

Client Certificate Impersonation via Forwarded Proxy Headers in OpenXPKI

Security Advisory	
Advisory ID:	WRS-SA-2026-002
Severity:	critical
CVE:	not assigned
Affected:	OpenXPKI Enterprise Edition and Community Edition 3.32.2 and later behind an Apache configuration derived from the shipped sample (fixed by a configuration change)
Published:	2026-10-19
Status:	Final v1.1

Executive Summary

A critical vulnerability in the reverse-proxy integration of **OpenXPKI Enterprise Edition** and **OpenXPKI Community Edition** allows an unauthenticated remote attacker to present the TLS client certificate of another identity without possessing its private key. Successful exploitation results in certificate issuance, renewal and revocation, and WebUI logins, in the name of that identity.

The OpenXPKI client daemon (`openxpki-clientd`) does not see the TLS session itself. It learns the client certificate from HTTP headers that the reverse proxy adds to each request, and accepts them in two encodings. The Apache sample configuration shipped with OpenXPKI removes client-supplied copies of these headers in one encoding only, and only for two of the variables that `openxpki-clientd` reads. A client can therefore supply the headers itself, and `openxpki-clientd` treats them as if the proxy had verified a client certificate. Backend checks still confirm that the presented certificate is valid, but nothing confirms that the sender holds the matching private key.

The most accessible attack path exists in the default configuration of both editions and uses the **RPC** or **EST** enrollment endpoint. The attacker needs only the public certificate of an existing client, which is typically easy to obtain, and requests a renewal in that client's name. Renewal requests are typically processed without RA Operator approval. The WebUI client-certificate login is affected in the same way.

Installations behind the nginx sample configuration are not affected as long as nginx keeps its default handling of header names containing underscores. SCEP and ACME are not affected.

OpenXPKI EE and CE 3.32.2 and all later releases, including 3.34, are affected when operated behind an Apache configuration derived from the shipped sample. Releases before 3.32.2, including all 3.30 releases, are not affected, because they do not accept the header encoding used in the attack.

The vulnerability is a configuration issue. It is fixed by a change to the Apache configuration, and no OpenXPKI software update is required. White Rabbit Security provides instructions for the change and for verifying it. The Apache sample configuration shipped with OpenXPKI will be corrected accordingly.

Affected Systems

Vulnerable Versions

The vulnerability is in the Apache reverse-proxy configuration derived from the shipped sample, in combination with the Mojolicious-based client daemon `openxpki-clientd`. The client daemon is part of the OpenXPKI packages; the Apache configuration is deployed from the `openxpki-config` repository at installation time and is maintained by the operator.

Product	Versions	Status
OpenXPKI Enterprise Edition (EE)	3.32.2 and later, 3.34	Vulnerable
OpenXPKI Community Edition (CE)	3.32.2 and later, 3.34	Vulnerable
OpenXPKI CE / EE	3.32.0, 3.32.1	Not affected
OpenXPKI CE / EE	3.30 and earlier	Not affected

OpenXPKI 3.32.2 added support for the URL-escaped header encoding that the attack uses. Earlier releases accept only the Base64 encoding, which the Apache sample configuration already removes for the relevant variables.

No OpenXPKI release fixes the vulnerability, because the affected component is the deployed web server configuration. The configuration change described in the Remediation section fixes the vulnerability on all affected versions and does not depend on the installed package version.

Affected Web Server Configurations

Configuration	Affected	Condition
Apache, derived from the sample <code>apache2-openxpki-site.conf</code>	Yes	Unless both header encodings are removed for <code>SSL_CLIENT_CERT</code> and <code>SSL_CLIENT_S_DN</code>
nginx, derived from the sample <code>nginx-openxpki-site.conf</code>	No	Only while <code>underscores_in_headers</code> remains off (nginx default)
Legacy FCGI setup (Define <code>NoMojolicious</code>)	No	<code>openxpki-clientd</code> is not involved

The Apache sample is located at `contrib/apache2-openxpki-site.conf` in the `openxpki-config` repository; in OpenXPki 3.34 it moved to `files/etc/httpd/conf.d/apache2-openxpki-site.conf`. Its content is unchanged with respect to this issue.

nginx drops request headers whose names contain an underscore unless `underscores_in_headers` is enabled. All variables relevant to this advisory contain underscores. nginx installations that have enabled this setting are affected.

Affected Interfaces

All interfaces below are affected only over HTTPS.

Interface	Affected	Condition for exploitation
EST (RFC 7030)	Yes	Always for <code>simpleenroll</code> , <code>simplereenroll</code> and <code>simplerevoke</code> ; the signer certificate is always taken from the forwarded headers
RPC API	Yes	Methods whose <code>env</code> configuration includes <code>signer_cert</code> or <code>signer_dn</code> (the default <code>generic</code> endpoint does), or endpoints using an authentication stack of type <code>x509</code>
CertEP	Yes	Always for enrollment and revocation; the authentication certificate is always read from the forwarded headers
CMC	Yes	If based on the standard configuration, which uses <code>signer_cert</code> as authentication source
WebUI	Yes	Where a client-certificate (<code>x509</code>) login stack is offered

Interface	Affected	Condition for exploitation
SCEP	No	The signer is taken from the signed SCEP message, not from the TLS session
ACME	No	The ACME service does not read client-certificate data in the default configuration

What the attacker can achieve through an affected interface is limited by the backend policy of the endpoint: the presented certificate must be a valid, unrevoked certificate known to the PKI, and the endpoint's signer authorisation, renewal and revocation rules still apply.

OpenXPKI Community Edition — Default Configuration

The default CE configuration uses the Apache sample and exposes EST, RPC, SCEP and the WebUI.

- The default EST endpoint accepts requests signed by certificates matching `CN=.:pkiclient`, and treats renewal as eligible without RA Operator approval.
- The default RPC `generic` endpoint passes the signer certificate to the enrollment and revocation workflows, with the same signer rule and eligibility settings.
- The default realm configuration offers a client-certificate login to the WebUI with the role `User`.

OpenXPKI Enterprise Edition — Default Configuration

The default EE configuration uses the same Apache sample and the same EST and RPC signer policy as CE. In addition:

- The default EE RPC `generic` endpoint selects an x509 authentication stack whenever a client certificate is present.
- The default EE EST endpoint permits revocation requests for certificates issued through the same endpoint.
- The default CertEP endpoint authenticates enrollment and revocation requests with a client certificate.
- The default CMC endpoint authenticates enrollment requests with a client certificate.

Determining the Frontend

The vulnerability affects only installations that use the Mojolicious-based client daemon. Check for the presence of the `client.d` directory:

```
ls /etc/openxpki/client.d
```

An installation that uses `openxpki-clientd` shows at least the two directories:

```
service system
```

If the command reports `No such file or directory`, the installation uses the legacy CGI setup and is not affected.

Determining Your Web Server Configuration

List the Apache directives that remove client-supplied reverse-proxy headers:

```
grep -rn 'unset X-ReverseProxy-ENV' /etc/apache2 /etc/httpd 2>/dev/null
```

An affected configuration shows only the `X-ReverseProxy-ENV-` form, for example:

```
/etc/apache2/sites-enabled/openxpki.conf:34:
    RequestHeader unset X-ReverseProxy-ENV-$key
```

A corrected configuration also shows:

```
RequestHeader unset X-ReverseProxy-ENVPLAIN-$key
```

The Remediation section describes the complete check.

Technical Details

Vulnerability Description

`openxpki-clientd` runs behind a reverse proxy and does not see the TLS session itself. The proxy passes TLS session data, such as the client certificate (`SSL_CLIENT_CERT`) and its subject (`SSL_CLIENT_S_DN`), to `openxpki-clientd` in HTTP request headers. Two header encodings are accepted:

Header	Encoding	Used by
X-ReverseProxy-ENV-<VAR>	Base64	Apache sample configuration
X-ReverseProxy-ENVPLAIN-<VAR>	URL-escaped	nginx sample configuration

The `before_dispatch` hook in `OpenXPKI::Client::Web` (`core/server/OpenXPKI/Client/Web.pm`, lines 275–294 in v3.32.17, unchanged in v3.34.4) imports every header that matches either form into the request environment. There is no allow-list of accepted variable names. The code relies on the web server to remove any such headers that the client sent itself, as the header comment of the module states.

The Apache sample configuration (`contrib/apache2-openxpki-site.conf` in the `openxpki-config` repository) implements this removal in the `OxiPreventInjection` macro. The macro removes only the Base64 form, and it is applied to only two variables:

```
# contrib/apache2-openxpki-site.conf
# (openxpki-config, branch enterprise), lines 42-44 and 58-59
<Macro OxiPreventInjection $key>
    RequestHeader unset X-ReverseProxy-ENV-$key
</Macro>
...
# Prevent injection of Apache ENV vars from browser into Mojolicious
use OxiPreventInjection SSL_CLIENT_S_DN
use OxiPreventInjection SSL_CLIENT_CERT
```

Client-supplied headers in the URL-escaped form are therefore forwarded to `openxpki-clientd` unchanged, for every variable. The same applies to the Base64 form of all variables other than the two listed, including the certificate chain (`SSL_CLIENT_CERT_CHAIN_n`).

One further property of `openxpki-clientd` contributes to the issue:

- The forwarded values carry no proof of origin, such as a secret shared between proxy and client daemon.

Trust Boundary Bypass

The `OxiPreventInjection` macro is the control that separates values set by the reverse proxy from values supplied by the client. It works as intended for the encoding and the

variables it covers. `openxpki-clientd` also accepts the URL-escaped encoding, which the nginx integration uses, but the Apache sample does not remove it. The URL-escaped form is therefore an unguarded alternate input path around a working control.

The vulnerability is thus a defect of the Apache sample configuration and is fixed entirely in the web server configuration by removing both encodings of `SSL_CLIENT_CERT` and `SSL_CLIENT_S_DN`. The subject CN (`SSL_CLIENT_S_DN_CN`) is not exploitable on its own; removing it as well is recommended as hardening.

Attack Mechanism

The attacker connects to the reverse proxy over HTTPS without presenting a TLS client certificate. Apache then sets no client-certificate headers of its own. The attacker adds the URL-escaped client-certificate headers to the request, filled with the public certificate and subject of another identity. Apache forwards them, and `openxpki-clientd` treats the request exactly like one for which the proxy has verified that certificate. The table lists the headers of the forwarded request as `X-ReverseProxy-<Encoding>-<Variable>`:

Variable	Encoding	Origin	Removed by the Apache sample
<code>SSL_CLIENT_CERT</code>	<code>ENV</code>	Apache, only if a client certificate was verified	Client copies: yes
<code>SSL_CLIENT_CERT</code>	<code>ENVPLAIN</code>	Client	No
<code>SSL_CLIENT_S_DN</code>	<code>ENVPLAIN</code>	Client	No
<code>SSL_CLIENT_CERT_CHAIN_n</code>	<code>ENV, ENVPLAIN</code>	Client	No

The attacker needs only the public certificate of the identity to impersonate, not its private key. Public certificates are not secret: they are sent in every TLS handshake, published in directories, and can often be retrieved from the PKI itself.

The OpenXPKI backend still validates the presented certificate: it must chain to a trusted issuer, be unrevoked, and satisfy the signer and eligibility rules of the endpoint. These checks confirm that the certificate is valid. They do not confirm that the sender holds the matching private key, because that proof is only ever made in the TLS handshake with the proxy.

Client-certificate data is only evaluated for requests that the proxy marks as HTTPS. Apache sets this marker (`X-Forwarded-Proto`) itself, so the client cannot influence it.

Attack Vectors

Three exploitation paths exist in the default configuration. All of them require network access to the HTTPS endpoint and the public certificate of an identity that the respective interface accepts. They are listed in decreasing order of default exposure.

Vector 1 — EST enrollment, renewal and revocation (CE and EE default configuration)

The EST service (`OpenXPKI::Client::Service::EST`) takes the signer certificate for `simpleenroll`, `simplereenroll` and `simplerevoke` from the forwarded client certificate without further condition. The enrollment workflow derives its mode from the signer certificate:

- **Renewal:** with the certificate of an existing EST client, the attacker submits a certificate request with that client's subject and a key pair of the attacker's choosing. The PKI issues a certificate for the client's identity bound to the attacker's key.
- **On-behalf enrollment:** with the certificate of an authorised signer (in the default configuration any certificate matching `CN=.:pkiclient`), the attacker requests certificates for other identities within the signer rules of the endpoint.
- **Revocation:** in the EE default configuration, the attacker can revoke certificates issued through the same endpoint.

Renewal requests are typically processed without RA Operator approval, so the renewal path usually requires no user interaction. Whether on-behalf enrollment is approved automatically depends on the policy of the endpoint.

Vector 2 — RPC API (CE and EE default configuration)

The RPC service takes the signer certificate, its subject and its chain from the forwarded headers for every method whose `env` configuration includes `signer_cert` or `signer_dn`. The default `generic` endpoint passes the signer certificate to its enrollment and revocation workflows, with the same signer rules and eligibility settings as EST. The enrollment workflow derives renewal and on-behalf enrollment from the signer certificate in the same way as for EST, so the renewal path described under Vector 1 applies here as well. In the EE default configuration, the presence of a client certificate also selects an x509 authentication stack for the backend session, which assigns the role configured for that stack.

Vector 3 — WebUI client-certificate login

Where the WebUI offers a login with a client certificate, the login handler reads the certificate, its subject and its chain from the forwarded headers. The attacker is logged in as the owner of the presented certificate, with the role that the authentication stack assigns to it. The default CE realm configuration offers such a login with the role `User`.

If a deployment maps client certificates to operator roles, the attacker obtains RA or CA Operator access with the certificate of an operator.

Non-default configuration — nginx with `underscores_in_headers` on

The nginx sample is not affected because nginx drops request headers whose names contain an underscore by default. All relevant variables contain underscores. If this setting has been enabled, Vectors 1 to 3 apply to nginx installations in the same way.

Impact

A successful attack lets an unauthenticated client act as any identity whose public certificate it holds, on every affected interface. Depending on the interface and the endpoint policy, this results in:

- issuance of a certificate for another identity bound to a key controlled by the attacker, which the attacker can then use to authenticate as that identity to any relying party that trusts the PKI;
- issuance of certificates for further identities within the rules for authorised signers;
- revocation of certificates, disrupting the services that depend on them;
- access to the WebUI with the privileges of the impersonated user.

Requests made in this way are indistinguishable from legitimate requests in the OpenXPKI workflow data (see Checking for Prior Exploitation).

White Rabbit Security has confirmed Vector 2 on a test system running a vulnerable version behind the Apache sample configuration. Vectors 1 and 3 are derived from source code review and have not been tested separately. They read the same forwarded headers, and Vector 1 passes the signer certificate to the same enrollment logic as Vector 2, so both are expected to be exploitable in the same way. The CertEP and CMC services read the authentication certificate from the same forwarded headers and are affected in the same way.

CVSS Score

The score below reflects the most accessible default-configuration attack path (Vectors 1 and 2: renewal via EST or RPC, processed without RA Operator approval).

Metric	Value
CVSS v3.1 Base Score	10.0 (Critical)
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User Interaction	None
Scope	Changed
Confidentiality	High
Integrity	High
Availability	None

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

Privileges Required is rated None because the attacker needs no account and no credential. A public certificate is not a credential, since the attack does not require the matching private key. Attack Complexity is rated Low because public certificates of existing EST and RPC clients are generally available, and the default configuration requires no further preconditions. User Interaction is None because renewal requests are typically processed without RA Operator approval: EST and RPC are used for unattended renewal of devices and services. Scope is Changed because the issued certificates are trusted by relying parties outside the vulnerable component, which the attacker can then access under the impersonated identity. Confidentiality and Integrity are rated High for this reason. Availability is rated None for this path; the revocation path described under Vector 1 additionally affects availability, which does not change the base score.

Score for endpoints with RA Operator approval: Where renewal requires RA Operator approval, User Interaction is Required, yielding a base score of **9.3 (Critical)**: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:N. The forged request looks like a legitimate request from the impersonated identity, so the operator has no indication to reject it.

Remediation

Fix: Correct the Apache Configuration

The vulnerability is a configuration issue and is fixed on all affected versions by a change to the Apache configuration. No OpenXPKI software update is required: no OpenXPKI files are modified, and the OpenXPKI services do not need to be restarted.

The Apache sample configuration in the `openxpki-config` repository will be corrected accordingly. Upgrading the OpenXPki packages does not change an Apache configuration that has already been deployed, so the change below must be applied to every existing installation.

Locating the Configuration

The change is made in the Apache site configuration derived from the OpenXPki sample. Typical locations are:

Installation	File
Debian / Ubuntu, file copied from the sample	<code>/etc/apache2/sites-enabled/openxpki.conf</code>
OpenXPki 3.34, installed from the files/ overlay	<code>/etc/httpd/conf.d/apache2-openxpki-site.conf</code>

To find the file on a given system, search for the macro:

```
grep -rln 'OxiPreventInjection' /etc/apache2 /etc/httpd 2>/dev/null
```

Expected output (the file name might differ):

```
/etc/apache2/sites-enabled/openxpki.conf
```

Only one file is expected. If the command reports more than one file, apply the change to each of them. If the reported file is a symbolic link, edit the link target.

The relevant part of the file is located near the top, before the first `<VirtualHost>` block. In a configuration created from the uncorrected sample, it reads:

```
<Macro OxiPreventInjection $key>
    RequestHeader unset X-ReverseProxy-ENV-$key
</Macro>
...
# Preparation for Mojolicious based client services
<IfModule mod_headers.c>
    RequestHeader set X-Forwarded-Proto expr=%{REQUEST_SCHEME}
    RequestHeader set X-ReverseProxy-ENVSET 1
    ...
```

```
# Prevent injection of Apache ENV vars from browser into Mojolicious
use OxiPreventInjection SSL_CLIENT_S_DN
use OxiPreventInjection SSL_CLIENT_CERT
</IfModule>
```

Applying the Change

One change is required to fix the vulnerability.

Extend the macro (mandatory). Add the ENVPLAIN line to the definition of the OxiPreventInjection macro, so that it removes both header encodings:

```
<Macro OxiPreventInjection $key>
  RequestHeader unset X-ReverseProxy-ENV-$key
  RequestHeader unset X-ReverseProxy-ENVPLAIN-$key
</Macro>
```

The existing lines use OxiPreventInjection SSL_CLIENT_S_DN and use OxiPreventInjection SSL_CLIENT_CERT in the <IfModule mod_headers.c> block shown above then remove both encodings of the client certificate and its subject. They must remain in place.

Recommended Configuration

White Rabbit Security additionally recommends applying the macro to the further variables that openxpki-clientd reads, including the subject CN and the certificate chain. Add these lines directly after the existing use OxiPreventInjection lines:

```
use OxiPreventInjection SSL_CLIENT_S_DN_CN
use OxiPreventInjection SSL_CLIENT_CERT_CHAIN_0
use OxiPreventInjection SSL_CLIENT_CERT_CHAIN_1
use OxiPreventInjection SSL_CLIENT_CERT_CHAIN_2
use OxiPreventInjection SSL_CLIENT_CERT_CHAIN_3
use OxiPreventInjection REMOTE_USER
use OxiPreventInjection OPENXPKI_USER
use OxiPreventInjection OPENXPKI_GROUP
```

If the configuration forwards further variables to openxpki-clientd, for example for single sign-on via use OxiForwardEnv <VAR> or RequestHeader set X-ReverseProxy-ENV-<VAR>, add a use OxiPreventInjection <VAR> line for each of them as well.

The corrected Apache sample configuration will contain the mandatory and the recommended lines by default.

Validating and Activating the Change

Validate the configuration:

```
# Debian / Ubuntu
apache2ctl configtest

# Red Hat Enterprise Linux, SUSE SLES
apachectl configtest
```

Expected output:

Syntax OK

Reload Apache so that the change takes effect:

```
# Debian / Ubuntu
systemctl reload apache2

# Red Hat Enterprise Linux
systemctl reload httpd

# SUSE SLES
systemctl reload apache2
```

Important: Configurations created from an uncorrected Apache sample contain the incomplete macro. Apply the change again whenever the web server configuration is recreated from such a sample.

Verifying the Change

Check that the macro removes both encodings:

```
grep -rn 'unset X-ReverseProxy-ENV' /etc/apache2 /etc/httpd 2>/dev/null
```

The output must contain both lines of the macro, for example:

```
/etc/apache2/sites-enabled/openxpki.conf:34:
    RequestHeader unset X-ReverseProxy-ENV-$key
/etc/apache2/sites-enabled/openxpki.conf:35:
    RequestHeader unset X-ReverseProxy-ENVPLAIN-$key
```

Check that the macro is applied to all required variables:

```
grep -rhn 'use OxiPreventInjection' /etc/apache2 /etc/httpd 2>/dev/null
```

The output must contain the lines for `SSL_CLIENT_S_DN` and `SSL_CLIENT_CERT`. With the recommended configuration applied, it also contains the lines for `SSL_CLIENT_S_DN_CN`, the four `SSL_CLIENT_CERT_CHAIN_n` variables, `REMOTE_USER`, `OPENXPKI_USER` and `OPENXPKI_GROUP`:

```
46: use OxiPreventInjection SSL_CLIENT_S_DN
47: use OxiPreventInjection SSL_CLIENT_CERT
48: use OxiPreventInjection SSL_CLIENT_S_DN_CN
49: use OxiPreventInjection SSL_CLIENT_CERT_CHAIN_0
50: use OxiPreventInjection SSL_CLIENT_CERT_CHAIN_1
51: use OxiPreventInjection SSL_CLIENT_CERT_CHAIN_2
52: use OxiPreventInjection SSL_CLIENT_CERT_CHAIN_3
53: use OxiPreventInjection REMOTE_USER
54: use OxiPreventInjection OPENXPKI_USER
55: use OxiPreventInjection OPENXPKI_GROUP
```

Mitigation: nginx Reverse Proxy

Installations behind the nginx sample configuration are not affected while `underscores_in_headers` is off, which is the nginx default. Verify that the setting has not been enabled:

```
nginx -T 2>/dev/null | grep -n 'underscores_in_headers'
```

No output, or only `underscores_in_headers off;`, means the installation is not affected. If the setting is on, remove client-supplied `X-ReverseProxy-ENV-*` and `X-ReverseProxy-ENVPLAIN-*` headers for all forwarded variables, or contact White Rabbit Security.

Mitigation: Reduce Endpoint Exposure

Until the configuration has been corrected, the following measures reduce the impact. They do not fix the vulnerability.

- Restrict network access to the EST and RPC endpoints to the client networks that need them.
- Require RA Operator approval for renewal and on-behalf enrollment on EST and RPC endpoints that currently accept them automatically.

Checking for Prior Exploitation

A request exploiting this vulnerability produces the same workflow data as a legitimate request from the impersonated identity. In the default configuration, neither OpenXPki nor Apache records whether the client presented a TLS certificate: `openxpki-clientd` logs the imported headers only at debug level, and Apache does not log request headers or client-certificate data by default. A definitive check is therefore not possible after the fact.

To identify suspicious activity, review renewals, on-behalf enrollments and revocations received via EST and RPC with a signer certificate since an affected version was installed. Treat a request as suspicious if it came from an unexpected source address, or if web server logs that record the TLS client certificate (for example via `%{SSL_CLIENT_S_DN}x` in the `LogFormat`) show no client certificate for that request.

Disclosure Timeline

Date	Event
2026-09-28	Vulnerability identified by White Rabbit Security during an internal review of the enrollment-interface documentation; initial analysis begins
2026-09-28	Vulnerability confirmed on a test system via the RPC interface
2026-09-28	Affected version range confirmed
2026-09-28	Configuration fix tested
2026-09-29	Apache sample configuration corrected in the <code>openxpki-config</code> repository
2026-09-29	Affected version range updated (<code><= v3.32.2</code> unaffected)
2026-09-29	Draft advisory distributed to affected customer environments; CVE registration initiated
2026-10-19	Public disclosure; advisory published in Final status. No CVE identifier was assigned

This advisory follows a coordinated vulnerability disclosure process. The vulnerability was identified internally; no external report was received.

References

- **CVE identifier** — not assigned
- **OpenXPKI Community Edition source repository** —
<https://github.com/openxpki/openxpki>
- **OpenXPKI sample configuration repository** —
<https://github.com/openxpki/openxpki-config>
- **Affected sample configuration** —
contrib/apache2-openxpki-site.conf (3.34:
files/etc/httpd/conf.d/apache2-openxpki-site.conf)
- **Source file that imports the forwarded headers** —
core/server/OpenXPKI/Client/Web.pm
- **Apache mod_headers reference** —
https://httpd.apache.org/docs/2.4/mod/mod_headers.html
- **nginx underscores_in_headers reference** —
https://nginx.org/r/underscores_in_headers
- **CVSS v3.1 Calculator** —
<https://www.first.org/cvss/calculator/3.1>

Credits

This vulnerability was identified by **White Rabbit Security** during an internal review of the OpenXPKI enrollment-interface documentation on 2026-09-28.

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