OpenXPKI Setup Guide for CBOM PKI Sensor

---

1. Purpose

The CBOM PKI sensor (`Discover_PKI`) integrates with **OpenXPKI**, the open source certificate management platform built around a realm scoped workflow engine, to discover and inventory cryptographic certificate assets issued by an OpenXPKI deployment.

During each discovery cycle the sensor queries a dedicated **OpenXPKI RPC endpoint** (`/rpc/<endpoint>`) in read only mode, invoking the `search_cert` and `get_cert` workflow actions, and collects the following certificate metadata:

- **Subject DN**, Common Name, Organization, and other distinguished name fields
- **Subject Alternative Names (SANs)**, DNS names, IP addresses, email addresses
- **Validity period**, `notbefore` and `notafter` timestamps
- **Issuer DN and CA alias**, the issuing CA's logical name within the realm
- **Certificate profile**, the OpenXPKI certificate profile used at issuance (e.g. `tls_server` , `user`)
- **Workflow state**, `ISSUED` , `REVOKED` , `EXPIRED` , or `CRL_ISSUANCE_PENDING`
- **Realm**, the OpenXPKI realm the certificate belongs to (OpenXPKI supports multiple isolated realms per instance)
- **Key algorithm and size**, RSA, ECC, and associated bit/curve length
- **Serial number and identifier**, the certificate's `cert_identifier` and X.509 serial

This guide covers registering a read only RPC endpoint scoped to certificate search, issuing sensor credentials, and configuring the `Discover_PKI` sensor YAML.

---

2. Prerequisites

Before beginning, confirm the following:

1. You have shell access to the **OpenXPKI server** with permission to edit files under `/etc/openxpki/config.d/` and run `openxpkictl` .
2. OpenXPKI **3.16 or later** is deployed with the RPC interface enabled (`openxpki-rpc-enable` , or the RPC vhost already present in your webserver config).
3. The CBOM sensor host has **HTTPS network connectivity** to the OpenXPKI RPC vhost (default port `443`).
4. A valid **TLS certificate** is installed on the RPC vhost, or the CBOM sensor is configured to trust the issuing CA, including OpenXPKI's own internal CA if the RPC endpoint is served on a self signed or internally issued certificate.
5. You know the **realm name** (e.g. `democa`) the sensor should query, and the **base URL** of the RPC vhost (e.g. `https://pki.example.com`).
6. Python 3.9+ or the CBOM sensor binary is installed on the host that will run `Discover_PKI` .

7. The CBOM configuration directory is writable by the user running the sensor process.

---

3. Register a Read Only RPC Endpoint

OpenXPKI does not expose a general purpose REST API by default, RPC access is granted per named endpoint, each scoped to a specific set of workflow actions. A dedicated endpoint keeps the CBOM sensor's access auditable and limited to search operations.

3.1 Create the RPC Endpoint Configuration

1. On the OpenXPKI server, navigate to the realm's RPC configuration directory:

```
bash cd /etc/openxpki/config.d/realm/democa/rpc/
```

1. Create a new endpoint definition file named after the endpoint (the filename becomes the URL path segment):

```
bash sudo vi cbom-discover.yaml
```

1. Populate it with a minimal configuration exposing only `search_cert` and `get_cert`:

```
yaml # /etc/openxpki/config.d/realm/democa/rpc/cbom-discover.yaml openapi: enabled: 0 output: format: json method: search_cert: workflow: search_cert param: status: ISSUED,REVOKED,EXPIRED output: - cert_identifier - subject - subject_key_identifier - notbefore - notafter - status get_cert: workflow: get_cert param: format: DER output: - cert_identifier - subject - issuer_dn - notbefore - notafter - subject_alt_name - profile
```

1. Reload the OpenXPKI server so the new endpoint is picked up:

```
bash sudo openxpkictl reload
```

> **Note:** Do not add `revoke_cert`, `enroll`, or any issuance related workflow to this endpoint. The CBOM sensor requires only `search_cert` and `get_cert`.

3.2 Confirm the Endpoint Is Reachable

1. Check the OpenXPKI system log for a successful endpoint registration message:

```
bash sudo tail -n 50 /var/log/openxpki/openxpki.log | grep cbom-discover
```

1. Confirm the endpoint appears at `https://pki.example.com/rpc/cbom-discover`. A `POST` with no body should return a JSON error (not a `404`), confirming the endpoint is routed correctly.

---

4. Configure Authentication for the RPC Endpoint

OpenXPKI RPC endpoints support two authentication modes for inbound requests: **TLS client certificate authentication** and **HTTP Basic Authentication** mapped to an OpenXPKI account. Choose one based on your deployment policy.

4.1 Issue a Client Authentication Certificate (Recommended)

1. Using the realm's internal CA (or your organization's PKI), issue a client authentication certificate for the sensor, e.g. with common name `cbom-sensor-svc`.
2. Map the certificate's subject to a read only role in the realm's authentication configuration:

`yaml # /etc/openxpki/config.d/realm/democa/auth/handler.yaml CbomSensorClientCert: type: ClientX509 role: '"RA Operator"' cacert: /etc/openxpki/tls/chain/democa-internal-ca.pem`

1. Copy the issued certificate and private key to the CBOM sensor host, e.g. `/etc/cbom/certs/cbom-sensor.crt` and `/etc/cbom/certs/cbom-sensor.key`.
2. Reload the server: `sudo openxpkictl reload`.

4.2 Alternative: HTTP Basic Authentication

1. Create a dedicated OpenXPKI account with the **RA Operator** role (or a custom read only role) via the OpenXPKI web UI under **Users**.
2. Assign it the username `cbom-sensor-svc` and a strong password.
3. Confirm the RPC endpoint's `auth` handler accepts Basic Auth for this role, or add a `Password` handler entry alongside the client cert handler in `handler.yaml`.
4. Store the password in your secrets manager, never in the CBOM configuration file directly.

> **Note:** Client certificate authentication is preferred for production deployments; Basic Auth is simpler for lab/PoC environments but requires periodic password rotation to remain within security policy.

---

5. Verify RPC Access

Before configuring the sensor, validate that the RPC endpoint responds correctly using either authentication method.

5.1 Test with Client Certificate

```
curl -s -X POST \
  "https://pki.example.com/rpc/cbom-discover" \
  --cert /etc/cbom/certs/cbom-sensor.crt \
  --key /etc/cbom/certs/cbom-sensor.key \
  -H "Content-Type: application/json" \
  -d '{"method": "search_cert", "params": {"status": "ISSUED", "limit": 5}}' \
  | python3 -m json.tool
```

5.2 Test with HTTP Basic Auth

```
curl -s -X POST \
  "https://pki.example.com/rpc/cbom-discover" \
  -u "cbom-sensor-svc:<password>" \
  -H "Content-Type: application/json" \
```

```
  -d '{"method": "search_cert", "params": {"status": "ISSUED", "limit": 5}}' \
  | python3 -m json.tool
```

> **Note:** The RPC endpoint name (`cbom-discover` in this guide) must exactly match the YAML filename created in Section 3.1. Requesting a mismatched endpoint name returns a generic `400 Bad Request` rather than a helpful error, which is the most common source of confusion during first time setup.

A successful response returns a JSON object with a `result.data` array of certificate identifiers. Confirm fields such as `cert_identifier`, `subject`, and `notafter` are present.

---

6. Configure CBOM Sensor

Create or update the sensor configuration file at the path recognized by your CBOM installation. The file is typically located under `config/sensors/` within the CBOM working directory.

```
# config/sensors/openxpki_pki.yaml
sensor:
  name: Discover_PKI
  vendor: OpenXPKI
  product: OpenXPKI Community Edition
  category: PKI
  enabled: true
  schedule: "0 3 * * *"          # Daily at 03:00 local time (cron syntax)
  timeout_seconds: 120

connection:
  base_url: "https://pki.example.com"
  rpc_endpoint: "cbom-discover"   # Must match the endpoint name from Section 3.1
  realm: "democa"
  verify_ssl: true
  ca_bundle: "/etc/ssl/certs/openxpki-internal-ca.pem"  # Path to CA bundle; set to
true for system store

auth:
  method: "client_cert"           # Options: "client_cert", "basic"

  # --- TLS Client Certificate (recommended) ---
  client_cert: "/etc/cbom/certs/cbom-sensor.crt"
  client_key: "/etc/cbom/certs/cbom-sensor.key"

  # --- Basic Auth (uncomment if using Basic Auth instead) ---
  # username: "cbom-sensor-svc"
  # password: "${OPENXPKI_PASSWORD}"   # Use env var reference; never hardcode

discovery:
  rpc_method: "search_cert"
  detail_method: "get_cert"       # Called per-identifier to fetch full metadata
  page_size: 100                  # Number of certificates per paginated request
  max_pages: 500                  # Safety cap; remove or increase for large
inventories
  status_filter: "ISSUED,REVOKED,EXPIRED"
  include_expired: true
```

```
  include_revoked: false          # Revoked certificates excluded from active CBOM by
default
  filters:
    # Optional OpenXPKI search_cert filters (passed as RPC params).
    # profile: "tls_server"
    # issuer_dn: "CN=Democa Issuing CA,O=Example Org"

output:
  format: "cbom-json"             # Output format consumed by the CBOM aggregator
  destination: "/var/cbom/output/openxpki_pki_discovery.json"
  append_timestamp: true

logging:
  level: "INFO"                   # DEBUG | INFO | WARNING | ERROR
  log_file: "/var/log/cbom/discover_pki_openxpki.log"
  max_size_mb: 50
  backup_count: 5
```

Replace all placeholder values before running the sensor:

Placeholder	Replace With
`pki.example.com`	Actual OpenXPKI RPC vhost hostname
`cbom-discover`	RPC endpoint name created in Section 3.1
`democa`	Target realm name
`/etc/cbom/certs/cbom-sensor.crt` / `.key`	Client certificate and key issued in Section 4.1
`${OPENXPKI_PASSWORD}`	Environment variable or secrets manager reference (Basic Auth only)
`/etc/ssl/certs/openxpki-internal-ca.pem`	Path to CA bundle, or `true` for system store

---

7. Validation

After writing the configuration file, run the sensor in dry run mode to confirm successful authentication and certificate discovery before enabling the scheduled job.

1. If using Basic Auth, set the required environment variable:

```
export OPENXPKI_PASSWORD="your-actual-password-here"
```

1. Execute the sensor with the `--dry-run` flag:

```
cbom-sensor run --config /path/to/config/sensors/openxpki_pki.yaml --dry-run
```

1. Confirm the output contains log lines similar to:

```
[INFO] Discover_PKI: Authenticated successfully to https://pki.example.com/rpc/cbom-
discover
[INFO] Discover_PKI: Page 1 -- retrieved 100 certificate identifiers
[INFO] Discover_PKI: Fetching detail records via get_cert...
[INFO] Discover_PKI: Discovery complete. Total certificates found: 312
[INFO] Discover_PKI: Output written to /var/cbom/output/openxpki_pki_discovery.json
```

1. Inspect the output file to confirm certificate fields are populated:

```
python3 -c "
import json
with open('/var/cbom/output/openxpki_pki_discovery.json') as f:
    data = json.load(f)
print(f'Total records: {len(data[\"certificates\"])}')
print('Sample record keys:', list(data['certificates'][0].keys()))
"
```

1. Enable the sensor in the CBOM scheduler once validation passes:

```
cbom-sensor enable Discover_PKI
```

---

8. Common Errors and Resolution

Error 1: `400 Bad Request`, Unknown RPC Endpoint

Symptom:

```
[ERROR] Discover_PKI: HTTP 400 Bad Request -- POST /rpc/cbom-discover
[ERROR] Discover_PKI: Response body: {"error":{"code":400,"message":"Could not
determine target endpoint"}}
```

Cause: The `rpc_endpoint` value in the sensor configuration does not match a YAML filename under `config.d/realm/<realm>/rpc/` on the server, or the realm reload in Section 3.1 did not complete.

Resolution:

1. Confirm the endpoint filename on the server exactly matches `rpc_endpoint` in the sensor config (case sensitive, without the `.yaml` extension).
2. Re run `sudo openxpkictl reload` on the OpenXPKI server and check `/var/log/openxpki/openxpki.log` for reload errors.
3. Re run the curl test from Section 5.1 manually against the exact URL the sensor is configured to call.

Error 2: `403 Forbidden`, Workflow Not Permitted for Role

Symptom:

```
[ERROR] Discover_PKI: HTTP 403 Forbidden -- POST /rpc/cbom-discover
[ERROR] Discover_PKI: Response body: {"error":{"code":403,"message":"Workflow
search_cert not permitted for role RA Operator"}}
```

Cause: The role mapped to the sensor's client certificate or Basic Auth account does not have permission to execute the `search_cert` or `get_cert` workflow in this realm.

Resolution:

1. Confirm the role assigned in `handler.yaml` (Section 4.1) or the account's role (Section 4.2) has ACL entries permitting `search_cert` and `get_cert`.
2. Check the realm's workflow ACL configuration under `config.d/realm/<realm>/workflow/def/` for the relevant `acl` stanza.
3. Re run the sensor after confirming ACLs, and consult the OpenXPKI system log for the exact role the request authenticated as.

Error 3: `SSL: CERTIFICATE_VERIFY_FAILED`

Symptom:

```
[ERROR] Discover_PKI: Connection error -- SSL: CERTIFICATE_VERIFY_FAILED
[ERROR] Discover_PKI: certificate verify failed: unable to get local issuer
certificate (_ssl.c:1129)
```

Cause: The RPC vhost presents a TLS certificate issued by OpenXPKI's own internal CA (common in lab/PoC deployments), which is not trusted by the sensor host's system certificate store.

Resolution:

1. Export the realm's internal CA certificate in PEM format from OpenXPKI (**Certificates > CA Certificates** in the web UI, or `openxpkiadm` on the server).
2. Copy it to the sensor host, for example `/etc/ssl/certs/openxpki-internal-ca.pem`.
3. Update the YAML configuration to reference the bundle:

```
connection:
  verify_ssl: true
  ca_bundle: "/etc/ssl/certs/openxpki-internal-ca.pem"
```

1. Do **not** set `verify_ssl: false` in production, this disables TLS certificate validation entirely and exposes the sensor to man in the middle attacks.

---

9. Security Recommendations

- **Prefer TLS client certificate authentication over Basic Auth** for the RPC endpoint. Client certificates avoid transmitting a long lived password and can be scoped to a short validity period with automated renewal.

- **Scope the RPC endpoint to `search_cert` and `get_cert` only.** Never add issuance, revocation, or approval workflows to the endpoint the CBOM sensor authenticates against, a compromised sensor credential should never be able to issue or revoke certificates.
- **Restrict the mapped role to read only ACLs.** Review the realm's workflow ACL definitions periodically to confirm the sensor's role has not accumulated additional permissions through configuration drift.
- **Store client certificate private keys and Basic Auth passwords in an external secrets manager** (e.g. HashiCorp Vault, AWS Secrets Manager, Azure Key Vault) and inject them at runtime. Never commit them to source control or the CBOM configuration file directly.
- **Enable and review OpenXPKI's system audit log** for the sensor's role or client certificate subject, confirming only `search_cert` and `get_cert` calls appear over time. Any unexpected workflow invocation associated with the sensor's identity should be investigated immediately.

---

10. Conclusion

This guide has walked through registering a dedicated, read only OpenXPKI RPC endpoint, authenticating it with either a TLS client certificate or a scoped Basic Auth account, and configuring the `Discover_PKI` sensor YAML to query the endpoint's `search_cert` and `get_cert` workflows. With scheduled execution, the CBOM sensor maintains a continuously updated inventory of certificate metadata including subject, SAN, validity, issuer, profile, and realm information. For questions about multi realm deployments, custom certificate profiles, or integrating discovery output with the CBOM aggregator pipeline, refer to the CBOM platform documentation or contact your CBOM administrator.