CVE-2021-3426: Remove the `getfile` feature of the :mod:`pydoc` module which could be abused to read arbitrary files on the disk (directory traversal vulnerability). Moreover, even source code of Python modules can contain sensitive data like passwords. Vulnerability reported by David Schwörer.

:mod:`ftplib` no longer trusts the IP address value returned from the server in response to the PASV command by default. This prevents a malicious FTP server from using the response to probe IPv4 address and port combinations on the client network.

Code that requires the former vulnerable behavior may set a `trust_server_pasv_ipv4_address` attribute on their :class:`ftplib.FTP` instances to `True` to re-enable it.

Add audit hooks for :func:`gc.get_objects`, :func:`gc.get_referrers` and :func:`gc.get_referents`. Patch by Pablo Galindo.

Update CPython bytecode magic number.

Raise ImportWarning when calling find_loader().

Fix crash that happens when replacing `sys.stderr` with a callable that can remove the object while an exception is being printed. Patch by Pablo Galindo.

The bytecode interpreter uses instruction, rather byte, offsets internally. This reduces the number of EXTENDED_ARG instructions needed and streamlines instruction dispatch a bit.

Fix reference leak in the :mod:`_hashopenssl` extension. Patch by Pablo Galindo.

Calls to find_module() by the import system now raise ImportWarning.

Improve the syntax error for invalid usage of double starred elements ('**') in f-strings. Patch by Pablo Galindo.

Speed up calls to `map()` by using the PEP 590 `vectorcall` calling convention. Patch by Dong-hee Na.

The import system now prefers using `__spec__` for `ModuleType.__repr__` over `module_repr()`.

Added micro-optimizations to `_PyType_Lookup()` to improve cache lookup performance in the common case of cache hits.

Report the column offset for :exc:`SyntaxError` for invalid line continuation characters. Patch by Pablo Galindo.

Fix misdetection of circular imports when using `from pkg.mod import attr`, which caused false positives in non-trivial multi-threaded code.

Emit SyntaxWarnings for assertions with tuple constants, this is a regression introduced in python3.7

Tracing now has correct line numbers for attribute accesses when the attribute is on a different line from the object. Improves debugging and profiling for multi-line method chains.

Python no longer fails at startup with a fatal error if a command line argument contains an invalid Unicode character. The :c:func:`Py_DecodeLocale` function now escapes byte sequences which would be decoded as Unicode characters outside the [U+0000; U+10ffff] range.

Fix a bug that was causing the parser to crash when emitting syntax errors when reading input from stdin. Patch by Pablo Galindo

Fix a possible race condition where `PyErr_CheckSignals` tries to execute a non-Python signal handler.

Add `__match_args__` to :c:type:`structsequence` based classes. Patch by Pablo Galindo.

CPython now sets the `SA_ONSTACK` flag in `PyOS_setsig` for the VM's default signal handlers. This is friendlier to other in-process code that an extension module or embedding use could pull in (such as Golang's cgo) where tiny thread stacks are the norm and `sigaltstack()` has been used to provide for signal handlers. This is a no-op change for the vast majority of processes that don't use sigaltstack.

Speed up calls to `filter()` by using the PEP 590 `vectorcall` calling convention. Patch by Dong-hee Na.

Add a radix tree based memory map to track in-use obmalloc arenas. Use to replace the old implementation of address_in_range(). The radix tree approach makes it easy to increase pool sizes beyond the OS page size. Boosting the pool and arena size allows obmalloc to handle a significantly higher percentage of requests from its ultra-fast paths.

It also has the advantage of eliminating the memory unsanitary behavior of the previous address_in_range(). The old address_in_range() was marked with the annotations _Py_NO_SANITIZE_ADDRESS, _Py_NO_SANITIZE_THREAD, and _Py_NO_SANITIZE_MEMORY. Those annotations are no longer needed.

To disable the radix tree map, set a preprocessor flag as follows: *-DWITH_PYMALLOC_RADIX_TREE=0*.

Co-authored-by: Tim Peters <tim.peters@gmail.com>

Only handle asynchronous exceptions and requests to drop the GIL when returning from a call or on the back edges of loops. Makes sure that :meth:`__exit__` is always called in with statements, even for interrupts.

Document various stdlib deprecations in imp, pkgutil, and importlib.util for removal in Python 3.12.

:class:`xmlrpc.client.ServerProxy` no longer ignores query and fragment in the URL of the server.

The :meth:`~array.array.index` method of :class:`array.array` now has optional *start* and *stop* parameters.

Enum: adjust `repr()` to show only enum and member name (not value, nor angle brackets) and `str()` to show only member name. Update and improve documentation to match.

Deprecate all module_repr() methods found in importlib as their use is being phased out by Python 3.12.

Raising an exception raised in a "future" instance will create reference cycles.

Finish updating the vendored libmpdec to version 2.5.1. Patch by Stefan Krah.

Revert the _decimal C API which was added in bpo-41324.

Fix deadlock when using :class:`ssl.SSLContext` debug callback with :meth:`ssl.SSLContext.sni_callback`.

It's now possible to create MPTCP sockets with IPPROTO_MPTCP

`image/heic` and `image/heif` were added to :mod:`mimetypes`.

The :mod:`hmac` module now uses OpenSSL's HMAC implementation when digestmod argument is a hash name or builtin hash function.

Implement PEP 597: Add `EncodingWarning` warning, `-X warn_default_encoding` option, :envvar:`PYTHONWARNDEFAULTENCODING` environment variable and `encoding="locale"` argument value.

`ast.unparse` can now render NaNs and empty sets.

:func:`pprint.pprint` gains a new boolean `underscore_numbers` optional argument to emit integers with thousands separated by an underscore character for improved readability (for example `1_000_000` instead of `1000000`).

:meth:`~collections.deque.rotate` calls are now slightly faster due to faster argument parsing.

:func:`subprocess.communicate` no longer raises an IndexError when there is an empty stdout or stderr IO buffer during a timeout on Windows.

Fixed long-standing bug of smtplib.SMTP where doing AUTH LOGIN with initial_response_ok=False will fail.

The cause is that SMTP.auth_login _always_ returns a password if provided with a challenge string, thus non-compliant with the standard for AUTH LOGIN.

Also fixes bug with the test for smtpd.

Add frozen modules to :data:`sys.stdlib_module_names`. For example, add "_frozen_importlib" and "_frozen_importlib_external" names.

Add keyword arguments support to ChainMap.new_child().

Add optional parameter *ignore_cleanup_errors* to :func:`tempfile.TemporaryDirectory` and allow multiple :func:`cleanup` attempts. Contributed by C.A.M. Gerlach.

Include changes from importlib_metadata 3.7:

Performance enhancements to distribution discovery.

entry_points only returns unique distributions.

Introduces new EntryPoints object for containing a set of entry points with convenience methods for selecting entry points by group or name. entry_points now returns this object if selection parameters are supplied but continues to return a dict object for compatibility. Users are encouraged to rely on the selection interface. The dict object result is likely to be deprecated in the future.

Added packages_distributions function to return a mapping of packages to the distributions that provide them.

Improves the networking efficiency of :mod:`http.client` when using a proxy via :meth:`~HTTPConnection.set_tunnel`. Fewer small send calls are made during connection setup.

Improve performance of :class:`fractions.Fraction` arithmetics for large components. Contributed by Sergey B. Kirpichev.

Unknown interpreted text role "class".

Allow passing a signal number to `_thread.interrupt_main()`.

Fix `ElementTree.extend` not working on iterators when using the Python implementation

Improve :mod:`sqlite3` error handling: If `sqlite3_column_text()` and `sqlite3_column_blob()` set `SQLITE_NOMEM`, :exc:`MemoryError` is now raised. Patch by Erlend E. Aasland.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.10.0a7.rst`, **line 566**); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.10.0a7.rst`, **line 566**); *backlink*
>
> Unknown interpreted text role "exc".

Fix a regression introduced in GH-24562, where an empty bytestring was fetched as `None` instead of `b''` in :mod:`sqlite3`. Patch by Mariusz Felisiak.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.10.0a7.rst`, **line 577**); *backlink*
>
> Unknown interpreted text role "mod".

Fixed stacklevel of `DeprecationWarning` emitted from `import distutils`.

`importlib.resources` now honors namespace packages, merging resources from each location in the namespace as introduced in `importlib_resources` 3.2 and including incidental changes through 5.0.3.

:meth:`datetime.datetime.strptime` now raises `ValueError` instead of `IndexError` when matching `'z'` with the `%z` format specifier.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.10.0a7.rst`, **line 609**); *backlink*
>
> Unknown interpreted text role "meth".

Return empty string if base64mime.body_encode receive empty bytes

:func:`curses.window.enclose` returns now `True` or `False` (as was documented) instead of `1` or `0`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.10.0a7.rst`, **line 628**); *backlink*
>
> Unknown interpreted text role "func".

Add MIME types for opus, AAC, 3gpp and 3gpp2

Add an invalidate_caches() method to the zipimport.zipimporter class to support importlib.invalidate_caches(). Patch by Desmond Cheong.

Fail fast in :func:`shutil.move()` to avoid creating destination directories on failure.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.10.0a7.rst`, **line 657**); *backlink*
>
> Unknown interpreted text role "func".

Enum's *repr()* and *str()* have changed: *repr()* is now *EnumClass.MemberName* and *str()* is *MemberName*. Additionally, stdlib Enum's whose contents are available as module attributes, such as *RegexFlag.IGNORECASE*, have their *repr()* as *module.name*, e.g. *re.IGNORECASE*.

Fixed bug where the :mod:`pdb` interactive run command echoed the args from the shell command line, even if those have been overridden at the pdb prompt.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.10.0a7.rst`, **line 680**); *backlink*

Unknown interpreted text role "mod".

Fixed bug where breakpoints did not persist across multiple debugger sessions in :mod:`pdb`'s interactive mode.

When the :data:`tempfile.tempdir` global variable is set to a value of type bytes, it is now handled consistently. Previously exceptions could be raised from some tempfile APIs when the directory did not already exist in this situation. Also ensures that the :func:`tempfile.gettempdir()` and :func:`tempfile.gettempdirb()` functions *always* return str and bytes respectively.

Expose X509_V_FLAG_ALLOW_PROXY_CERTS as :data:`~ssl.VERIFY_ALLOW_PROXY_CERTS` to allow proxy certificate validation as explained in https://www.openssl.org/docs/man1.1.1/man7/proxy-certificates.html.

Add builtins.aiter and builtins.anext. Patch by Joshua Bronson (@jab), Daniel Pope (@lordmauve), and Justin Wang (@justin39).

Answer "Why is there no goto?" in the Design and History FAQ.

Clarified that a result from :func:`time.monotonic`, :func:`time.perf_counter`, :func:`time.process_time`, or :func:`time.thread_time` can be compared with the result from any following call to the same function - not just the next immediate call.

Fix type documentation for Fault.faultCode; the type has to be int instead of str.

Clarified wording of s * n in the Common Sequence Operations

Fix test_getsetlocale_issue1813() of test_locale: skip the test if `setlocale()` fails. Patch by Victor Stinner.

Add workaround for Ubuntu's custom OpenSSL security level policy.

Introduce and correctly use ALIGNOF_X in place of SIZEOF_X for alignment-related code in optimized string routines. Patch by Jessica Clarke.

Update macOS, Windows, and CI to OpenSSL 1.1.1k.

Improve configure.ac: Check for presence of autoconf-archive package and remove our copies of M4 macros.

The `configure` script now supports `--with-openssl-rpath` option.

Use `_freeze_importlib` to generate code for the `__hello__` module. This approach ensures the code matches the interpreter version. Previously, PYTHON_FOR_REGEN was used to generate the code, which might be wrong. The marshal format for code objects has changed with bpo-42246, commit 877df851. Update the code and the expected code sizes in ctypes test_frozentable.

Build :mod:`sqlite3` with the `R*Tree` module enabled. Patch by Erlend E. Aasland.

Document that IDLE can fail on Unix either from misconfigured IP masquerade rules or failure displaying complex colored (non-ascii) characters.

The limited C API is now supported if Python is built in debug mode (if the `Py_DEBUG` macro is defined). In the limited C API, the :c:func:`Py_INCREF` and :c:func:`Py_DECREF` functions are now implemented as opaque function calls, rather than accessing directly the :c:member:`PyObject.ob_refcnt` member, if Python is built in debug mode and the `Py_LIMITED_API` macro targets Python 3.10 or newer. It became possible to support the limited C API in debug mode because the :c:type:`PyObject` structure is the same in release and debug mode since Python 3.8 (see :issue:`36465`).

The limited C API is still not supported in the `--with-trace-refs` special build (`Py_TRACE_REFS` macro).

Patch by Victor Stinner.

Remove the `pyarena.h` header file with functions:

- `PyArena_New()`
- `PyArena_Free()`
- `PyArena_Malloc()`
- `PyArena_AddPyObject()`

These functions were undocumented, excluded from the limited C API, and were only used internally by the compiler. Patch by Victor Stinner.

Remove the compiler and parser functions using `struct _mod` type, because the public AST C API was removed:

- `PyAST_Compile()`
- `PyAST_CompileEx()`
- `PyAST_CompileObject()`
- `PyFuture_FromAST()`
- `PyFuture_FromASTObject()`
- `PyParser_ASTFromFile()`
- `PyParser_ASTFromFileObject()`
- `PyParser_ASTFromFilename()`
- `PyParser_ASTFromString()`
- `PyParser_ASTFromStringObject()`

These functions were undocumented and excluded from the limited C API. Patch by Victor Stinner.

Remove `ast.h`, `asdl.h`, and `Python-ast.h` header files. These functions were undocumented and excluded from the limited C API. Most names defined by these header files were not prefixed by `Py` and so could create names conflicts. For example, `Python-ast.h` defined a `Yield` macro which was conflict with the `Yield` name used by the Windows `<winbase.h>` header. Use the Python :mod:`ast` module instead. Patch by Victor Stinner.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.10.0a7.rst`, line 931);** *backlink*
>
> Unknown interpreted text role "mod".

Fix a `PyEval_EvalCodeEx()` regression: fix reference counting on builtins. Patch by Victor Stinner.

Remove the `symtable.h` header file and the undocumented functions:

- `PyST_GetScope()`
- `PySymtable_Build()`
- `PySymtable_BuildObject()`
- `PySymtable_Free()`
- `Py_SymtableString()`
- `Py_SymtableStringObject()`

The `Py_SymtableString()` function was part the stable ABI by mistake but it could not be used, because the `symtable.h` header file was excluded from the limited C API.

The Python :mod:`symtable` module remains available and is unchanged.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.10.0a7.rst`, line 969);** *backlink*
>
> Unknown interpreted text role "mod".

Patch by Victor Stinner.

Remove the `PyAST_Validate()` function. It is no longer possible to build a AST object (`mod_ty` type) with the public C API. The function was already excluded from the limited C API (PEP 384). Patch by Victor Stinner.