Prevent environment variables injection in subprocess on Windows. Prevent passing other environment variables and command arguments.

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) and 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()`.

Fix urllib.parse.splithost() to correctly parse fragments. For example, `splithost('//127.0.0.1#@evil.com/')` now correctly returns the `127.0.0.1` host, instead of treating `@evil.com` as the host in an authentication (`login@host`).

Update expat copy from 2.1.1 to 2.2.0 to get fixes of CVE-2016-0718 and CVE-2016-4472. See https://sourceforge.net/p/expat/bugs/537/ for more information.

Relative import from unloaded package now reimports the package instead of failing with SystemError. Relative import from non-package now fails with ImportError rather than SystemError.

Avoid blocking in pthread_mutex_lock() when PyThread_acquire_lock() is asked not to block.

Fixed various segfaults with dict when input collections are mutated during searching, inserting or comparing. Based on patches by Duane Griffin and Tim Mitchell.

Fixed type.__setattr__() and type.__delattr__() for non-interned attribute names. Based on patch by Eryk Sun.

Fixed error messages in the index() method of tuple, list and deque when pass indices of wrong type.

`bool(range)` works even if `len(range)` raises :exc:`OverflowError`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.5.4rc1.rst`, line 112); *backlink***
>
> Unknown interpreted text role "exc".

Fix wrapping coroutine return values in StopIteration.

Restore runtime compatibility with bytecode files generated by CPython 3.5.0 to 3.5.2, and adjust the eval loop to avoid the problems that could be caused by the malformed variant of the BUILD_MAP_UNPACK_WITH_CALL opcode that they may contain. Patch by Petr Viktorin, Serhiy Storchaka, and Nick Coghlan.

Support __rmod__ for subclasses of str being called before str.__mod__. Patch by Martijn Pieters.

Fix incorrect handling of signed zeros in complex constructor for complex subclasses and for inputs having a __complex__ method. Patch by Serhiy Storchaka.

Fixed possibly dereferencing undefined pointers when creating weakref objects.

Fixed use-after-free problem in key sharing dict.

Prevent RunMainFromImporter overwriting sys.path[0].

Fixed possible BytesWarning when compare the code objects. Warnings could be emitted at compile time.

If max_line_length=None is specified while using the Compat32 policy, it is no longer ignored. Patch by Mircea Cosbuc.

Fix `unittest.mock`'s autospec to not fail on method-bound builtin functions. Patch by Aaron Gallagher.

Fix decrementing a borrowed reference in tracemalloc.

Fix multiprocessing.Queue.join_thread(): it now waits until the thread completes, even if the thread was started by the same process which created the queue.

Fix segfault in readline when using readline's history-size option. Patch by Nir Soffer.

signal.setitimer() may disable the timer when passed a tiny value. Tiny values (such as 1e-6) are valid non-zero values for setitimer(), which is specified as taking microsecond-resolution intervals. However, on some platform, our conversion routine could convert 1e-6 into a zero interval, therefore disabling the timer instead of (re-)scheduling it.

Fix bug when modifying os.environ while iterating over it

Fix email header value parser dropping folding white space in certain cases.

Update zlib to 1.2.11.

os.listdir() and os.scandir() now emit bytes names when called with bytes-like argument.

Prohibited the '=' character in environment variable names in `os.putenv()` and `os.spawn*()`.

Fixed the lgettext() family of functions in the gettext module. They now always return bytes.

Fix path calculation in imp.load_package(), fixing it for cases when a package is only shipped with bytecodes. Patch by Alexandru Ardelean.

unittest.TestCase.assertRaises() now manually breaks a reference cycle to not keep objects alive longer than expected.

inspect.signature() now supports callables with variable-argument parameters wrapped with partialmethod. Patch by Dong-hee Na.

Fixed comparison check for ipaddress.ip_interface objects. Patch by Sanjay Sundaresan.

Avoid race condition in multiprocessing cleanup.

The traceback no longer displayed for SystemExit raised in a callback registered by atexit.

Don't log exceptions if Task/Future "cancel()" method was called.

Updates to typing module: Add generic AsyncContextManager, add support for ContextManager on all versions. Original PRs by Jelle Zijlstra and Ivan Levkivskyi

Fix ssl sockets leaks when connection is aborted in asyncio/ssl implementation. Patch by Michaël Sghaïer.

Closing transport during handshake process leaks open socket. Patch by Nikolay Kim

Fix waiter cancellation in asyncio.Lock. Patch by Mathieu Sornay.

On Windows, subprocess.Popen.communicate() now also ignore EINVAL on stdin.write() if the child process is still running but closed the pipe.

Fix the problem that logging.handlers.SysLogHandler cannot handle IPv6 addresses.

Preserve generator state when _random.Random.setstate() raises an exception. Patch by Bryan Olson.

multiprocessing.Queue._feed background running thread do not break from main loop on exception.

Fix handling escape characters in HZ codec. Based on patch by Ma Lin.

Fix AttributeError when using SimpleQueue.empty() under *spawn* and *forkserver* start methods.

imaplib and poplib now catch the Windows socket WSAEINVAL error (code 10022) on shutdown(SHUT_RDWR): An invalid operation was attempted. This error occurs sometimes on SSL connections.

Warnings emitted when compile a regular expression now always point to the line in the user code. Previously they could point into inners of the re module if emitted from inside of groups or conditionals.

Fixed `Task.cancel()` can be ignored when the task is running coroutine and the coroutine returned without any more `await`.

Fix range checking in GB18030 decoder. Original patch by Ma Lin.

Change resulted because of zipfile breakage. (See also: bpo-29094)

Removed the __init__ methods of _json's scanner and encoder. Misusing them could cause memory leaks or crashes. Now scanner and encoder objects are completely initialized in the __new__ methods.

Avoid KeyboardInterrupt tracebacks in forkserver helper process when Ctrl-C is received.

Various updates to typing module: add typing.NoReturn type, use WrapperDescriptorType, minor bug-fixes. Original PRs by Jim Fasarakis-Hilliard and Ivan Levkivskyi.

Fix getsockname() for unbound AF_UNIX sockets on Linux.

Fixed leaks and crashes in errors handling in the parser module.

Fixed crashes in IOBase methods __next__() and readlines() when readline() or __next__() respectively return non-sizeable object. Fixed possible other errors caused by not checking results of PyObject_Size(), PySequence_Size(), or PyMapping_Size().

_io._IOBase.readlines will check if it's closed first when hint is present.

Fixed race condition in pathlib mkdir with flags parents=True. Patch by Armin Rigo.

Fixed arbitrary unchaining of RuntimeError exceptions in contextlib.contextmanager. Patch by Siddharth Velankar.

Pickling and copying ImportError now preserves name and path attributes.

Fix a crash in itertools.chain.from_iterable when encountering long runs of empty iterables.

Fixed multiple crashes in ElementTree caused by race conditions and wrong types.

Fixed a bug in pools in multiprocessing.pool that raising an exception at the very first of an iterable may swallow the exception or make the program hang. Patch by Davin Potts and Xiang Zhang.

Avoid incorrect errors raised by Path.mkdir(exist_ok=True) when the OS gives priority to errors such as EACCES over EEXIST.

Release references to tasks, their arguments and their results as soon as they are finished in multiprocessing.Pool.

faulthandler: Restore the old sigaltstack during teardown. Patch by Christophe Zeitouny.

Fixed crashes in repr of recursive buffered file-like objects.

Fix crashes in partial.__repr__ if the keys of partial.keywords are not strings. Patch by Michael Seifert.

get_extra_info() raises exception if get called on closed ssl transport. Patch by Nikolay Kim.

Fixed possible failing or crashing input() if attributes "encoding" or "errors" of sys.stdin or sys.stdout are not set or are not strings.

Fix a bug that prevented array 'Q', 'L' and 'I' from accepting big intables (objects that have __int__) as elements. Patch by Oren Milman.

SimpleXMLRPCDispatcher no longer chains KeyError (or any other exception) to exception(s) raised in the dispatched methods. Patch by Petr Motejlek.

asyncio.subprocess.SubprocessStreamProtocol no longer closes before all pipes are closed.

Fix asyncio to support instantiation of new event loops in child processes.

Fix assertion error in threading._DummyThread.is_alive().

Fix file object leak in aifc.open() when file is given as a filesystem path and is not in valid AIFF format. Patch by Anthony Zhang.

Fix unittest.mock._Call helper: don't ignore the name parameter anymore. Patch written by Jiajun Huang.

Altering a kwarg dictionary passed to functools.partial() no longer affects a partial object after creation.

Various updates to typing module: typing.Counter, typing.ChainMap, improved ABC caching, etc. Original PRs by Jelle Zijlstra, Ivan Levkivskyi, Manuel Krebber, and Łukasz Langa.

Fix datetime.fromtimestamp() regression introduced in Python 3.6.0: check minimum and maximum years.

Fix weakref spewing exceptions during interpreter shutdown when used with a rare combination of multiprocessing and custom codecs.

Prevent infinite loop in pathlib.Path.mkdir

Fixed out-of-bounds buffer access in the group() method of the match object. Based on patch by WGH.

Fix subprocess.Popen.wait() when the child process has exited to a stopped instead of terminated state (ex: when under ptrace).

Fix a regression in argparse that help messages would wrap at non-breaking spaces.

Fixed the comparison of mock.MagickMock with mock.ANY.

Fix an important omission by adding Deque to the typing module.

Fixed infinite recursion in the repr of uninitialized ctypes.CDLL instances.

Fixed race condition in C implementation of functools.lru_cache. KeyError could be raised when cached function with full cache was simultaneously called from different threads with the same uncached arguments.

In urllib.request, suffixes in no_proxy environment variable with leading dots could match related hostnames again (e.g. .b.c matches a.b.c). Patch by Milan Oberkirch.

Add missing attribute related constants in curses documentation.

Add missing info of code object in inspect documentation.

Link the documentation to its source file on GitHub.

Document smtpd.py as effectively deprecated and add a pointer to aiosmtpd, a third-party asyncio-based replacement.

Add canonical header link on each page to corresponding major version of the documentation. Patch by Matthias Bussonnier.

Fix Python 2 syntax in code for building the documentation.

Fix regrtest command line parser to allow passing -u extralargefile to run test_zipfile64.

regrtest: Enhance regrtest and backport features from the master branch. Add options: --coverage, --testdir, --list-tests (list test files, don't run them), --list-cases (list test identifiers, don't run them, :issue:`30523`), --matchfile (load a list of test filters from a text file, :issue:`30540`), --slowest (alias to --slow). Enhance output: add timestamp, test result, currently running tests, "Tests result: xxx" summary with total duration, etc. Fix reference leak hunting in regrtest, --huntrleaks: regrtest now warms up caches, create explicitly all internal singletons which are created on demand to prevent false positives when checking for reference leaks. (:issue:`30675`).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.5.4rc1.rst`, **line 1002**); ***backlink***

Unknown interpreted text role "issue".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.5.4rc1.rst`, **line 1002**); ***backlink***

Unknown interpreted text role "issue".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.5.4rc1.rst`, **line 1002**); ***backlink***

Unknown interpreted text role "issue".

test_thread: setUp() now uses support.threading_setup() and support.threading_cleanup() to wait until threads complete to avoid random side effects on following tests. Initial patch written by Grzegorz Grzywacz.

Skip test_asyncore and test_eintr poll failures on macOS. Skip some tests of select.poll when running on macOS due to unresolved issues with the underlying system poll function on some macOS versions.

Enhanced functions swap_attr() and swap_item() in the test.support module. They now work when delete replaced attribute or item inside the with statement. The old value of the attribute or item (or None if it doesn't exist) now will be assigned to the target of the "as" clause, if there is one.

to match the behaviour of the `re.LOCALE` flag, test_re.test_locale_flag now uses `locale.getpreferredencoding(False)` to determine the candidate encoding for the test regex (allowing it to correctly skip the test when the default locale encoding is a multi-byte encoding)

Prevent unnecessary rebuilding of Python during `make test`, `make install` and some other make targets when configured with `--enable-optimizations`.

Don't regenerate generated files based on file modification time anymore: the action is now explicit. Replace `make touch` with `make regen-all`.

Fix `--enable-optimization` didn't work.

Locate msbuild.exe on Windows when building rather than vcvarsall.bat

Prevent crash when passing invalid arguments into msvcrt module.

Function PySlice_GetIndicesEx() is replaced with a macro if Py_LIMITED_API is set to the value between 0x03050400 and 0x03060000 (not including) or 0x03060100 or higher.

Fixed the declaration of some public API functions. PyArg_VaParse() and PyArg_VaParseTupleAndKeywords() were not available in limited API. PyArg_ValidateKeywordArguments(), PyArg_UnpackTuple() and Py_BuildValue() were not available in limited API of version < 3.3 when PY_SSIZE_T_CLEAN is defined.