



CRITICAL BUG: Passkey Data Loss During JSON Round-Trip

Reported: 2025-12-07

CLI Version: 2.32.0

Severity: CRITICAL - Irreversible data loss

Summary

A pure JSON round-trip (**op item get --format json | op item edit <uuid>**) permanently deletes passkey data from LOGIN items.

The WebAuthn/FIDO2 private key is destroyed and cannot be recovered.

Reproduction Steps

```
# 1. Start with a LOGIN item that has a passkey
op item get <uuid-with-passkey> --format json > /tmp/before.json

# 2. Pure round-trip - NO modifications to the JSON
cat /tmp/before.json | op item edit <uuid-with-passkey>

# 3. Result: Passkey is DELETED from the item
```

Note: Both **op item edit <uuid>** and **op item edit <uuid>** produce identical results when piping JSON.

Evidence

Test Item

- **UUID:** `hqguqsaovla1odxkabkzujksa4`
- **Title:** Autofill
- **URL:** autofill.me
- **Had passkey:** Yes (FIDO2/WebAuthn)

Before Round-Trip (exported from 1Password UI)

```
{
  "overview": {
    "passkey": {
      "credentialId": "bC_HcuI8JIieZedDG-zn1U1AJQhtVSaNGILKcJ1M",
      "userHandle": "e7f8Z00qKit0V_yf-Z1V8Q8yiGbHdWWhVsmV8uW0hfI",
```

```

    "rpId": "autofill.me"
  },
  "details": {
    "passkey": {
      "type": "webauthn",
      "createdAt": 1765121634,
      "privateKey": "eyJrdHki0iJFQyIsImNydiI6IlAtMjU2IiwZCI6InJtRmd...<EC
P-256 PRIVATE KEY>",
      "userHandle": "e7f8Z00qKit0V_yf-Z1V8Q8yiGbHdWWhVsmV8uW0hfI"
    },
    "fields": [
      {
        "value": "1@email.com",
        "id": "identifier",
        "name": "",
        "type": "T",
        "designation": "username"
      }
    ]
  }
}

```

CLI Output (`op item get --format json`)

```

{
  "id": "hqguqsaovla1odxkabkzujska4",
  "title": "Autofill",
  "category": "LOGIN",
  "fields": [
    {"id": "username", "type": "STRING", "purpose": "USERNAME", "value": "1@email.com"},
    {"id": "password", "type": "CONCEALED", "purpose": "PASSWORD"},
    {"id": "notesPlain", "type": "STRING", "purpose": "NOTES"}
  ]
}

```

⚠ NOTE: No passkey data anywhere in CLI JSON output!

After Round-Trip (exported from 1Password UI)

```

{
  "overview": {
    "passkey": {
      "credentialId": "bC_HcuI8JIieZedDG-zn1U1AJQhtVSaNGILKcJ1M",
      "rpId": "autofill.me",
      "userHandle": "e7f8Z00qKit0V_yf-Z1V8Q8yiGbHdWWhVsmV8uW0hfI"
    }
  },

```

```
"details": {  
  "fields": [...],  
  "notesPlain": "",  
  "passwordHistory": [],  
  "sections": []  
}
```

✗ The **details.passkey** object is GONE - the WebAuthn private key has been permanently deleted!

Root Cause Analysis

1. **op item get --format json** does **NOT** include passkey data in its output
2. **op item edit <uuid>** interprets missing data as "delete this field"
3. Result: The passkey's WebAuthn private key is permanently destroyed

This is a **replace** semantic rather than a **merge** semantic. The CLI treats the input JSON as the complete desired state, not as a partial update.

Official Documentation Conflict

The 1Password CLI documentation explicitly supports this workflow:

EDIT AN ITEM USING A TEMPLATE

1. Get the item you want to edit in JSON format and save it to a file:
`op item get oldLogin --format=json > updatedLogin.json`
2. Edit the file.
3. Use the '--template' flag to specify the path to the edited file:
`op item edit oldLogin --template=updatedLogin.json`

You can also edit an item using piped input:

```
cat updatedLogin.json | op item edit oldLogin
```

The documented workflow assumes **op item get --format=json** returns all item data. Since passkey data is omitted from the output, the workflow silently destroys it.

No Recovery Options

- **op item get** has no **--include-passkey** or similar flag
 - CLI does not expose item version history
 - No way to recover deleted passkey data via CLI
-

Impact

Aspect	Details
Severity	CRITICAL
Data Lost	WebAuthn/FIDO2 EC P-256 private keys
Reversible	NO - private key is destroyed, CLI has no history access
User Impact	Must re-register passkey with each affected service
Scope	Any LOGIN item with passkey edited via <code>op item edit -</code>

Broken State Created

The bug creates a **corrupt/orphaned passkey state**:

Location	Data	Status
<code>overview.passkey</code>	Metadata (credentialId, rpId)	✅ Preserved
<code>details.passkey</code>	Private key (EC P-256)	❌ DELETED

Observed Behavior:

Component	Shows Passkey?	Authentication Works?
macOS 1Password app	❌ No	N/A
Safari extension	✅ Yes (offers to use)	❌ Fails

This is **worse than complete deletion** because:

- 1. User sees passkey offered in Safari autofill
- 2. User attempts to authenticate with it
- 3. Authentication fails (private key missing)
- 4. User has no indication the passkey is corrupted
- 5. User doesn't know to re-register

Workaround

DO NOT USE `op item edit <uuid>` ON ITEMS WITH PASSKEYS

For items with passkeys, use only field assignment syntax:

```
# ✅ SAFE - only modifies the specific field
op item edit <uuid> "Section.Field[text]=value"
op item edit <uuid> --tags "new-tag"

# ❌ UNSAFE - deletes passkey!
cat item.json | op item edit <uuid>
op item get <uuid> --format json | op item edit <uuid>
```

Detection

Before editing, check if item has a passkey:

```
# This will NOT show passkey (that's the bug), but you can check UI  
# or maintain a list of items known to have passkeys
```

Requested Fix Options

Option A: Merge semantics instead of replace (Preferred)

`op item edit` with piped/template JSON should preserve fields not present in input JSON, rather than deleting them. This is the safest fix as it:

- Prevents accidental deletion of any data not in CLI output
- Maintains backward compatibility for existing workflows
- Doesn't require exposing sensitive passkey private keys in CLI output

Option B: Add explicit delete syntax

Require explicit `"fieldName": null` or similar syntax to delete fields. Missing fields should be left unchanged.

Option C: Add safety flag

Add `--preserve-passkey` or `--no-delete-missing` flag to prevent accidental deletion of data not in the input JSON.

Option D: Warning/confirmation

When `op item edit` – would delete a passkey, prompt for confirmation or show a warning.

Questions for 1Password Support

1. Is this behavior intentional or a bug?
2. Is there a way to include passkey data in the CLI JSON output?
3. Is there a safe way to add fields (like REFERENCE links) to items with passkeys via CLI?
4. Are there other item types/fields that are similarly excluded from CLI output and would be deleted on round-trip?
5. Can passkey data be restored from 1Password's item history via UI or API?

Investigation Notes

CLI Flags Checked

```
$ op item get --help
# No --include-passkey, --full, or similar flag exists
# Available: --fields, --otp, --reveal, --share-link, --vault

$ op item --help
# Subcommands: create, get, edit, delete, list, move, share, template
# No "history" or "restore" command
```

Version History

```
$ op item get <uuid> --format json | jq '.version'
4 # Item is at version 4, but no way to access previous versions via CLI
```

Passkey Data Location

The passkey private key is stored in `details.passkey.privateKey` (Base64-encoded JWK):

```
{
  "details": {
    "passkey": {
      "type": "webauthn",
      "createdAt": 1765121634,
      "privateKey": "eyJrdHkiOiJFQyIsImNydiI6IlAtMjU2IiwiaCI6Ii4uLiJ9",
      "userHandle": "...
    }
  }
}
```

This entire object is missing from `op item get --format json` output.

Related Files

- `test - before.json` - Full item export before round-trip (with passkey)
- `test - after.json` - Full item export after round-trip (passkey deleted)