fix: harden gateway slash command security #127

Merged tjb-tech merged 2 commits into `HKUDS:main` from `Hinotoi-agent:fix/gateway-command…` 3 days ago

Hinotoi-agent commented 3 days ago • edited

Contributor

Summary

This PR hardens the gateway/slash-command path against two verified security issues:

- remote gateway users could invoke local-only administrative commands from chat
- `/memory show` could read files outside the project memory directory via path traversal
- the gateway now supports a secure opt-in path for trusted remote admin commands using an explicit config gate plus a command allowlist
- regression tests now cover the default-deny path and the trusted opt-in behavior

Security issues covered

Issue	Impact	Severity
Remote slash-command permission-mode escalation	Remote users can change a local safety boundary from chat	High
Arbitrary file read via `/memory show`	Remote users can read host files outside project memory	High

Before this PR

- inbound remote messages could execute sensitive slash commands without distinguishing local-only administrative actions from remote-safe commands

- `/memory show` accepted attacker-controlled path input and could resolve reads outside the project memory directory
- there was no secure built-in way to preserve trusted remote admin workflows without leaving sensitive commands remotely reachable by default
- these trust boundaries were not explicitly covered by regression tests in the gateway/slash-command path

After this PR

- sensitive administrative slash commands remain denied by default when received through the remote gateway path
- selected administrative slash commands can now be re-enabled only through an explicit gateway config gate and per-command allowlist
- accepted remote administrative commands emit warning-level audit logs and gateway startup warns when the opt-in is enabled
- `/memory show` resolves targets safely and enforces containment under the project memory directory before reading
- regression tests verify default-deny, trusted opt-in, and traversal rejection behavior

Why this matters

These issues sit on trust boundaries that are reachable from remote chat/gateway usage:

1. permission mode is a safety boundary and should not be mutable by remote chat users unless an operator explicitly opts into that behavior
2. memory entry reads should stay inside the project memory directory and never escape to arbitrary host files

In practice, one issue weakens a meaningful safety control and the other exposes arbitrary file-read behavior from the host running OpenHarness.

Attack flow

```
Remote user message
    -> gateway slash-command handling
        -> sensitive command accepted from remote chat
            -> permission mode changed

Remote user message
    -> /memory show <attacker-controlled path>
        -> path joined to memory directory without containment check
            -> host file outside project memory returned to remote user
```

Affected code

Issue	Files
Remote slash-command permission-mode escalation	`ohmo/gateway/runtime.py`, `ohmo/gateway/models.py`, `ohmo/gateway/service.py`, `ohmo/cli.py`, `src/openharness/commands/registry.py`
Arbitrary file read via `/memory show`	`src/openharness/commands/registry.py`, `src/openharness/memory/paths.py`

Root cause

Issue 1: remote slash-command permission-mode escalation

- the gateway accepted slash commands directly from inbound remote messages
- sensitive administrative commands were not distinguished from normal remote-safe commands before execution
- there was no explicit secure-default override model for operators who wanted trusted remote admin behavior

Issue 2: arbitrary file read via `/memory show`

- `/memory show` joined attacker-controlled input onto the memory directory path
- the resulting path was read without enforcing containment under the project memory directory

CVSS assessment

Issue	CVSS v3.1	Vector
Remote slash-command permission-mode escalation	8.8 High	`AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H`
Arbitrary file read via `/memory show`	8.6 High	`AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N`

Rationale:

- Issue 1 allows a remote gateway user with chat access to disable a meaningful protection boundary and unlock dangerous follow-on operations when the deployment has not intentionally opted into remote admin behavior.
- Issue 2 allows a remote gateway user with chat access to read arbitrary host files reachable by the OpenHarness process.

Safe reproduction steps

1. Remote slash-command permission-mode escalation

1. Run the gateway and connect a remote channel.
2. From the remote chat, send:
   - `/permissions full_auto`
3. Observe that permission mode changes from chat, even though this is a local safety control.

2. Arbitrary file read via `/memory show`

1. Run the gateway or slash-command surface against a project.
2. From chat or command input, request:
   - `/memory show ../../../../../../etc/hosts`
3. Observe that the command returns file content outside the memory directory instead of rejecting traversal.

Expected vulnerable behavior

- remote users should not be able to switch permission mode from chat by default
- `/memory show` should not read files outside the project memory directory

Changes in this PR

- add `remote_admin_opt_in` to `SlashCommand`
- keep `/permissions` and `/plan` remote-denied by default while marking them eligible for explicit remote opt-in
- add `allow_remote_admin_commands` and `allowed_remote_admin_commands` to gateway config
- allow only explicitly listed remote admin commands when the gateway opt-in is enabled
- emit warning-level audit logs when a remote administrative command is accepted
- emit a gateway startup warning when remote admin opt-in is enabled
- expose the remote admin opt-in and allowlist in the gateway config wizard and config summary
- resolve `/memory show` targets safely and enforce containment under the project memory directory
- add regression tests for default-deny, trusted opt-in, and traversal rejection

Files changed

Category	Files	What changed
Gateway enforcement	`ohmo/gateway/runtime.py`	Keeps remote admin denied by default, permits only explicitly

Category	Files	What changed
		opted-in commands, and logs accepted remote admin events
Gateway config model	`ohmo/gateway/models.py`, `ohmo/workspace.py`	Adds persisted remote admin opt-in fields and secure defaults for new workspaces
Gateway startup behavior	`ohmo/gateway/service.py`	Warns at startup when remote admin opt-in is enabled
Gateway configuration UX	`ohmo/cli.py`	Adds config wizard prompts and summary output for remote admin opt-in and allowlisted commands
Command metadata + memory read hardening	`src/openharness/commands/registry.py`	Adds `remote_admin_opt_in`, marks sensitive commands as explicitly opt-in only, and hardens `/memory show`
Regression coverage	`tests/test_commands/test_registry.py`, `tests/test_ohmo/test_gateway.py`	Adds tests for explicit remote admin opt-in and preserves traversal/blocking coverage

Maintainer impact

- no frontend behavior is changed
- the patch remains narrowly scoped to gateway command enforcement, gateway configuration, memory-path validation, and tests
- deployments that do nothing keep the secure default: remote admin commands stay blocked
- self-hosted trusted deployments now have a built-in, explicit, auditable opt-in path instead of needing to weaken the default boundary
- regression coverage reduces the risk of reintroducing these trust-boundary failures during later refactors

Suggested fix rationale

- keep administrative safety toggles local-only by default
- allow trusted operators to opt into specific remote admin commands explicitly instead of reopening the whole slash-command surface
- validate filesystem containment before any read
- add regression tests so these issues do not reappear through future command or gateway refactors

Reference patterns from other software

- GitHub Actions requires explicit approval for workflow runs from public forks before untrusted PR code gets access to privileged CI execution: https://docs.github.com/en/actions/managing-workflow-runs/approving-workflow-runs-from-public-forks
- GitLab restricts access to protected variables and runners for merge request pipelines unless explicitly allowed, keeping privileged automation behind an opt-in trust boundary: https://docs.gitlab.com/ci/pipelines/merge_request_pipelines/#control-access-to-protected-variables-and-runners
- Kubernetes RBAC separates privileged operations behind explicit role bindings rather than exposing administrative actions to every authenticated caller by default: https://kubernetes.io/docs/reference/access-authn-authz/rbac/
- Keycloak documents fine-grained administrative permissions so sensitive management actions can be delegated narrowly instead of broadly enabling admin behavior: https://www.keycloak.org/docs/latest/server_admin/#_fine_grain_permissions

These are not identical products, but they reflect the same secure-default pattern: privileged actions stay off by default for untrusted remote inputs, and trusted exceptions require explicit configuration.

Type of change

- [x] Security fix
- [x] Tests
- [ ] Documentation update
- [ ] Refactor with no behavior change

Test plan

- [x] `tests/test_commands/test_registry.py::test_permissions_command_persists`
- [x] `tests/test_commands/test_registry.py::test_permissions_command_is_marked_local_only`
- [x] `tests/test_commands/test_registry.py::test_permissions_command_supports_explicit_remote_admin_opt_in`
- [x] `tests/test_commands/test_registry.py::test_memory_show_rejects_path_traversal`
- [x] `tests/test_commands/test_registry.py::test_memory_show_reads_normal_entries_with_md_fallback`
- [x] `tests/test_ohmo/test_gateway.py::test_runtime_pool_stream_message_emits_progress_and_tool_hint`
- [x] `tests/test_ohmo/test_gateway.py::test_runtime_pool_blocks_local_only_commands_from_remote_`

`messages`

- [x] `tests/test_ohmo/test_gateway.py::test_runtime_pool_allows_opted_in_remote_admin_commands`
- [x] `uv run ruff check src tests ohmo`

Executed with:

- `PYTHONPATH=src:. .venv/bin/python -m pytest tests/test_commands/test_registry.py::test_permissions_command_persists tests/test_commands/test_registry.py::test_permissions_command_is_marked_local_only tests/test_commands/test_registry.py::test_permissions_command_supports_explicit_remote_admin_opt_in tests/test_commands/test_registry.py::test_memory_show_rejects_path_traversal tests/test_commands/test_registry.py::test_memory_show_reads_normal_entries_with_md_fallback tests/test_ohmo/test_gateway.py::test_runtime_pool_stream_message_emits_progress_and_tool_hint tests/test_ohmo/test_gateway.py::test_runtime_pool_blocks_local_only_commands_from_remote_messages tests/test_ohmo/test_gateway.py::test_runtime_pool_allows_opted_in_remote_admin_commands -q`
- `uv run ruff check src tests ohmo`

Disclosure notes

- claims are intentionally bounded to what is demonstrated by code review and local reproduction
- this PR fixes the behavior directly and adds regression coverage
- no unrelated project files are changed

Merge upstream main and add Docker sandbox example pierg/OpenHarness#13

Merged

Reviewers

No reviews

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

2 participants