sa-20170909-pypi-malicious-code](#) advisory

fate0:

- 2017-05-27 04:38 - 2017-05-31 12:24 (5 days): 10,685 downloads
- May-June, 2017
- <https://mail.python.org/pipermail/distutils-sig/2017-June/030592.html>
- <http://blog.fatezero.org/2017/06/01/package-fishing/>
- <https://github.com/pypa/pypi-legacy/issues/644>
- <http://evilpackage.fatezero.org/>
- <https://github.com/fate0/cookiecutter-evilpy-package>
- Packages (this list needs to be validated):
 - `caffe`
 - `ffmpeg`
 - `ftp`

- git
- hbase
- memcached
- mkl
- mongodb
- opencv
- openssl
- phantomjs
- proxy
- pygpu
- python-dev
- rabbitmq
- requirement.txt
- requirements.txt
- rrequirements.txt
- samba
- shadowsock
- smb
- tkinter
- vtk
- youtube-dl
- zookeeper
- ztz
- ...

See also:

- pytosquatting.org project

Example of typos:

- `urllib`, `urllib2`: part of the standard library
- `urllib3` instead of `urllib3`

1.2.7 Links

- [The Update Framework \(TUF\)](#): Like the S in HTTPS, a plug-and-play library for securing a software updater.

1.3 Python SSL and TLS security

Evolutions of the `ssl` module.

1.3.1 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'
)
```

Python 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'
```

1.3.2 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
```

(continues on next page)

(continued from previous page)

```
options |= SSL_OP_SINGLE_ECDH_USE;
#endif
SSL_CTX_set_options(self->ctx, options);
```

1.3.3 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](#)

1.3.4 SSLContext

New in Python 3.2.

1.3.5 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

1.3.6 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

1.3.7 TLS versions

- SSLv2 now black listed
- SSLv3 now black listed

1.3.8 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).

1.3.9 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`
- `boringssl`
- `multissl` (by Christian Heimes): Run Python tests against multiple installations of OpenSSL and LibreSSL

1.4 Python Security

1.4.1 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](#)

1.4.2 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.

1.4.3 Shell command injection

See https://www.owasp.org/index.php/Command_Injection

Whenever possible, avoid `subprocess.Popen(shell=True)` and `os.popen()`. On UNIX, `shlex.quote()` can be used to escape command line arguments to use them safely in a shell command.

For Windows, see:

- `subprocess.list2cmdline()` (private function)
- `distutils.spawn._nt_quote_args()` (private function)
- <https://bugs.python.org/issue8987>
- <https://bugs.python.org/issue20744>

1.4.4 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

1.4.5 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.

1.4.6 CPython Security Experts

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

1.4.7 Windows

ASLR and DEP

ASLR and DEP protections enabled since Python 3.4 (and Python 2.7.11 if built using `PCbuild/` directory).

Unsafe Python 2.7 default installation directory

Python 2.7 installer uses `C:\Python27` directory by default. The created directory has the “Modify” access rights given to the “Authenticated Users” group. An attacker can modify the standard library or even modify `python.exe`. Python 3 installer now installs Python in `C:\Program Files` by default to fix this issue. Override the default installation directory, or fix the directory permissions.

DLL injection

On Windows 8.1 and older, the installer is vulnerable to DLL injection: evil DLL written in the same download directory that the downloaded Python installer. See [DLL Hijacking Just Won't Die](#).

DLL injection using PATH

Inject a malicious DLL in a writable directory included in `PATH`. The “pip” step of the Python installer will run this DLL.

We consider that it is not an issue of Python (Python installer) itself.

Once you have write access to a directory on the system `PATH` (not the current user `PATH`) and the ability to write binaries that are not validated by the operating system before loading, there are many more interesting things you can do rather than wait for the Python installer to be run.

1.4.8 Module Search Path (`sys.path`)

- `python3 -E`: ignore `PYTHON*` environment variables like `PYTHONPATH`
- `python3 -I`: isolated mode, also implies `-E` and `-s`
- [bpo-5753: CVE-2008-5983 python: untrusted python modules search path \(2009\)](#) added `PySys_SetArgvEx()` (to Python 2.6.6, 2.7.0, 3.1.3, 3.2.0): allows embedders of the interpreter to set `sys.argv` without also modifying `sys.path`. This helps fix CVE-2008-5983.
- [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.”

1.4.9 Static analysers of CPython code base

- Coverity:
 - Coverity Scan: Python
 - devguide info about Coverity
 - analysis of 2012 by Coverity Software resulted in CPython receiving their highest quality rating.
- LGTM
- Svace static analyzer

1.4.10 Fuzzing

- Introduction to Fuzzing in Python with AFL (2015-04-13) by Alex Gaynor

1.4.11 Misc

- 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
- 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.

1.4.12 Python Security Response Team (PSRT)

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

1.4.13 Links

- Reporting security issues in Python
- Python Security Announce public mailing list
- OWASP Python Security Project (pythonsecurity.org)
- bandit: Python AST-based static analyzer from OpenStack Security Group
- Python CVEs (cvedetails.com)
- <https://gemnasium.com/>
- `owasp-pysec`: OWASP Python Security Project
- LWN: Python `ssl` module update by Christian Heimes at the Python Language Summit 2017 (during Pycon US, Portland, OR)

1.5 TODO list

TODO list for this python-security documentation.

- Get Red Hat impact from a Red Hat URL?

1.5.1 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.
```