

hongbomiao.com

github.com/hongbo-miao/hongbomiao.com

Scanned 07 October 2026 at 04:38 · Report generated 29 October 2026 · Python · 299 stars

OVERALL ASSESSMENT

Critical

Combined score 1578 · 666 findings after filtering

Summary

This repository has two distinct security issues that require immediate attention before it can be considered safe for production use. The most critical problem is that the code contains 666 findings, with one specific vulnerability allowing attackers to bypass file system restrictions by manipulating paths in the source code, which could lead to unauthorized access or data theft. Additionally, there are several high-risk secrets and credentials exposed in the repository, including AWS keys, GitHub tokens, and generic API keys that have been shared publicly, meaning anyone who can view this code can steal these sensitive authentication details. While some of the detected patterns appear to be malicious attempts to hide code or execute commands, many of the issues stem from standard insecure coding practices like using untrusted data for file paths or failing to use prepared statements when running database queries against SQL databases.

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

Critical

 (1562)

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

MALICIOUS PATTERN

Low

 (15)

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
semgrep	Insecure coding patterns across many languages	111	358	97	566
bandit	Insecure coding patterns in Python source	0	63	11	74
yara	Signatures of malicious behaviour in source files	7	0	14	21
guarddog	Malicious package and install-time behaviour	1	1	3	5

324 of 666 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	semgrep	The application builds a file path from potentially untrusted data, which can lead to a path traversal vulnerability. An attacker can manipulate the path which the application uses to access files. If the application does not validate user input and sanitize file paths, sensitive files such as confi	6	/autonomy/camera-radar-lidar-fusion/src/shared/radar/services/load_radar_data.rs:11
HIGH	semgrep	Found an insecure gRPC connection using 'insecure.NewCredentials()'. This creates a connection without encryption to a gRPC server. A malicious attacker could tamper with the gRPC message, which could compromise the machine. Instead, establish a secure connection with an SSL certificate using the 'g	6	/api/api-go/internal/graphql_server/utis/verify_password.go:20
HIGH	semgrep	AWS Secret Access Key detected	12	/kubernetes/argo-cd/projects/nebius/production-hm/tempo/environments/production/kubernetes-manifests/hm-tempo-secret.yaml:10

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	GitHub Token detected	3	/kubernetes/argo-cd/projects/nebius/production-hm/gha-runner-scale-set-large/environments/production/secrets/hm-gha-runner-scale-set-large-secret.unsealed.yaml.example:11
HIGH	semgrep	The last user in the container is 'root'. This is a security hazard because if an attacker gains control of the container they will have root access. Switch back to another user after running commands as 'root'.	1	/data-visualization/grafana/hm-panel-plugin/.config/Dockerfile:26
HIGH	guarddog	Detects steganography decode followed by code execution [threat-runtime-obfuscation-steganography]	1	/computer-vision/anyup/src/main.py:51
HIGH	semgrep	This workflow uses `secrets: inherit` to pass all of the calling workflow's secrets to a reusable workflow. This violates the principle of least privilege because the called workflow receives access to every secret in the repository, not just the ones it needs. If the called workflow is compromised	20	/.github/workflows/test.yml:1156
HIGH	semgrep	sqlalchemy.text passes the constructed SQL statement to the database mostly unchanged. This means that the usual SQL injection protections are not applied and this function is vulnerable to SQL injection if user input can reach here. Use normal SQLAlchemy operators (such as `or_()`, `and_()`, etc.)	1	/data-storage/delta-lake/read-delta-lake-by-trino/src/main.py:20

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	Avoiding SQL string concatenation: untrusted input concatenated with raw SQL query can result in SQL Injection. In order to execute raw query safely, prepared statement should be used. SQLAlchemy provides TextualSQL to easily used prepared statement with named parameters. For complex SQL composition	4	/data-storage/hm-duckdb/query-amazon-s3/src/main.py:29
HIGH	semgrep	Generic API Key detected	1	/kubernetes/argo-cd/projects/aws/production-hm/litellm/environments/production/kubernetes-manifests/hm-litellm-secret.yaml:9
HIGH	semgrep	Functions reliant on pickle can result in arbitrary code execution. Consider loading from `state_dict`, using fickling, or switching to a safer serialization method like ONNX	2	/graph-neural-network/src/main.py:215
HIGH	semgrep	Generic Secret detected	13	/api/api-node/.env.development.local.example:1
HIGH	semgrep	Insecure WebSocket Detected. WebSocket Secure (wss) should be used for all WebSocket connections.	11	/video/media-streaming-server/livekit/multimedia-publisher/src/shared/egress/services/start_track_composite_hls_egress.rs:26
HIGH	semgrep	JWT token detected	1	/security/defguard/docker-compose.development.yaml:34
HIGH	semgrep	bcrypt hash detected	2	/kubernetes/argo-cd/justfile:7

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	By not specifying a USER, a program in the container may run as 'root'. This is a security hazard. If an attacker can control a process running as root, they may have control over the container. Ensure that the last USER in a Dockerfile is a USER other than 'root'.	28	/data-visualization/hm-dash/parquet-visualizer/Dockerfile:15
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	4	/machine-learning/hm-mlflow/experiments/classify-mnist/skypilot.yaml
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	3	/operating-system/ubuntu/justfile
MEDIUM	semgrep	This pyproject.toml configures uv but does not set a dependency cooldown. Newly published packages can be malicious or unstable. Add `exclude-newer = "7 days"` under `[tool.uv]` to wait 7 days before resolving newly published package versions. Added in: 0.9.17 Reference: https://docs.astral.sh/uv/co	211	/large-language-model-post-training/fine-tuning/proximal-policy-optimization/low-rank-adaptation/pyproject.toml:19
MEDIUM	semgrep	If possible, it is better to rely on automatic pinning in PyTorch to avoid undefined behavior and for efficiency	11	/generative-model/variational-autoencoder/src/main.py:36
MEDIUM	semgrep	This .npmrc does not set a minimum release age or sets it too low. Newly published packages can be malicious or unstable. Add `min-release-age = 7` to wait 7 days before resolving newly published package versions. Added in: v11.10 Reference: https://github.blog/changelog/2026-02-18-npm-bulk-trusted-	9	/data-processing/pulsar/audio-stream/audio-stream-web/.npmrc:1
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	1	/aerospace/flightradar24/src/config.py:12

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	The application exports an activity. Any application on the device can launch the exported activity which may compromise the integrity of your application or its data. Ensure that any exported activities do not have privileged access to your application's control plane.	1	/mobile/mobile-android/app/src/main/AndroidManifest.xml:13
MEDIUM	semgrep	Data is being piped into `bash` from a `curl` command. An attacker with control of the server in the `curl` command could inject malicious code into the pipe, resulting in a system compromise. Avoid piping untrusted data into `bash` or any other shell if you can. If you must do this, consider checki	2	/kubernetes/scripts/utils/install_pixie_cloud.sh:7
MEDIUM	semgrep	Container or pod is running in privileged mode. This grants the container the equivalent of root capabilities on the host machine. This can lead to container escapes, privilege escalation, and other security concerns. Remove the 'privileged' key to disable this capability.	2	/kubernetes/argo-cd/projects/nebius/production-hm/gha-runner-scale-set-large/environments/production/helm-chart/values.yaml:72
MEDIUM	semgrep	Detected possible formatted SQL query. Use parameterized queries instead.	4	/data-storage/hm-duckdb/query-amazon-s3/src/main.py:29
MEDIUM	semgrep	Detected a dynamic value being used with urllib. urllib supports 'file://' schemes, so a dynamic value controlled by a malicious actor may allow them to read arbitrary files. Audit uses of urllib calls to ensure user data cannot control the URLs, or consider using the 'requests' library instead.	1	/machine-learning/nvidia-dali/src/main.py:31
MEDIUM	semgrep	CORS policy allows any origin (using wildcard '*'). This is insecure and should be avoided.	1	/api/api-python/src/app.py:45

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	bandit	Use of unsafe PyTorch load	4	/multimodal-model/vision-language-action-model/vision-language-action/infer/src/shared/agent/utils/create_vla_agent.py:67
MEDIUM	bandit	Audit url open for permitted schemes. Allowing use of file:/ or custom schemes is often unexpected.	1	/machine-learning/nvidia-dali/src/main.py:31
MEDIUM	semgrep	Found an HTTP server without TLS. Use 'http.ListenAndServeTLS' instead. See https://golang.org/pkg/net/http/#ListenAndServeTLS for more information.	1	/api/api-go/cmd/graphql_server/main.go:64
MEDIUM	bandit	Possible SQL injection vector through string-based query construction.	7	/data-visualization/hmdash/parquet-visualizer/src/main.py:60
MEDIUM	bandit	Possible binding to all interfaces.	1	/cloud-platform/cilium/multi-cloud-data-pipeline/sensor-gateway/src/main.py:24
MEDIUM	semgrep	This tag is missing an 'integrity' subresource integrity attribute. The 'integrity' attribute allows for the browser to verify that externally hosted files (for example from a CDN) are delivered without unexpected manipulation. Without this attribute, if an attacker can modify the externally hosted	1	/wireless/software-defined-radio/radio-streamer/web/index.html:7

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	2	/data-storage/doris/storage-compute-decoupled/kubernetes/foundationdb/crds/foundationdbbackups.yaml:1156
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	2	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:323
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	2	/cloud-platform/cilium/multi-cloud-data-pipeline/kubernetes/foundationdb/crds/foundationdbbackups.yaml:1156
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/nebius/production-hm/gha-runner-scale-set-medium/environments/production/helm-chart/values.yaml:71
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/nebius/production-hm/gha-runner-scale-set-large/environments/production/helm-chart/values.yaml:71

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:123
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:476
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:173
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:223
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:273

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:374
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:425
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	27	/kubernetes/manifests/hydra/hydra-statefulset.yaml:21
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	11	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:438
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	10	/api/real-time-ai-agent/kubernetes/agent/agent.yaml:31

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	3	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:489
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	3	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:85
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	2	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:236
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:136
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:286

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:336
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes/argo-cd/projects/aws/production-hm/ray-cluster/environments/production/kubernetes-manifests/hm-ray-cluster.yaml:387
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	29	/data-storage/doris/storage-compute-decoupled/kubernetes/paimon-seed/flink-session-cluster.yaml:37
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	8	/api/real-time-ai-agent/kubernetes/sensor-stream-publisher/sensor-stream-publisher.yaml:18
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	2	/kubernetes/manifests/east/grpc-server-deployment.yaml:20

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	bandit	Probable insecure usage of temp file/directory.	1	/data-orchestration/hm-airflow/dags/s3_download.py:31

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 07 October 2026.