

terraform-provider-aws

github.com/hashicorp/terraform-provider-aws

Scanned 24 September 2026 at 06:01 · Report generated 29 October 2026 · Go · 11,095 stars

OVERALL ASSESSMENT

Medium

Combined score 38 · 64 findings after filtering

Summary

This repository has been flagged as having a medium vulnerability risk score of 28.0, which indicates that the code contains several insecure coding patterns an attacker could potentially exploit. While the overall malicious-pattern risk is low at 9.0, suggesting the code isn't intentionally harmful, there are specific issues that look more like ordinary mistakes than deliberate attacks. The scan found multiple instances of weak cryptographic hashes like SHA1 and MD5, which are outdated and should not be used for security purposes. Additionally, several lines of code appear to contain hardcoded passwords or sensitive tokens such as AWS credentials, which is a common error in production software rather than proof of malicious intent. There are also two high-severity YARA rules that detected the execution of encoded payloads and downloads followed by execution, but these were identified with very low confidence scores, meaning they likely represent false positives from test fixtures or scripts rather than active threats.

Written by a language model running locally on the scanning machine, from the findings in this report. The risk levels above and below were decided by the scoring model, not by this summary.

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

Medium (28)

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

MALICIOUS PATTERN

Low (9)

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	0	8	42	50
yara	Signatures of malicious behaviour in source files	3	0	8	11
guarddog	Malicious package and install-time behaviour	0	3	0	3

21 of 64 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	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	1	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/botocore/credentials.py
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	2	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/botocore/vendor/six.py
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	3	/internal/provider/framework/provider.go:155

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	bandit	Use of weak SHA1 hash for security. Consider usedforsecurity=False	4	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/botocore/credentials.py:847
MEDIUM	bandit	Use of weak MD5 hash for security. Consider usedforsecurity=False	1	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/botocore/compat.py:162
MEDIUM	bandit	Audit url open for permitted schemes. Allowing use of file:/ or custom schemes is often unexpected.	1	/internal/service/sns/test-fixtures/lambda_confirm_sns_extracted/main.py:24
MEDIUM	bandit	Use of exec detected.	2	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/six.py:740

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: 'aws_secret_access_key'	3	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/botocore/credentials.py:1426
LOW	bandit	Possible hardcoded password: 'AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE'	1	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/botocore/credentials.py:2074
LOW	bandit	Possible hardcoded password: 'colon'	1	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/jmespath/parser.py:234
LOW	bandit	Possible hardcoded password: '21600'	1	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/botocore/utils.py:398

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: 'number'	1	/internal/ service/ bedrockag entcore/ test- fixtures/ mcp_lambd a_extract ed/ jmespath/ parser.py: 240
LOW	bandit	Possible hardcoded password: 'rbracket'	1	/internal/ service/ bedrockag entcore/ test- fixtures/ mcp_lambd a_extract ed/ jmespath/ parser.py: 233
LOW	bandit	Possible hardcoded password: ' '	1	/internal/ service/ bedrockag entcore/ test- fixtures/ mcp_lambd a_extract ed/ dateutil/ parser/ _parser.py :122
LOW	bandit	Possible hardcoded password: 'h'	1	/internal/ service/ bedrockag entcore/ test- fixtures/ mcp_lambd a_extract ed/ botocore/ useragent. py:83

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: 'q'	1	/internal/ service/ bedrockag entcore/ test- fixtures/ mcp_lambd a_extract ed/ botocore/ useragent. py:89
LOW	bandit	Possible hardcoded password: 'AWSSecretKey'	1	/internal/ service/ bedrockag entcore/ test- fixtures/ mcp_lambd a_extract ed/ botocore/ credentia ls.py:1334
LOW	bandit	Possible hardcoded password: 'AWS_CONTAINER_AUTHORIZATION_TOKEN'	1	/internal/ service/ bedrockag entcore/ test- fixtures/ mcp_lambd a_extract ed/ botocore/ credentia ls.py:2073
LOW	bandit	Possible hardcoded password: 'AWS_WEB_IDENTITY_TOKEN_FILE'	1	/internal/ service/ bedrockag entcore/ test- fixtures/ mcp_lambd a_extract ed/ botocore/ credentia ls.py:1883

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: 'AWS_SECRET_ACCESS_KEY'	1	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/botocore/credentials.py:1189
LOW	bandit	Possible hardcoded password: 'latest/api/token'	1	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/botocore/utills.py:397
LOW	bandit	Using ElementTree to parse untrusted XML data is known to be vulnerable to XML attacks. Replace ElementTree with the equivalent defusedxml package, or make sure defusedxml.defuse_stdlib() is called.	1	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/botocore/serialize.py:49
LOW	bandit	Using xml.etree.ElementTree to parse untrusted XML data is known to be vulnerable to XML attacks. Replace xml.etree.ElementTree with the equivalent defusedxml package, or make sure defusedxml.defuse_stdlib() is called.	1	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/botocore/compat.py:87

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: 'None'	1	/internal/ service/ bedrockag entcore/ test- fixtures/ mcp_lambd a_extract ed/ botocore/ paginate. py:389
LOW	bandit	Consider possible security implications associated with the subprocess module.	2	/internal/ service/ bedrockag entcore/ test- fixtures/ mcp_lambd a_extract ed/ dateutil/ zoneinfo/ rebuild.py :6
LOW	bandit	Starting a process with a partial executable path	1	/internal/ service/ bedrockag entcore/ test- fixtures/ mcp_lambd a_extract ed/ dateutil/ zoneinfo/ rebuild.py :52

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	subprocess call - check for execution of untrusted input.	2	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/dateutil/zoneinfo/rebuild.py:62
LOW	yara	YARA rule matched: detect_downloader — Downloads a remote resource - common and low-signal alone; see detect_download_and_execute for the composite version	8	/.teamcity/scripts/pullrequest_tests/install_github_cli.sh
LOW	bandit	Use of assert detected. The enclosed code will be removed when compiling to optimised byte code.	18	/internal/service/bedrockagentcore/test-fixtures/mcp_lambda_extracted/urllib3/util/ssltransport.py:160

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 24 September 2026.