[security] fix(ohmo): isolate shared-chat sessions by sender #159

Merged tjb-tech merged 1 commit into `HKUDS:main` from `Hinotoi-agent:fix/ohmo-session-se…`

4 days ago

Hinotoi-agent commented 4 days ago — Contributor

Summary

This PR hardens ohmo gateway session isolation by scoping chat sessions to the remote sender, not just the shared chat or thread.

- fixes cross-user session reuse in shared chats and threaded conversations
- prevents one participant from inheriting another participant's restored ohmo session state
- prevents one participant from canceling or replacing another participant's in-flight run solely by sending a newer message in the same shared context
- adds focused regression coverage for sender-scoped session routing and sender-scoped snapshot restore behavior

Security issues covered

Issue	Impact	Severity
Shared-chat/session-key collision in ohmo gateway	Another allowed participant in the same shared chat or thread can inherit prior assistant state and interrupt an in-flight run started by someone else	Medium

Before this PR

- ohmo gateway keyed sessions by `channel + chat_id`, with optional thread metadata when present
- sender identity was not part of the session boundary

- runtime restore loaded the latest saved snapshot for that shared session key
- bridge interruption logic also used that same shared key, so a new message from one participant could replace another participant's active task
- tests covered thread-aware routing, but did not assert sender isolation inside the same shared chat or thread

After this PR

- ohmo gateway keys sessions by `channel + chat_id + sender_id`, with thread metadata included when present
- saved snapshots are only restored for the same sender-scoped session key
- one participant's new message no longer targets another participant's in-flight run in the same shared chat/thread
- regression tests now lock in sender separation for router behavior, restart-path session keys, and runtime snapshot restore behavior

Why this matters

The broken trust boundary was not remote admission, but principal separation after admission.

Once a shared chat or thread was allowed to talk to the gateway, ohmo treated every participant in that shared context as the same session owner. That meant one user could inherit another user's conversation state and cancel another user's running task without needing slash-command access, plugin control, or an allowlist bypass.

How this differs from related issue/PR

This PR is related to the already-public ohmo/OpenHarness remote trust-boundary work, but it fixes a different layer.

1. Different boundary

- PR **[security] fix(ohmo): secure default remote channel allowlists** #147 hardened who may enter the remote channel at all through secure-default allowlists
- PR **fix: harden gateway slash command security** #127 hardened which slash-command/control-plane actions are reachable remotely after admission
- this PR hardens session ownership after admission by separating participants inside the same shared chat/thread

2. Different trigger condition

- this issue does not require dangerous slash commands, plugin lifecycle control, or any bypass of channel admission
- an ordinary message from another participant in the same allowed shared context is enough

3. Different vulnerable code

- this patch changes ohmo session routing and sender-scoped restore expectations
- it does not change channel allowlist defaults, slash-command privilege flags, or plugin trust policy

4. Different failure mode

- the failure here is cross-user session reuse and interruption inside an already-allowed shared context
- the prior public fixes addressed earlier admission and command-boundary problems, not per-sender session ownership

5. Why this is a variant instead of a duplicate

- all of these issues belong to the broader remote trust-boundary family
- but this PR fixes a distinct ownership boundary that remains relevant even when channel admission and remote command exposure are already hardened

Attack flow

```
allowed remote participant in a shared chat/thread
    -> ohmo session routing omits sender identity
        -> runtime restore and bridge interruption operate on a shared session key
            -> cross-user context reuse and task interruption
```

Affected code

Issue	Files
Shared-chat/session-key collision in ohmo gateway	`ohmo/gateway/router.py`, `tests/test_ohmo/test_gateway.py`

Root cause

Issue 1: shared-chat/session-key collision in ohmo gateway

- session routing derived ownership from chat/thread identifiers only and ignored `sender_id`
- runtime restore and bridge interruption both trusted that shared key as if it represented a single principal

CVSS assessment

Issue	CVSS v3.1	Vector
Shared-chat/session-key collision in ohmo gateway	6.5 Medium	`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:L`

Rationale:

- the attacker must already be an allowed participant in the shared remote context, but no additional privilege or special interaction is required
- confidentiality impact is the primary concern because previously saved context can be inherited across users in the same shared chat/thread
- integrity and availability are also affected because a newer message can replace another participant's active run

Safe reproduction steps

1. Shared-chat/session-key collision in ohmo gateway

1. Use a vulnerable OpenHarness snapshot before this patch.
2. Enable ohmo on a shared remote channel/chat that admits multiple participants.
3. Have user A send a message that creates a session and leaves conversation state or a persisted snapshot.
4. From the same shared chat/thread, have user B send a normal follow-up message.
5. Observe that the gateway computes the same session key for both users because it keys only on chat/thread scope.
6. Observe that user B reuses/restores user A's session state.
7. If user A has an in-flight task, send a new message from user B and observe that the prior task is replaced by the newer message.

Expected vulnerable behavior

- users sharing the same allowed chat/thread should never inherit each other's ohmo session state solely because they share a room
- users sharing the same allowed chat/thread should never be able to cancel each other's active runs with ordinary follow-up messages
- pre-patch behavior allowed both because sender identity was not part of the session boundary

Changes in this PR

- include `sender_id` in computed ohmo session keys when no explicit session override is provided

- preserve thread-aware routing while separating participants inside the same shared thread
- add regression coverage proving two senders in the same chat/thread receive different session keys
- add regression coverage proving runtime restore does not load another sender's snapshot
- update the existing gateway restart-path test to match sender-scoped session keys

Files changed

Category	Files	What changed
Session routing	`ohmo/gateway/router.py`	add sender identity to the default session-key boundary while preserving thread-aware routing
Regression tests	`tests/test_ohmo/test_gateway.py`	add sender-isolation routing/restore tests and update session-key expectations

Maintainer impact

- the patch is intentionally narrow and limited to ohmo sender-scoped session routing plus tests
- channel admission logic, plugin trust logic, and remote slash-command policy are unchanged
- this makes the session ownership model easier to reason about in future channel or bridge changes because the principal boundary now matches the actual sender

Fix rationale

The correct boundary is not “everyone in the same chat is one session.”
The correct boundary is “each sender owns their own session unless the channel explicitly overrides that behavior.”

This fix is narrow, durable, and aligned with the existing `sender_id` data already carried through inbound channel messages. The tests are sufficient because they lock the exact ownership boundary that previously failed: routing, restore, and gateway command paths that depend on the computed session key.

Type of change

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

Test plan

- [x] `PYTHONPATH=src .venv/bin/python -m pytest tests/test_ohmo/test_gateway.py tests/test_ohmo/test_ohmo_session_storage.py -q`
- [x] `PYTHONPATH=src .venv/bin/python -m ruff check ohmo/gateway/router.py tests/test_ohmo/test_gateway.py tests/test_ohmo/test_ohmo_session_storage.py`
- [ ] Full project test suite was not run for this targeted fix

Executed with:

- `PYTHONPATH=src .venv/bin/python -m pytest tests/test_ohmo/test_gateway.py tests/test_ohmo/test_ohmo_session_storage.py -q`
- `PYTHONPATH=src .venv/bin/python -m ruff check ohmo/gateway/router.py tests/test_ohmo/test_gateway.py tests/test_ohmo/test_ohmo_session_storage.py`

Disclosure notes

- this PR is intentionally bounded to sender/session isolation in shared ohmo chats and threads
- it does not claim a new remote admission bypass, slash-command privilege bypass, or plugin-execution issue
- no unrelated files were changed

`fix(ohmo): isolate gateway sessions by sender` ✓ `92f6400`

tjb-tech merged commit **3186851** into `HKUDS:main` 4 days ago

View details

4 checks passed

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

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