

datadog-agent

github.com/DataDog/datadog-agent

Scanned 29 October 2026 at 09:56 · Report generated 29 October 2026 · Go · 3,770 stars

OVERALL ASSESSMENT

Critical

Combined score 354 · 548 findings after filtering

Summary

104 findings likely need attention, 415 look like noise. Most notable: Detects suspicious autostart persistence mechanisms [threat-filesystem-autostart] (99% confidence).

Generated from finding counts. A plain-English summary was not available for this scan.

Risk, by type

MalDet scores two different questions separately, because they call for different responses: code can be badly written without being hostile, and hostile code is not always badly written.

VULNERABILITY

High (242)

Insecure coding an attacker could exploit. Usually mistakes, and usually fixable in place.

MALICIOUS PATTERN

Critical (112)

Signs the code may be trying to do harm: hidden payloads, credential access, persistence, exfiltration.

What each detector found

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
bandit	Insecure coding patterns in Python source	13	102	334	449
yara	Signatures of malicious behaviour in source files	44	0	36	80
guarddog	Malicious package and install-time behaviour	7	4	8	19

104 of 548 findings were rated more likely genuine than noise by the classifier.

Evidence

Identical findings are grouped, so a single rule firing many times takes one row. Ordered worst first. “Seen” is how many times that finding occurred.

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	guarddog	Detects suspicious autostart persistence mechanisms [threat-filesystem-autostart]	3	/tasks/ agent.py:615
HIGH	guarddog	Detects destructive operations (recursive deletion, wiping) [threat-filesystem-destruction]	1	/tasks/ omnibus.py:73 4
HIGH	guarddog	Detects download-and-execute patterns: fetching a remote file then executing it [threat-process-download-exec]	3	/tasks/ agent.py:384
HIGH	bandit	tarfile.extractall used without any validation. Please check and discard dangerous members.	3	/tasks/ coverage.py:3 41
HIGH	bandit	Starting a process with a shell, possible injection detected, security issue.	4	/tasks/ debugging/ delve.py:16
HIGH	bandit	subprocess call with shell=True identified, security issue.	6	/tasks/ unit_tests/ go_mod_formatter_tests.py :27
HIGH	yara	YARA rule matched: detect_credential_harvester — References credential/secret material by name AND a deliberate outbound-send API in the same file - narrower than 'reads any env var', which large legitimate codebases do constantly regardless of intent. Still not proof the two are connected (YARA c	5	/tasks/libs/ common/ omnibus.py
HIGH	yara	YARA rule matched: detect_anti_analysis — Checks for a debugger, VM, or sandbox before continuing - a common malware evasion technique, essentially never present in ordinary application code	4	/pkg/fleet/ installer/ packages/ datadog_agent_linux.go
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	17	/tasks/ collector.py

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	yara	YARA rule matched: detect_obfuscated_string_execution — Reconstructs a string character-by-character or from hex escapes, then immediately executes it - fromCharCode/chr() alone are common in ordinary text-processing code, so this requires both the reconstruction and an execution call together	4	/pkg/ template/ html/html.go
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	14	/comp/core/ gui/impl/ systray/ Sources/ gui.swift
MEDIUM	guarddog	Detects unicode homoglyphs and uncommon characters used for obfuscation [threat-runtime-obfuscation-unicode]	1	/pkg/ebpf/ uprobes/ telemetry.go:41
MEDIUM	guarddog	Detects URLs to suspicious domains often used for exfiltration or C2 [threat-network-exfiltration]	1	/pkg/util/ cloudproviders/alibaba/alibaba.go:23
MEDIUM	bandit	Use of insecure and deprecated function (mktemp).	2	/tasks/ go_deps.py:574
MEDIUM	bandit	Use of unsafe yaml load. Allows instantiation of arbitrary objects. Consider yaml.safe_load().	1	/tasks/ e2e_framework/ config.py:166
MEDIUM	bandit	Chmod setting a permissive mask 0o755 on file (dst).	4	/tasks/ security_agent.py:327
MEDIUM	bandit	Chmod setting a permissive mask 0o755 on file (bpftool_path).	1	/cmd/system-probe/ subcommands/ usm/ debug_scripts/ usm_leak_detector/ backends/ bpftool_downloader.py:143
MEDIUM	bandit	Call to requests with timeout set to None	4	/tasks/libs/ package/ url.py:92

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	bandit	Chmod setting a permissive mask 0o755 on file (bin_path).	1	/tasks/libs/build/bazel.py:287
MEDIUM	bandit	Chmod setting a permissive mask 0o755 on file (binary_path).	1	/tasks/collector.py:216
MEDIUM	bandit	Chmod setting a permissive mask 0o755 on file (call_ps1_from_bat).	1	/tasks/coverage.py:67
MEDIUM	bandit	Chmod setting a permissive mask 0o755 on file (cov_test_path).	1	/tasks/coverage.py:66
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	2	/cmd/cluster-agent/subcommands/start/command.go:511
MEDIUM	bandit	Using xml.etree.ElementTree.parse to parse untrusted XML data is known to be vulnerable to XML attacks. Replace xml.etree.ElementTree.parse with its defusedxml equivalent function or make sure defusedxml.defuse_stdlib() is called	6	/tasks/bazel.py:251
MEDIUM	bandit	Using xml.etree.ElementTree.fromstring to parse untrusted XML data is known to be vulnerable to XML attacks. Replace xml.etree.ElementTree.fromstring with its defusedxml equivalent function or make sure defusedxml.defuse_stdlib() is called	4	/tasks/libs/package/url.py:54
MEDIUM	bandit	Audit url open for permitted schemes. Allowing use of file:/ or custom schemes is often unexpected.	7	/tasks/collector.py:215
MEDIUM	bandit	Use of weak SHA1 hash for security. Consider usedforsecurity=False	1	/tasks/libs/common/omnibus.py:214
MEDIUM	bandit	Use of possibly insecure function - consider using safer ast.literal_eval.	2	/tasks/renovate.py:85

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	bandit	Use of exec detected.	2	/tasks/ new_e2e_tests .py:182
MEDIUM	bandit	Probable insecure usage of temp file/directory.	55	/tasks/ custom_task/ custom_task. py:23
MEDIUM	bandit	Call to requests without timeout	9	/tasks/ kernel_matri x_testing/ vmconfig.py:9 33
LOW	bandit	Possible hardcoded password: 'coucou'	6	/tasks/ unit_tests/ github_tasks_ tests.py:567
LOW	bandit	Possible hardcoded password: 'from-env'	3	/tasks/ unit_tests/ kmt_tests.py: 119
LOW	bandit	Possible hardcoded password: 'fake'	2	/tasks/ unit_tests/ libs/ dynamic_test/ jev_tools_te sts.py:75
LOW	bandit	Possible hardcoded password: 'deadbeef'	2	/tasks/ unit_tests/ bazel_remote_ cache_tests. py:122
LOW	bandit	Possible hardcoded password: 'explicit'	1	/tasks/ unit_tests/ e2e_framework_ tool_tests. py:111
LOW	bandit	Possible hardcoded password: 'from-config'	1	/tasks/ unit_tests/ e2e_framework_ tool_tests. py:91
LOW	bandit	Possible hardcoded password: 'slock'	1	/tasks/ unit_tests/ release_tests .py:532

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: 'sluck'	1	/tasks/ unit_tests/ release_tests .py:717
LOW	bandit	Possible hardcoded password: 'zidane'	1	/tasks/ unit_tests/ pipeline_tests.py:36
LOW	bandit	Possible hardcoded password: 'ci.datadog-agent.pulumi_password'	1	/tasks/ e2e_framework/ tool.py:187
LOW	bandit	Possible hardcoded password: 'probe'	1	/tasks/ kernel_matrix_testing/ compiler.py:49
LOW	bandit	Possible hardcoded password: '\${localEnv:GITHUB_TOKEN}'	1	/tasks/ devcontainer.py:171
LOW	bandit	Possible hardcoded password: '\${localEnv:GITLAB_TOKEN}'	1	/tasks/ devcontainer.py:172
LOW	bandit	Possible hardcoded password: 'slick'	1	/tasks/ unit_tests/ release_tests.py:599
LOW	bandit	Possible hardcoded password: 'tool --name "two words"'	1	/tasks/ unit_tests/ libs/ dynamic_test/ jev_tools_tests.py:52
LOW	bandit	Possible hardcoded password: '618'	1	/tasks/ invoke_unit_tests.py:14
LOW	bandit	Possible hardcoded password: '1234'	2	/tasks/ kernel_matrix_testing/ stacks.py:257
LOW	bandit	Possible hardcoded password: 'override'	1	/tasks/ unit_tests/ libs/ dynamic_test/ jev_tools_tests.py:61

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: 'Used to sign packages'	2	/tasks/libs/common/omnibus.py:100
LOW	bandit	Possible hardcoded password: 'fake-token'	11	/tasks/unit_tests/issue_tests.py:103
LOW	bandit	Possible hardcoded password: 'custom-token'	2	/tasks/unit_tests/libs/dynamic_test/jev_selection_tests.py:27
LOW	bandit	Possible hardcoded password: 'local-token'	1	/tasks/unit_tests/gitlab_api_tests.py:738
LOW	bandit	Possible hardcoded password: 'legacy-token'	1	/tasks/unit_tests/gitlab_api_tests.py:702
LOW	guarddog	Detects active collection of system information (hostname, platform, architecture, user) [threat-runtime-system-info]	3	/cmd/serverless-init/cloudservice/service.go:203
LOW	guarddog	Detects reading of environment variables (credential access, often contains secrets) [threat-runtime-environment-read]	3	/.dda/extend/commands/gh/cherry_pick_pr/__init__.py:55
LOW	bandit	Possible hardcoded password: 'Password for the ephemeral signing keychain'	1	/tasks/libs/common/omnibus.py:110
LOW	bandit	Possible hardcoded password: 'App-specific password for notarization'	1	/tasks/libs/common/omnibus.py:106

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: 'OIDC token used by the Bazel credential helper to authenticate against the Buildbarn remote cache'	1	/tasks/libs/common/omnibus.py:20
LOW	guarddog	Detects fully silent process execution (suppressing all output channels) [threat-process-spawn-silent]	2	/.dda/extend/pythonpath/lab/providers/local/kind.py:100

Showing the 60 most significant of 70 distinct findings. The counts in the previous section cover all of them.

How to read this report

- **This is a static scan.** The code was read, never executed. Behaviour that only appears at runtime is outside what any of these detectors can see.
- **Findings are not all equally real.** A classifier rates each finding on how likely it is to be genuine rather than noise, and that rating weights the scores. Measured against a hand-reviewed sample it reached precision 0.52 and recall 0.62 — useful for ranking, not a verdict on any single finding.
- **A high score is not proof of malice.** Security tools, penetration-testing frameworks and malware research repositories contain the same patterns as real malware by design. The two risk types above exist to keep that distinction visible.
- **Absence of findings is not proof of safety.** A detector that is not installed, or a language none of them parse, produces silence rather than a warning.
- **This scan is a snapshot.** The repository may have changed since 29 October 2026.