Cross-Realm Disclosure of Workflow History and Log in OpenXPKI

Security Advisory	
Advisory ID:	WRS-SA-2026-003
Severity:	low
CVE:	not assigned
Affected:	OpenXPKI Enterprise Edition and Community Edition through 3.32.17 and 3.34.3 (fixed in 3.32.18 and 3.34.4)
Published:	2026-11-16
Status:	Final v1.0

Executive Summary

A low-severity vulnerability in the workflow access control of **OpenXPKI Enterprise Edition** and **OpenXPKI Community Edition** allows an authenticated operator of one PKI realm to read the workflow history and the technical workflow log of workflows that belong to another realm on the same OpenXPKI installation. Successful exploitation discloses workflow metadata across the realm boundary: states, actions, the names of the users involved, timestamps, and the log messages written while the workflow was processed.

OpenXPKI separates realms on a shared installation: a workflow can only be loaded and displayed in the realm it was created in. The API commands that return the workflow history and the technical log do not apply this check. They load the requested data by workflow ID only, and decide access by the access rules of the caller's own realm. A role that may view the history and log of all workflows of a given type in its own realm can therefore view them for workflows of the same type in every other realm.

The attack path is the web interface. An operator logs in to a realm in which they hold such a role, for example **RA Operator**, and opens the history or log page of a workflow ID from another realm. The page displays the history and log of the foreign workflow. Workflow IDs are assigned sequentially across all realms and can be enumerated. The attacker cannot read the workflow context, cannot act on the workflow, and cannot access certificates, keys or other data of the foreign realm.

Exploitation requires an installation with more than one realm and an operator account. The default configurations of OpenXPKI EE and CE both contain two realms, but only the EE default exposes workflows across them, and only those of the root CA realm. The practical exposure is highest for installations that operate several issuing realms with shared workflow definitions and assign the operator roles of different realms to different people or organisations.

All versions of OpenXPKI EE and CE up to and including 3.32.17, and 3.34.0 through 3.34.3, are affected. No hotfix is provided. Upgrading to version 3.32.18 or 3.34.4, or any later version, provides a permanent fix.

Installations in which every operator holds the same role in all realms are not affected in practice, because those operators can already read the workflows of every realm.

Affected Systems

Vulnerable Versions

Product	Versions	Status
OpenXPKI Enterprise Edition (EE)	All versions through 3.32.17; 3.34.0 to 3.34.3	**Vulnerable**
OpenXPKI Community Edition (CE)	All versions through 3.32.17; 3.34.0 to 3.34.3	**Vulnerable**
OpenXPKI CE / EE	3.32.18 and later 3.32 releases	Fixed
OpenXPKI CE / EE	3.34.4 and later	Fixed

The affected API commands have never compared the realm of the requested workflow with the realm of the session. Versions prior to 3.30 are also vulnerable. Those versions are end-of-life; upgrading to a fixed version is the only remediation available.

An installation is exposed only if it runs more than one realm. Installations with a single realm are not affected.

Affected Interfaces

Interface	Affected	Condition for exploitation
WebUI	**Yes**	Login with a role that has the `history` or `techlog` permission for workflows of other users
RPC API	No	The RPC service does not expose the affected API commands
SCEP	No	The SCEP service does not expose the affected API commands
EST (RFC 7030)	No	The EST service does not expose the affected API commands
ACME	No	The ACME service does not expose the affected API commands

The `history` permission grants access to the workflow history: the sequence of states and actions with user names and timestamps. The `techlog` permission grants access to the technical log of the workflow. Both are set per role and workflow type in the `acl` section of the workflow definition. A role has access to the workflows of other users if its `creator` rule in the same section is `any`, or a pattern that matches the creator of the foreign workflow.

The OpenXPKI command line tools are not considered an attack path: they require administrative access to the server host.

OpenXPKI Community Edition — Default Configuration

The default CE configuration grants the **RA Operator** role the `history` and `techlog` permissions for all workflows of eleven workflow types, including certificate signing and revocation requests. No other role has either permission.

The default configuration contains the realms `democa` and `rootca`. The `rootca` realm defines only the system status workflow, which is not stored in the database. An RA Operator of `democa` therefore finds no workflow in `rootca` to read, and an RA Operator of `rootca` has no access rules for the workflow types of `democa`. A default CE installation is not exploitable in practice. It becomes exploitable as soon as a further realm with the shared workflow definitions is added, or workflows are added to `rootca`.

OpenXPKI Enterprise Edition — Default Configuration

The default EE configuration grants the **RA Operator** role the `history` and `techlog` permissions for all workflows of 36 workflow types, and the **LocalRA** role (label "LRA Operator") the `techlog` permission for all workflows of twelve workflow types.

The default configuration contains the realms `democa` and `rootca`. The `rootca` realm stores the workflows for CA certificate publication and CRL import, and `democa` grants the RA Operator role access to both types. An RA Operator of `democa` can therefore read the history and log of these workflows in `rootca`. The reverse direction is limited to the same two workflow types.

Multi-Realm Installations

Installations with several issuing realms commonly share one set of workflow definitions, for example through the `realm.tpl` directory. In such installations, every role with the `history` or `techlog` permission in one realm can read the history and log of all workflows of the shared types in every other realm. The exposure is relevant where the realms are operated by different people or organisations, for example by different RA teams or for different customers.

Determining Your Version

On the OpenXPKI server host, run:

```
openxpkictl version
```

Alternatively, query the installed package:

```
# OpenXPKI Community Edition (Debian / Ubuntu)
dpkg -l libopenxpki-perl | grep 'ii'

# OpenXPKI Enterprise Edition (Red Hat Enterprise Linux, SUSE SLES)
rpm -q myperl-openxpki-core

# OpenXPKI Enterprise Edition (Ubuntu LTS)
dpkg -l libopenxpki-myperl | grep 'ii'
```

Determining Your Realms

List the realms configured on the installation:

```
grep -E '^[a-zA-Z0-9_-]+:' /etc/openxpki/config.d/system/realms.yaml
```

Expected output for the default configuration:

```
democa:
rootca:
```

An installation that lists a single realm is not affected.

Technical Details

Vulnerability Description

OpenXPKI runs several realms on one server and one database. Each workflow is stored with the realm it was created in, and code that loads a workflow instance rejects a workflow from another realm. The workflow handler (`core/server/OpenXPKI/Workflow/Handler.pm`) and the workflow utility class (`core/server/OpenXPKI/Server/API2/Plugin/Workflow/Util.pm`) both perform this check before they load a workflow:

```
# We can not load workflows from other realms as this will break config and security
# The watchdog switches the session realm before instantiating a new factory
if (CTX('session')->data->pki_realm ne $wf->{pki_realm}) {
    OpenXPKI::Exception->throw(
        message => 'I18N_OPENXPKI_UI_WORKFLOW_HANDLER_NOT_IN_CURRENT_REALM',
        ...
```

The API commands `get_workflow_history` and `get_workflow_log` do not load the workflow instance. They read the rows of the `workflow_history` and `application_log` tables directly by workflow ID, after an access check that does not consider the realm:

```
# core/server/OpenXPKI/Server/API2/Plugin/Workflow/get_workflow_history.pm,
# lines 56-76 in v3.34.3 and v3.32.17
$self->api->check_workflow_acl( id => $wf_id )
or OpenXPKI::Exception->throw (
    message => 'I18N_OPENXPKI_UI_WORKFLOW_ACCESS_NOT_ALLOWED_FOR_USER',
);

my $wf_type = CTX('api2')->get_workflow_type_for_id( id => $wf_id );

$util->factory->can_access_handle($wf_type, 'history')
or OpenXPKI::Exception->throw ( ... );

my $history = CTX('dbi')->select_hashes(
    from => 'workflow_history',
    columns => [ '*' ],
```

```
    where => { workflow_id => $wf_id },
    ...
```

`get_workflow_log.pm` (lines 54–89 in v3.34.3) has the same structure, with the `techlog` permission and the `application_log` table.

The access check uses three API commands that read the workflow type, the creator and the tenant by workflow ID, without a realm condition: `check_workflow_acl`, `get_workflow_type_for_id`, `get_workflow_creator`. The workflow factory then evaluates the `acl` section of that workflow type in the configuration of the **caller's** realm (`OpenXPKI::Workflow::Factory::can_access_workflow`, line 367). If the caller's realm defines a workflow type with the same name and grants the caller's role access to workflows of any creator, the check succeeds for a workflow of any realm.

Trust Boundary Bypass

The realm check in the workflow handler works as intended: the API command `get_workflow_info` rejects a workflow from another realm with an error. The history and log commands bypass this control because they read the database tables directly instead of loading the workflow.

The WebUI does not compensate for the missing check. The history and log pages (`core/server/OpenXPKI/Client/Service/WebUI/Page/Workflow/History.pm` and `Log.pm`, line 18 in v3.34.3) call `get_workflow_info` with the option `nostatus`, which suppresses the error. They then call the history or log command regardless of the result.

Attack Mechanism

The attacker is logged in to the WebUI of realm A with a role that has the `history` or `techlog` permission and the creator rule `any` for some workflow type. They request the history or log page for the ID of a workflow of the same type in realm B. The WebUI identifies these pages as:

```
workflow!history!wf_id!<id>
workflow!log!wf_id!<id>
```

The page shows no error message. Its title lacks the workflow ID and type, but the page lists the history or log entries of the foreign workflow. Workflow IDs are assigned from one sequence shared by all realms. An attacker can therefore find the workflows of other realms by trying the IDs that are not visible in their own realm.

Attack Vectors

There is one attack path, the WebUI. It is available to the following roles, in decreasing order of required privilege.

Vector 1 — RA Operator (CE and EE default configuration)

The RA Operator role has the `history` and `techlog` permissions with the creator rule `any` for most stored workflow types in both default configurations. An RA Operator of one realm can read the history and log of the workflows of these types in every other realm that uses the same type names. In the default configurations this applies to the CA certificate publication and CRL import workflows of the EE `rootca` realm. In installations with several realms that share their workflow definitions, it applies to all certificate request, revocation and enrollment workflows.

Vector 2 — LRA Operator (EE default configuration)

The EE role `LocalRA` (label "LRA Operator") has the `techlog` permission with the creator rule `any` for twelve workflow types, including certificate signing, revocation and enrollment requests. An LRA Operator of one realm can read the technical log, but not the history, of the workflows of these types in other realms.

Vector 3 — Other roles (non-default configuration)

The default configurations do not grant the `history` or `techlog` permission to the User role. Where a configuration does, a user can read the workflows of another realm if the creator rule of the role matches the creator of the foreign workflow. With the rule `self`, this is the case when the same login name exists in both realms, for example because both realms authenticate against the same directory.

Impact

The attacker can read, for workflows of another realm:

- **History:** the sequence of states and actions, the name of the user who performed each action, and the timestamps.
- **Technical log:** all log messages written for the workflow, with timestamp and priority. Depending on the workflow type and the processing steps, these messages contain certificate subjects, certificate identifiers, user names, policy decisions and error details.

The attacker cannot read the workflow context or attributes, which hold the request data such as the CSR. They cannot perform any action on the foreign workflow: all

commands that change a workflow load the workflow instance first, which enforces the realm. Certificates, keys and other objects of the foreign realm are not affected.

The vulnerability was confirmed by the reporter on OpenXPKI CE 3.34.0 with two realms, using the RA Operator role. Users without the required permission were rejected as expected. The other versions and the EE configuration were assessed by source code review.

CVSS Score

The score below reflects the most accessible default-configuration attack path (Vector 1: RA Operator of `democa` reading workflows of the EE `rootca` realm).

Metric	Value
CVSS v3.1 Base Score	**2.7 (Low)**
Attack Vector	Network
Attack Complexity	Low
Privileges Required	High
User Interaction	None
Scope	Unchanged
Confidentiality	Low
Integrity	None
Availability	None

CVSS v3.1 vector: `CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:L/I:N/A:N`

Privileges Required is rated High because the attacker needs an operator role of the PKI, which grants control over certificate issuance in its realm. Scope is Unchanged because all realms are managed by the same OpenXPKI server, which enforces the realm separation. Confidentiality is rated Low because only workflow metadata and log messages are disclosed, not the request data, certificates or keys, and the attacker cannot choose which information the log contains.

> **Elevated score for non-default configurations:** Where a configuration grants the `history` or `techlog` permission to the User role (Vector 3), Privileges Required is Low, yielding a base score of **4.3 (Medium)**: `CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N`.

Remediation

Recommended Fix: Upgrade to OpenXPKI 3.32.18 or 3.34.4

The vulnerability is resolved in OpenXPKI CE and EE versions 3.32.18 and 3.34.4. The API command that reads the workflow type for the access check now rejects workflows from another realm, so the history and log commands fail for them. In addition, the WebUI history and log pages stop rendering when the workflow cannot be loaded. Upgrade installations running 3.32 to version 3.32.18 or a later 3.32 release, and all other installations to version 3.34.4 or later, as soon as the releases become available.

Because of the low severity, White Rabbit Security does not provide a hotfix.

Mitigation: Restrict the History and Log Permissions

No action is required on installations with a single realm, and on installations in which every operator holds the same role in all realms.

Where operators of one realm must not see the workflows of another realm, remove the `history` and `techlog` permissions from the operator roles until the installation is upgraded. The access rules of the caller's realm decide access, so the permissions must be removed in every realm whose operators must not read other realms. This also removes the history and log pages for the workflows of the operators' own realm.

List the workflow definitions that grant either permission:

```
grep -Rl -E '\s+(history|techlog):\s*1' /etc/openxpki/config.d/realm/*/workflow/def | sort
```

Expected output on an affected configuration (the list is shortened):

```
/etc/openxpki/config.d/realm/.../workflow/def/certificate_enroll.yaml
/etc/openxpki/config.d/realm/.../workflow/def/certificate_privkey_export.yaml
...
```

In each listed file, remove the lines `history: 1` and `techlog: 1` from the `acl` section of the affected roles, for example:

```
acl:
    RA Operator:
        creator: any
        fail: 1
        resume: 1
        wakeup: 1
```

Realms that are symbolic links to a shared directory such as `realm.tpl` share the same files. Then restart the OpenXPKI server process so that the changed configuration is loaded:

```
# Installations using the older FCGI setup
systemctl restart openxpkid

# Installations using the new client/server setup
systemctl restart openxpki-serverd
```

After all permissions are removed, the `grep` command above produces no output.

> **Important:** Configuration updates from the OpenXPKI sample configuration restore the permissions. Remove them again after any such update until the installation is upgraded to OpenXPKI 3.32.18, 3.34.4 or later.

Checking for Prior Exploitation

Reading the history or log of a workflow does not change the OpenXPKI database. Prior access can be identified from the web server access log, as long as it is retained.

The WebUI requests the history and log pages with the realm in the URL path and the workflow ID in the query string. List all such requests with the realm and the workflow ID:

```
zgrep -hoE '/webui/[^/]+/cgi-bin/webui\.fcgi\?[^ ]*' \
    /var/log/apache2/*access*.log* \
    /var/log/httpd/*access*log* 2>/dev/null \
  | sed 's/%21/!/g' \
  | sed -nE 's#^/webui/([^/]+)/.*(history|log)!wf_id!([0-9]+).*#\1 \2 \3#p' \
  | sort -u
```

Expected output, one line per realm, page and workflow ID:

```
democa history 511
democa log 511
```

Look up the realm of the listed workflows in the OpenXPKI database, for example:

```
SELECT workflow_id, pki_realm, workflow_type
  FROM workflow
 WHERE workflow_id IN (511);
```

Expected output:

```
workflow_id | pki_realm | workflow_type
------------+-----------+-------------------------------
        511 | democb    | certificate_signing_request_v2
```

A workflow whose `pki_realm` differs from the realm in the request was accessed from another realm. This comparison assumes the default WebUI configuration, in which the URL path selects the realm. If the WebUI maps realms by host name instead, use the host name of the request.

OpenXPKI server log. From OpenXPKI 3.28 on, the WebUI history and log pages first try to load the workflow. For a workflow of another realm this fails, and the server logs the error in its system log. Current sample configurations write this log to `/var/log/openxpki-server/openxpki.log`, older ones to `/var/log/openxpki/openxpki.log`:

```
zgrep -hE 'NOT_IN_CURRENT_REALM|not available in this realm' \
    /var/log/openxpki-server/openxpki.log* \
    /var/log/openxpki/openxpki.log* 2>/dev/null
```

Each line marks an attempt to open a workflow of another realm. Not every attempt is an exploitation: the same error occurs when an operator follows an outdated link to a workflow of another realm. The access log shows which page was requested.

Disclosure Timeline

Date	Event
2026-10-07	Vulnerability reported to White Rabbit Security by kta1kri; initial analysis begins
2026-10-08	Vulnerability confirmed; code fix committed to the development branch

Date	Event
2026-10-12	Affected versions and default configurations assessed
2026-10-27	OpenXPKI 3.32.18 and 3.34.4 released with the fix
2026-10-27	Draft advisory distributed to affected customer environments
2027-11-16	Public disclosure; advisory published in Final status

This advisory follows a coordinated vulnerability disclosure process. White Rabbit Security thanks kta1kri for identifying the vulnerability and for the professional and responsible manner in which it was reported and handled throughout the disclosure process.

References

- **CVE identifier** — pending assignment; this advisory will be updated upon registration
- **OpenXPKI Community Edition source repository** — `https://github.com/openxpki/openxpki`
- **Affected source files** — `core/server/OpenXPKI/Server/API2/Plugin/Workflow/get_workflow_history.pm` `core/server/OpenXPKI/Server/API2/Plugin/Workflow/get_workflow_log.pm` `core/server/OpenXPKI/Server/API2/Plugin/Workflow/check_workflow_acl.pm`
- **CVSS v3.1 Calculator** — `https://www.first.org/cvss/calculator/3.1`

Credits

This vulnerability was discovered by **kta1kri** and reported to White Rabbit Security on 2026-10-07, together with a reproduction on a two-realm test installation and a suggested fix.

For questions or additional information regarding this advisory, contact security@whiterabbitsecurity.com.