APPENDIX

A. Instrumentation Strategies Comparison

Strategy 1: Python-level Monkey Patching. Instrumentation using monkey patching involves dynamically modifying the behavior of a function or class method at runtime by replacing it with a wrapper that triggers events before and after calling the original. This is done by changing the function pointer to point to the wrapper, leaving the external interface unchanged.

Pros:

- Precisely targets specific functions without affecting unrelated ones.
- Fast and lightweight—only reassigns a function reference.
- Preserves the external API and return values of the function, reducing the chance of bugs.

Cons:

- Cannot patch built-in read-only types such as `list` and `dict` in CPython, which are implemented in C.
- Cannot instrument literals like `[]`, `{}`, `+`, `-`, `==`, `<`, etc.
- Cannot handle control-flow constructs like `for` loops or `if` statements.
- Instrumentation applies globally and cannot be toggled per module.

We found a partial workaround to the first limitation: by replacing references to read-only classes with references to a subclass, we can monkey-patch the subclass. However, this only affects newly created objects—not existing ones (e.g., from dependencies or the Python standard library). Literals like `[]` still point to the original built-in types and remain uninstrumented.

Strategy 2: Python-level Monkey Patching + C-level Monkey Patching via `forbiddenfruit` and `ctypes`. This strategy extends monkey patching using the `forbiddenfruit` library [1], which leverages CPython's `ctypes` API [2] to enable patching of built-in read-only types. It mitigates several limitations of strategy 1 by allowing direct manipulation of C-level type slots.

Pros:

- Enables instrumentation of built-in read-only types like `list`, `dict`, and `set`.
- Affects literals such as `[]`, `{}`, and `{"key": value}` directly.

Cons:

- Cannot instrument some CPython-optimized operations (e.g., `1 + 2`) or certain `__init__` methods that bypass Python-level dispatch due to C-level shortcuts.
- Lacks support for some methods like comparison dunder methods such as `__lt__`, `__eq__`, `__ne__`, although we suspect that such support is possible. we opened an issue to confirm that assumption with the maintainers [3].
- Works only with CPython and is incompatible with other Python implementations.

Strategy 3: Monkey Patching + AST Rewriting. This strategy modifies source code at the AST (Abstract Syntax Tree) level before execution. Using `sys.meta_path` [4], we intercept module imports and dynamically rewrite their AST just before the module is executed. We use this to replace literals, built-ins, or function calls with instrumented versions. Unlike offline instrumentation, runtime AST rewriting avoids redundant disk I/O and ensures that only the actually imported code is transformed.

In our implementation, this is achieved by wrapping the existing path finders in `sys.meta_path` with new ones that delegate most of their functionality to the original ones, but override parts responsible for loading the modules. This ensures compatibility with Python's standard module resolution and preserves import behavior. The wrappers return custom loaders that parse and transform the module's AST, injecting instrumentation selectively. This mechanism is initialized early using `sitecustomize.py` [5], ensuring that the transformation hooks are active before any user-defined code is imported.

Pros:

- Can instrument literals (`[]`, `{}`), CPython-optimized expressions like `1 + 2`, and implicit calls like `__init__`.
- Supports instrumentation of read-only built-in types.
- Enables instrumentation of dunder methods and control-flow constructs (`if`, `for`).
- Allows selective instrumentation of parts of the source code.

Cons:

- Slower due to the cost of AST parsing and transformation.
- May introduce bugs if AST transformations are not applied carefully and conservatively.

Comparison. Different instrumentation strategies are suitable for different scenarios. Strategy 1 (pure monkey patching) is ideal for instrumenting regular Python functions and classes in a lightweight and portable way. Strategy 2 (monkey patching with `ctypes`) is useful when monitoring built-in types and their literals is required. Finally, Strategy 3 (AST rewriting) offers the most comprehensive coverage, including literals, control flow, and selective instrumentation, and should be used when deep instrumentation is required or when only certain modules need to be monitored without affecting the rest of the system.

[1] https://github.com/clarete/forbiddenfruit
[2] https://docs.python.org/3/library/ctypes.html
[3] https://github.com/clarete/forbiddenfruit/issues/80
[4] https://docs.python.org/3/library/sys.html#sys.meta_path
[5] https://docs.python.org/3/library/site.html#module-sitecustomize

Capability	Strategy 1	Strategy 2	Strategy 3
Instrument regular Python functions	✓	✓	✓
Instrument built-in read-only types	✗	✓	✓
Instrument literals (`[]`, `1+2`)	✗	✓*	✓
Instrument comparison dunders (`__eq__`, etc.)	✗	✗	✓
Instrument control flow (`for`, `if`)	✗	✗	✓
Selective instrumentation per module	✗	✗	✓
Cross-implementation support (e.g., PyPy)	✓	✗	✓
Ease of implementation	✓	**Medium**	**Medium**
Performance overhead	✓	✓	✗

TABLE I: Comparison of instrumentation strategies in PyMOP.

* C-level Monkey Patching via `forbiddenfruit` can handle some literals like list literals `[]` but not others like integer literal addition `1+2`.

Discussion. Implementing instrumentation across the Python stack introduced several challenges. Monkey patching is lightweight but cannot intercept operations on literals or built-in types. C-level patching extends coverage, though it relies on CPython internals and remains incomplete in areas like comparison dunders. AST rewriting enables deeper instrumentation, including control flow and implicit operations, but comes with transformation complexity and runtime overhead.

We also experimented with implementing instrumentation directly in C as a Python extension. While this approach worked reliably, we ultimately used the `forbiddenfruit` library, which achieves similar effects via CPython's `ctypes` API, offering more flexibility without requiring compilation or deployment of native extensions.

In parallel, Python provides built-in instrumentation hooks such as `sys.setprofile` [6], which triggers callbacks on every function call, return, or exception. While useful for general tracing, it does not capture low-level operations on literals or built-in types. Similarly, Runtime Audit Hooks[7] allow inspection of various runtime events, but lack the selectivity and context awareness needed to monitor specific program behaviors without substantial post-processing.

[6]https://docs.python.org/3/library/sys.html
[7]https://peps.python.org/pep-0578/

B. RQ1 - Statistical Analysis Results for Algorithms Comparison

TABLE II: Statistical tests. Columns 2–3 show performance difference between pairs. Columns 4–5 show effect sizes of pairwise comparisons as the rank-biserial correlation coefficient; higher X vs. Y mean higher effect sizes of Y. Data: all projects except those without RV takes less than 2s. (628 projects)

Pair	p**-value**	**Level**	**Rank-Biserial r**	**Magnitude**
$\mathbb{B}\langle X\rangle$ vs. $\mathbb{C}\langle X\rangle$	2.17e-100	High	0.9854	Large
$\mathbb{B}\langle X\rangle$ vs. $\mathbb{C}^{+}\langle X\rangle$	6.07e-100	High	0.9832	Large
$\mathbb{B}\langle X\rangle$ vs. $\mathbb{D}\langle X\rangle$	2.60e-101	High	0.9900	Large
$\mathbb{B}\langle X\rangle$ vs. Baseline	2.34e-103	High	n/a	n/a
$\mathbb{C}\langle X\rangle$ vs. $\mathbb{C}^{+}\langle X\rangle$	1.00	Insignificant	0.0169	Negligible
$\mathbb{C}\langle X\rangle$ vs. $\mathbb{D}\langle X\rangle$	1.10e-81	High	0.8879	Large
$\mathbb{C}^{+}\langle X\rangle$ vs. $\mathbb{D}\langle X\rangle$	8.82e-91	High	0.9396	Large
$\mathbb{D}\langle X\rangle$ vs. Baseline	12.34e-103	High	n/a	n/a

TABLE III: Statistical tests. Columns 2–3 show performance difference between pairs. Columns 4–5 show effect sizes of pairwise comparisons as the rank-biserial correlation coefficient; (All projects)

Pair	p**-value**	**Level**	**Rank-Biserial r**	**Magnitude**
$\mathbb{B}\langle X\rangle$ vs. $\mathbb{C}\langle X\rangle$	3.89e-243	High	0.9738	Large
$\mathbb{B}\langle X\rangle$ vs. $\mathbb{C}^{+}\langle X\rangle$	7.01e-244	High	0.9759	Large
$\mathbb{B}\langle X\rangle$ vs. $\mathbb{D}\langle X\rangle$	2.84e-247	High	0.9824	Large
$\mathbb{B}\langle X\rangle$ vs. Baseline	1.21e-256	High	n/a	n/a
$\mathbb{C}\langle X\rangle$ vs. $\mathbb{C}^{+}\langle X\rangle$	1.00e+00	Insignificant	0.0199	Negligible
$\mathbb{C}\langle X\rangle$ vs. $\mathbb{D}\langle X\rangle$	8.04e-119	High	0.7073	Large
$\mathbb{C}^{+}\langle X\rangle$ vs. $\mathbb{D}\langle X\rangle$	2.84e-154	High	0.7792	Large
$\mathbb{D}\langle X\rangle$ vs. Baseline	1.21e-256	High	n/a	n/a

C. RQ3 - All PR & Issues Opened

TABLE IV: List of PRs open - **visiting those links could reveal our identity.**

project	spec	PRs	Status
mad-lab-transit	FileMustClose	https://github.com/mad-lab/transit/pull/27	Pending
privacyidea-privacyidea	FileMustClose	https://github.com/privacyidea/privacyidea/pull/4097	Accepted
alstr-todo-to-issue-action	FileMustClose	https://github.com/alstr/todo-to-issue-action/pull/213	Accepted
bitsandbytes-foundation-bitsandbytes	RandomMustUseSeed	https://github.com/bitsandbytes-foundation/bitsandbytes/pull/1383	Accepted
slaclab-lume-model	RandomMustUseSeed	https://github.com/slaclab/lume-model/pull/95	Accepted
JeanExtreme002-FastSnake	RandomMustUseSeed	https://github.com/JeanExtreme002/FastSnake/pull/2	Pending
hetailang-SqueezeAttention	KeyInList	https://github.com/hetailang/SqueezeAttention/pull/4	Accepted
Abjad-abjad	KeyInList	https://github.com/Abjad/abjad/pull/1590	Accepted
mercedes-benz-selfsupervised_flow	TfFunction_NoSideEffect	https://github.com/mercedes-benz/selfsupervised_flow/pull/21	Accepted
fonttools-fonttools	XMLParser_ParseMustFinalize	https://github.com/fonttools/fonttools/pull/3669	Accepted
pamoller-xmlformatter	XMLParser_ParseMustFinalize	https://github.com/pamoller/xmlformatter/pull/16	Accepted
biopython-biopython	XMLParser_ParseMustFinalize	https://github.com/biopython/biopython/pull/4873	Accepted
alengwenus-pypck	Pydocs_ShouldUseStreamWriterCorrectly	https://github.com/alengwenus/pypck/pull/126	Accepted
Toilal-rebulk	KeyInList	https://github.com/Toilal/rebulk/pull/30	Pending
eandersson-amqpstorm	RandomMustUseSeed	https://github.com/eandersson/amqpstorm/pull/138	Accepted
auth0-auth0-python	RandomMustUseSeed	https://github.com/auth0/auth0-python/pull/690	Accepted
compassinformatics-wayfarer	PyDocs_MustSortBeforeGroupBy	https://github.com/compassinformatics/wayfarer/pull/67	Pending
geopython-pygeoapi	KeyInList	https://github.com/geopython/pygeoapi/pull/1951	Rejected
pandas-dev-pandas	KeyInList	https://github.com/pandas-dev/pandas/pull/61046	Accepted
Preia-Starle-casino-war-game	RandomMustUseSeed	https://github.com/Preia-Starle/casino-war-game/pull/20	Pending
nedbat-coveragepy	KeyInList	https://github.com/nedbat/coveragepy/pull/1932	Accepted
WindyLab-GenSwarm	KeyInList	https://github.com/WindyLab/GenSwarm/pull/218	Pending
davidhalter-jedi	KeyInList	https://github.com/davidhalter/jedi/pull/2051	Accepted
mila-iqia-clockwork	RandomMustUseSeed	https://github.com/mila-iqia/clockwork/pull/217	Pending
stub42-pytz	FileMustClose	https://github.com/stub42/pytz/pull/135	Pending
visgl-deck.gl	FileMustClose	https://github.com/visgl/deck.gl/pull/9500	Accepted
NOAA-ORR-ERD-gridded	Pydocs_MustShutdownBeforeCloseSocket	https://github.com/NOAA-ORR-ERD/gridded/pull/95	Accepted
paramiko-paramiko	Pydocs_MustShutdownBeforeCloseSocket	https://github.com/paramiko/paramiko/pull/2506	Pending
SeleniumHQ-selenium	Pydocs_MustShutdownBeforeCloseSocket	https://github.com/SeleniumHQ/selenium/pull/15453	Accepted
redis-redis-py	Pydocs_MustShutdownBeforeCloseSocket	https://github.com/redis/redis-py/pull/3567	Accepted
urllib3-urllib3	Pydocs_MustShutdownBeforeCloseSocket	https://github.com/urllib3/urllib3/pull/3573	Rejected
twisted-twisted	KeyInList	https://github.com/twisted/twisted/pull/12440	Pending
kizniche-Mycodo	RandomMustUseSeed	https://github.com/kizniche/Mycodo/pull/1415	Pending
Lightning-AI-pytorch-lightning	KeyInList	https://github.com/Lightning-AI/pytorch-lightning/pull/20672	Accepted
lmfit-lmfit-py	KeyInList	https://github.com/lmfit/lmfit-py/pull/997	Pending
TAMUparametric-PPOPT	RandomMustUseSeed	https://github.com/TAMUparametric/PPOPT/pull/73	Pending
PyThaiNLP-pythainlp	KeyInList	https://github.com/PyThaiNLP/pythainlp/pull/1087	Accepted
ConsumrBuzzy-RogueAsteroid	RandomMustUseSeed	https://github.com/ConsumrBuzzy/RogueAsteroid/pull/1	Accepted
lepture-mistune	KeyInList	https://github.com/lepture/mistune/pull/413	Pending
suds-community-suds	KeyInList	https://github.com/suds-community/suds/pull/112	Accepted
1200wd-bitcoinlib	KeyInList	https://github.com/1200wd/bitcoinlib/pull/456	Accepted
iop-apl-uw-basestation3	KeyInList	https://github.com/iop-apl-uw/basestation3/pull/4	Rejected
PaloAltoNetworks-xsoar-panos-upgrade-automation	FileMustClose	https://github.com/PaloAltoNetworks/xsoar-panos-upgrade-automation/pull/56	Pending
ska-sa-aiokatcp	RandomMustUseSeed	https://github.com/ska-sa/aiokatcp/pull/110	Pending
jaraco-cssutils	KeyInList	https://github.com/jaraco/cssutils/pull/63	Pending
classy-python-ccbv	KeyInList	https://github.com/classy-python/ccbv/pull/245	Pending
pyca-cryptography	RandomMustUseSeed	https://github.com/pyca/cryptography/pull/12737	Rejected
microsoft-onnxruntime	KeyInList	https://github.com/microsoft/onnxruntime/pull/24331	Pending
geyang-ml-logger	RandomMustUseSeed	https://github.com/geyang/ml-logger/pull/71	Rejected
carpedm20-emoji	KeyInList	https://github.com/carpedm20/emoji/pull/318	Accepted
opendatacube-datacube-core	KeyInList	https://github.com/opendatacube/datacube-core/pull/1774	Accepted
myhdl-myhdl	RandomMustUseSeed	https://github.com/myhdl/myhdl/pull/454	Accepted
marshmallow-code-apispec	KeyInList	https://github.com/marshmallow-code/apispec/pull/972	Accepted
fgnt-meeteval	KeyInList	https://github.com/fgnt/meeteval/pull/107	Accepted
pdyba-lambdalizator	KeyInList	https://github.com/pdyba/lambdalizator/pull/108	Accepted
OpenCTI-Platform-client-python	KeyInList	https://github.com/OpenCTI-Platform/client-python/pull/873	Pending
se-sic-VaRA-Tool-Suite	KeyInList	https://github.com/se-sic/VaRA-Tool-Suite/pull/924	Rejected
chmarti1-PYroMat	KeyInList	https://github.com/chmarti1/PYroMat/pull/105	Pending
durableOne-orgmunge	KeyInList	https://github.com/durableOne/orgmunge/pull/18	Accepted
tuplex-tuplex	RandomMustUseSeed	https://github.com/tuplex/tuplex/pull/156	Pending
jupyterhub-binderhub	RandomMustUseSeed	https://github.com/jupyterhub/binderhub/pull/1958	Pending
robinthibaut-skbel	KeyInList	https://github.com/robinthibaut/skbel/pull/7	Pending
kura-blackhole	Pydocs_ShouldUseStreamWriterCorrectly	https://github.com/kura/blackhole/pull/37	Pending
dbcli-mycli	Pydocs_NoReadAfterAccess	https://github.com/dbcli/mycli/pull/1203	Accepted
databricks-cli	FileMustClose	https://github.com/databricks/cli/pull/2740	Accepted
psf-requests	HostnamesTerminatesWithSlash	https://github.com/psf/requests/pull/6936	Accepted
snstac-pytak	Pydocs_ShouldNotInstantiateStreamWriter	https://github.com/snstac/pytak/pull/86	Pending
snstac-pytak	Pydocs_ShouldNotInstantiateStreamWriter	https://github.com/TestInABox/stackInABox/pull/93	Accepted
jdotpy-streamline	Pydocs_MustCloseSocket	https://github.com/jdotpy/streamline/pull/2	Pending
SpockBotMC-SpockBot	RandomMustUseSeed	https://github.com/SpockBotMC/SpockBot/pull/237	Pending
bitshares-python-bitshares	RandomMustUseSeed	https://github.com/bitshares/python-bitshares/pull/329	Accepted
MacHu-GWU-single_file_module-project	RandomMustUseSeed	https://github.com/MacHu-GWU/single_file_module-project/pull/1	Pending
cle-b-niobium	KeyInList	https://github.com/cle-b/niobium/pull/11	Pending
redhat-partner-tech-automated-smart-management	KeyInList	https://github.com/redhat-partner-tech/automated-smart-management/pull/8	Pending
dbischof90-sdetools	KeyInList	https://github.com/dbischof90/sdetools/pull/1	Accepted
Telefonica-HomePWN	KeyInList	https://github.com/Telefonica/HomePWN/pull/42	Accepted
Aggify-aggify	KeyInList	https://github.com/Aggify/aggify/pull/151	Pending
samdmarshall-pbPlist	KeyInList	https://github.com/samdmarshall/pbPlist/pull/11	Pending
Krebs-Busters-zone-segmentation	BuiltinAllAnalysis	https://github.com/Krebs-Busters/zone-segmentation/pull/3	Pending
achuthrajula-SER516-pairprogramming-pati	BuiltinAllAnalysis	https://github.com/achuthrajula/SER516-pairprogramming-patil/pull/21	Pending
manthanmtg-live_differ	Pydocs_NoReadAfterAccess	https://github.com/manthanmtg/live_differ/pull/3	Pending

TABLE V: List of Issue open - **visiting those links could reveal our identity.**

project	spec	PRs	Status
amoffat-sh	FileMustClose	https://github.com/amoffat/sh/issues/738	Accepted
python-cpython	XMLParser_ParseMustFinalize	https://github.com/python/cpython/issues/125397	Accepted
microsoft-playwright-python	Pydocs_ShouldUseStreamWriterCorrectly	https://github.com/microsoft/playwright-python/issues/2612	Rejected
python-cpython	Pydocs_MustShutdownBeforeCloseSocket	https://github.com/python/cpython/issues/130850	Rejected
pytorch-torchx	Pydocs_MustShutdownBeforeCloseSocket	https://github.com/pytorch/torchx/issues/1012	Pending
django-asgiref	Pydocs_MustShutdownThreadPoolExecutor	https://github.com/django/asgiref/issues/498	Accepted
python-cpython	Pydocs_MustShutdownBeforeCloseSocket	https://github.com/python/cpython/issues/130902	Rejected
tornadoweb-tornado	Pydocs_MustShutdownBeforeCloseSocket	https://github.com/tornadoweb/tornado/issues/3470	Rejected
numpy-numpy	KeyInList	https://github.com/numpy/numpy/issues/28613	Rejected
kubeflow-pipelines	RandomMustUseSeed	https://github.com/kubeflow/pipelines/issues/11812	Pending
python-cpython	KeyInList	https://github.com/python/cpython/issues/132372	Pending
psf-requests	HostnamesTerminatesWithSlash	https://github.com/psf/requests/issues/6935	Accepted

D. Description of the 73 Specs Implemented

TABLE VI: Summary of **73** specs implemented in PyMOP. (Full descriptions of the specs are available in the specs file.)

No.	Spec Name	Category	From	Short Description
1	ArgParse_Parent	Python	PyMOP	You must fully initialize the parsers before passing them via parents=.
2	Arrays_Comparable	Python	PyMOP	Checks if the elements of an array are comparable before sorting them.
3	Arrays_SortBeforeBinarySearch	Python	PyMOP	Checks if the elements of an array are sorted before binary search.
4	Console_CloseErrorWriter	Python	PyMOP	Warns if close() is invoked on sys.stderr which is a useless invocation.
5	Console_CloseReader	Python	PyMOP	Warns if close() is invoked on sys.stdin which is a useless invocation.
6	Console_CloseWriter	Python	PyMOP	Warns if close() is invoked on sys.stdout which is a useless invocation.
7	CreateWidgetOnSameFrameCanvas	Python	PyMOP	Ensures that canvas widgets are added only to the CanvasFrame's designated canvas.
8	FTP_MustLoginOnceOnly	Python	PyMOP	Must not call login more than once per instance.
9	Flask_NoModifyAfterServe	Library	PyMOP	Must not modify the flask app after it is served.
10	Flask_NoOptionsChangeAfterEnvCreate	Library	PyMOP	Warns jinja_options were changed after jinja_env is accessed.
11	Flask_UnsafeFilePath	Library	PyMOP	Must use send_from_directory() instead of flask.send_file for safety.
12	HostnamesTerminatesWithSlash	Library	PyMOP	Recommends to terminate full hostnames with a /.
13	Logging_MustNotLogAfterShutdown	Python	PyMOP	Must not log anything after the logger was shutdown.
14	NLTK_MissingMegamExplicitArg	Library	PyMOP	If explicit=True, must add -explicit argument when running call_megam.
15	NLTK_MissingMegamFvalsArg	Library	PyMOP	Must pass -fvals option when bernoulli=False.
16	NLTK_MustImplementEntries	Library	PyMOP	Flags missing required entries in probability_tables for IBM models.
17	NLTK_MutableProbDistSumToOne	Library	PyMOP	Ensures that total probabilities in update do not exceed 1.
18	NLTK_NonterminalSymbolMutability	Library	PyMOP	The symbol provided to Nonterminal must be immutable.
19	NLTK_RegexpTokenizerCapturingParentheses	Library	PyMOP	RegexpTokenizer pattern must not contain capturing parentheses.
20	NLTK_regexp_span_tokenize	Library	PyMOP	Regular expression passed to regexp_span_tokenize must not be empty.
21	PriorityQueue_NonComparable	Python	PyMOP	Checks if PriorityQueue contains non-comparable elements.
22	PyDocs_MustLockOnce	Python	PyMOP	Must not acquire the same blocking lock more than once on the same thread.
23	PyDocs_MustOnlyAddSynchronizableDataToSharedList	Python	PyMOP	Must only add synchronizable data to shared list.
24	PyDocs_MustOnlyUseDictReader	Python	PyMOP	Must always release locks after acquiring them.
25	PyDocs_MustSortBeforeGroupBy	Python	PyMOP	Must sort the list before calling groupby.
26	PyDocs_MustWaitForPopenToFinish	Python	PyMOP	Must wait for Popen to finish before exiting the program.
27	PyDocs_SharedMemoryUseAfterUnlink	Python	PyMOP	Should not access or modify shared memory after unlink().
28	PyDocs_UnsafeIterUseAfterTee	Python	PyMOP	Should not call next on iterator after tee.
29	PyDocs_UselessIterTee	Python	PyMOP	Should use all iterators created from tee.
30	Pydocs_HTTPConnectionSendSequence	Python	PyMOP	Should call send() only after endheaders() and before getresponse().
31	Pydocs_MustCloseSocket	Python	PyMOP	Must always close socket objects.
32	Pydocs_MustFlushMmap	Python	PyMOP	Must always call flush on mmap objects.
33	Pydocs_MustReleaseLock	Python	PyMOP	Must always release locks after acquiring them.
34	Pydocs_MustReleaseRLock	Python	PyMOP	Must always release RLocks after acquiring them.
35	Pydocs_MustShutdownBeforeCloseSocket	Python	PyMOP	Must call shutdown on socket before closing.
36	Pydocs_MustShutdownProcessPoolExecutor	Python	PyMOP	Must shut down ProcessPoolExecutor eventually.
37	Pydocs_MustShutdownThreadPoolExecutor	Python	PyMOP	Must shut down ThreadPoolExecutor eventually.
38	Pydocs_MustUnlinkSharedMemory	Python	PyMOP	Must call unlink() once across processes needing shared memory.
39	Pydocs_NoReadAfterAccess	Python	PyMOP	Using access() before open() creates security holes.
40	Pydocs_ShouldNotInstantiateStreamWriter	Python	PyMOP	Should not instantiate StreamWriter directly.
41	Pydocs_ShouldUseStreamWriterCorrectly	Python	PyMOP	Should call drain, close, and wait_closed() correctly.
42	Pydocs_UselessFileOpen	Python	PyMOP	Detects files opened but never used.
43	Pydocs_UselessProcessPoolExecutor	Python	PyMOP	Must use ProcessPoolExecutor if created.
44	Pydocs_UselessThreadPoolExecutor	Python	PyMOP	Must use ThreadPoolExecutor if created.
45	RandomMustUseSeed	Python	PyMOP	Must explicitly use a seed for predictable RNG.
46	RandomParams_NoPositives	Python	PyMOP	Validates mu and sigma parameters for random distributions.
47	RandomRandrange_MustNotUseKwargs	Python	PyMOP	Should not use keyword arguments in randrange.
48	Requests_DataMustOpenInBinary	Library	PyMOP	Files should be opened in binary mode with Requests.
49	Requests_MustCloseSession	Library	PyMOP	Must always close requests.Session objects.
50	Requests_PreparedRequestInit	Library	PyMOP	Checks if PreparedRequest initialized from a Request.
51	Scipy_IntegrateRange	Library	PyMOP	Ensures integration range is not too large in scipy.integrate.
52	Session_DataMustOpenInBinary	Library	PyMOP	Session data must be opened in binary mode.
53	Sets_Comparable	Python	PyMOP	Checks if set elements are comparable before sorting.
54	StringTemplate_ChangeAfterCreate	Python	PyMOP	Delimiter in Template cannot be changed after creation.
55	Thread_OverrideRun	Python	PyMOP	Checks if Thread overrides run or sets target.
56	Thread_StartOnce	Python	PyMOP	Checks if a thread is started more than once.
57	Tornado_NoAdditionalOutputCallsAfterFinish	Library	PyMOP	No output methods after finish() in Tornado.
58	Turtle_LastStatementDone	Python	PyMOP	Checks if functions are called after turtle.done().
59	UnsafeArrayIterator	Python	PyMOP	Should not call next on iterator after modifying the list.
60	UnsafeListIterator	Python	PyMOP	Should not call next on iterator after modifying the list.
61	UnsafeDictIterator	Python	PyMOP	Should not call next on iterator after modifying the dict.
62	UseProtp_in_FTP_TLS	Python	PyMOP	Must explicitly secure the data connection using prot_p().
63	VerifyPathProcessed	Library	PyMOP	Directory used for verify must be processed using c_rehash.
64	XMLParser_ParseMustFinalize	Python	PyMOP	isfinal must be true on the final call.
65	faulthandler_disableBeforeClose	Python	PyMOP	File must remain open until faulthandler.disable() is called.
66	faulthandler_tracetrackDumpBeforeClose	Python	PyMOP	File must remain open until traceback is dumped.
67	faulthandler_unregisterBeforeClose	Python	PyMOP	File must remain open until the signal is unregistered.
68	socket_create_connection	Python	PyMOP	Timeout argument must not be negative.
69	socket_setdefaulttimeout	Python	PyMOP	Timeout must not be a negative number.
70	socket_socket_settimeout	Python	PyMOP	Timeout must not be a negative number.
71	BuiltinAllAnalysis	Python	DyLin	Flags unintended truthy results from all() or any() on nested/empty lists.
72	FilesClosedAnalysis	Python	DyLin	Ensures all open files are closed.
73	ItemInList	Python	DyLin	Checks if item in list is used.