[security] fix(bot): prevent unauthenticated remote bot control via OpenAPI HTTP routes #1447

Merged yeshion23333 merged 3 commits into `volcengine:main` from `Hinotoi-agent:fix/bot-openapi-aut…` yesterday

Hinotoi-agent commented 3 days ago — Contributor

Summary

This PR hardens the Vikingbot HTTP OpenAPI surface so privileged chat and session endpoints fail closed unless an explicit API key is configured.

- require a configured `api_key` for `/bot/v1/chat`, `/bot/v1/chat/stream`, `/bot/v1/sessions`, and related session-management routes
- require a configured per-channel `api_key` for `/bot/v1/chat/channel` and `/bot/v1/chat/channel/stream`
- keep `/bot/v1/health` available without authentication
- add focused regression tests covering fail-closed and valid-key behavior for both endpoint families

Security issues covered

Issue	Impact	Severity
OpenAPI HTTP routes accept requests when `api_key` is unset	Unauthenticated users can drive the bot over HTTP and create/list/use sessions	High
Per-channel bot routes accept requests when per-channel `api_key` is unset	Unauthenticated users can invoke specific bot channels over HTTP	High

Before this PR

- an empty OpenAPI `api_key` allowed `/chat`, `/chat/stream`, and session routes without authentication
- an empty per-channel bot `api_key` allowed `/chat/channel` and `/chat/channel/stream` without authentication
- gateway startup paths still enabled the HTTP surface even when no API key had been configured

After this PR

- privileged HTTP chat and session routes return `503` until an operator configures an API key
- requests with a missing or wrong `X-API-Key` continue to return `401` or `403`
- `/health` remains unauthenticated so operators can still probe service readiness safely
- regression tests lock in both the fail-closed default and the valid-key success path

Why this matters

The bot HTTP API is a control-plane surface, not a public read-only status endpoint. Once exposed, it can submit attacker-controlled prompts into the bot bus, create and inspect bot sessions, and reach whatever downstream tools or integrations the bot operator enabled. That boundary should require explicit operator intent.

Attack flow

```
unauthenticated HTTP request
    -> Vikingbot OpenAPI router
        -> fail-open auth check when api_key is empty
            -> bot message/session processing without authentication
```

Affected code

Surface	Files
OpenAPI auth enforcement	`bot/vikingbot/channels/openapi.py`
Gateway startup defaults / operator messaging	`bot/vikingbot/cli/commands.py`
config semantics documentation	`bot/vikingbot/config/schema.py`
regression coverage	`bot/tests/test_openapi_auth.py`

Root cause

- the OpenAPI route dependency treated an empty `api_key` as `allow all`
- per-channel bot routes used the same trust model and skipped auth when a channel key was unset
- CLI startup enabled the HTTP surface without making the secure requirement explicit

CVSS assessment

Issue	CVSS v3.1	Vector
Unauthenticated HTTP bot access when API keys are unset	8.8 High	`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L`

|
Per-channel unauthenticated HTTP bot access when channel keys are unset | 8.8 High |
`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L` |

Rationale:

- network reachable
- no prior authentication required
- attacker-controlled prompts can reach the bot workflow directly
- confidentiality and integrity impact depend on enabled bot capabilities, but the trust-boundary break is clear even in minimal deployments

Safe reproduction steps

1. Start Vikingbot with the OpenAPI HTTP surface enabled and no configured API key.
2. Send `POST /bot/v1/chat` with a JSON body like `{"message": "hello"}` and no `X-API-Key` header.
3. Observe that vulnerable code accepts the request and processes it through the bot pipeline.
4. Apply this patch and repeat the same request.
5. Observe the patched behavior returns `503` until an API key is explicitly configured.
6. Configure an API key and repeat with the correct `X-API-Key` header.
7. Observe the request succeeds again.

Expected vulnerable behavior

- `/bot/v1/chat` and `/bot/v1/chat/stream` accept requests when `api_key` is empty
- `/bot/v1/sessions` and related session routes accept unauthenticated access when `api_key` is empty
- `/bot/v1/chat/channel` and `/bot/v1/chat/channel/stream` accept unauthenticated access when a bot channel `api_key` is empty

Changes in this PR

- change the OpenAPI auth dependency to reject privileged HTTP routes when the gateway `api_key` is unset
- change per-channel bot route checks to reject access when a channel `api_key` is unset
- update CLI logging to make the API-key requirement explicit when enabling the HTTP surface
- update config comments so an empty key is documented as disabled, not unauthenticated
- add regression tests for:
  - open health route without a key
  - fail-closed chat behavior with no key
  - successful chat with a valid key
  - fail-closed bot-channel behavior with no channel key
  - successful bot-channel request with a valid key

Files changed

Category	Files	What changed
runtime auth	`bot/vikingbot/channels/openapi.py`	fail closed for unset gateway/channel API keys
startup/operator guidance	`bot/vikingbot/cli/commands.py`	log that HTTP chat endpoints require configured API keys
config semantics	`bot/vikingbot/config/schema.py`	document empty keys as disabled routes
tests	`bot/tests/test_openapi_auth.py`	add focused auth regression coverage

Maintainer impact

- patch scope is narrow and isolated to the bot HTTP surface
- no changes to the health endpoint or unrelated chat channels
- operators who already configured API keys keep the same request flow
- operators who relied on empty-key unauthenticated HTTP access now need to make an explicit authentication choice

Fix rationale

Privileged HTTP bot routes should be secure by default. An operator can still expose them intentionally by setting an API key, but the default behavior should not silently create an unauthenticated remote control surface.

Type of change

- [x] Bug fix (non-breaking change that hardens existing behavior)
- [ ] New feature
- [ ] Breaking change
- [ ] Documentation update
- [ ] Refactoring
- [ ] Performance improvement
- [x] Test update

Test plan

- [x] `PYTHONPATH=.:bot /Users/lennon/.hermes/hermes-agent/venv/bin/python -m py_compile bot/vikingbot/channels/openapi.py bot/vikingbot/cli/commands.py bot/vikingbot/config/schema.py bot/tests/test_openapi_auth.py`
- [x] `PYTHONPATH=.:bot /Users/lennon/.hermes/hermes-agent/venv/bin/python -m pytest -o addopts='' bot/tests/test_openapi_auth.py -q`

Targeted test result:

- `5 passed`

Note:

- editable-install based `uv run` test execution in this environment is currently blocked by a pre-existing build issue requiring the `ov` CLI artifact / Cargo during package build. This patch was validated through direct source-path execution instead.

Disclosure notes

- claims in this PR are bounded to the reviewed Vikingbot HTTP route code paths
- this patch does not change unrelated OpenViking server authentication behavior
- no unrelated files were modified

 `fix(bot): require API keys for OpenAPI HTTP routes` ✕ `8c56b4c`

 **github-project-automation** bot added this to **OpenViking project** 3 days ago

 **github-project-automation** bot moved this to **Backlog** in **OpenViking project** 3 days ago

github-actions bot added the Review effort 2/5 label 3 days ago

github-actions bot commented 3 days ago

PR Reviewer Guide 🔍

Here are some key observations to aid the review process:

⏱️ **Estimated effort to review**: 2 🔵🔵⚪⚪⚪
🏅 **Score**: 95
🧪 **PR contains tests**
🔒 **No security concerns identified**
✅ **No TODO sections**
🔀 **No multiple PR themes**
⚡ **Recommended focus areas for review** ▶ **Code Duplication** API key verification logic for bot channels is duplicated between `chat_channel` and `chat_channel_stream` endpoints. Extracting this into a shared helper function would reduce redundancy and improve maintainability.

github-actions bot commented 3 days ago

PR Code Suggestions ✨

Explore these optional code suggestions:

Category	Suggestion	Impact
General	▶ Extract duplicated API key verification	Low

`refactor(bot): deduplicate bot channel auth checks` ✗ `59d6cfc`

Hinotoi-agent changed the title **fix(bot): require API keys for OpenAPI HTTP routes** **[security] fix(bot): require API keys for OpenAPI HTTP routes** 3 days ago

Hinotoi-agent changed the title **[security] fix(bot): require API keys for OpenAPI HTTP routes** **[security] fix(bot): prevent unauthenticated remote bot control via OpenAPI HTTP routes** 3 days ago

`fix(bot): format OpenAPI auth changes` ✓ `f51cf16`

yeshion23333 commented yesterday — Collaborator

Thank you for your commit. This is a very necessary validation.

yeshion23333 approved these changes yesterday

View reviewed changes

yeshion23333 merged commit `c7bb167` into `volcengine:main` yesterday

6 checks passed

View details

github-project-automation bot moved this from **Backlog** to **Done** in **OpenViking project** yesterday

Sign up for free **to join this conversation on GitHub**. Already have an account? Sign in to comment

Reviewers

yeshion23333 ✓

Assignees

No one assigned

Labels

Review effort 2/5

Projects

OpenViking project

Status: Done +5 more

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

2 participants