Venafi Setup Guide for CBOM PKI Sensor

1. Purpose

The CBOM PKI sensor (`Discover_PKI`) integrates with **Venafi**, the machine identity management platform now part of **CyberArk Machine Identity Security**, to discover and inventory certificates orchestrated across the multiple Certificate Authorities Venafi manages. The sensor supports both deployment models: **Venafi Trust Protection Platform (TPP)**, self hosted on premises, and **Venafi as a Service (VaaS)**, the cloud hosted offering.

During each discovery cycle the sensor queries the Venafi certificate inventory REST API in read only mode and collects the following metadata:

- **Subject DN**, Common Name, Organization, and other distinguished name fields
- **Subject Alternative Names (SANs)**, DNS names, IP addresses
- **Validity period**, Valid From and Valid To timestamps
- **Issuer DN and CA template**, the underlying CA (Microsoft CA, DigiCert, EJBCA, etc.) Venafi issued the certificate through
- **Policy folder / application zone**, the Venafi policy folder (TPP) or zone (VaaS) the certificate is managed under
- **Key algorithm and size**, RSA, ECC, and associated bit/curve length
- **Serial number and Thumbprint (SHA 1)**, unique certificate identifiers
- **Validation state**, whether Venafi's periodic revalidation confirms the certificate still matches the CA's record

This guide covers registering a read only API integration, obtaining OAuth access credentials (TPP) or an API key (VaaS), and configuring the `Discover_PKI` sensor YAML.

---

2. Prerequisites

Before beginning, confirm the following:

1. You have access to the **Venafi TPP Aperture admin console**, or the **Venafi as a Service** tenant admin portal, with permission to create API integrations and service accounts.
2. For TPP: version **22.1 or later**, with the WebSDK REST API enabled. For VaaS: an active tenant with API access enabled for your organization.
3. The CBOM sensor host has **HTTPS network connectivity** to the TPP WebSDK endpoint (default port `443`) or to `api.venafi.cloud` for VaaS.
4. A valid **TLS certificate** is installed on the TPP server, or the CBOM sensor is configured to trust the issuing CA.
5. You have the **base URL** of the TPP instance (e.g. `https://tpp.example.com`), or confirm the VaaS API base URL (`https://api.venafi.cloud`).
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 API Integration

A dedicated integration isolates the CBOM sensor's access from interactive user sessions and limits it to certificate inventory reads.

3.1 TPP: Create an API Integration and Service Account

1. Log in to the **TPP Aperture** admin console using an account with PKI administrator privileges.
2. Navigate to **Support > API Integrations**.
3. Click **New Integration**, and configure: - **Name**: `CBOM Discover PKI` - **Scopes**: `certificate:manage,discover`, grant only the read/discover portion; do not enable `admin` or `revoke` scopes.
4. Save the integration and note the generated **Client ID** (e.g. `cbom-discover-pki`).
5. Under **Identity > Users**, create a dedicated service account `cbom-sensor-svc` with the **PKI Read Only** or an equivalent custom role that grants read access to the certificate tree, and no issuance, approval, or revocation permissions.

3.2 VaaS: Create a Service Account and API Key

1. Log in to the **Venafi as a Service** web portal.
2. Navigate to **Settings > Service Accounts**.
3. Click **New Service Account**, name it `cbom-discover-pki`, and assign it the **Guest** or a custom **read only** role scoped to the applications/zones the sensor should cover.
4. Generate an **API Key** for the service account and copy it immediately, it is shown only once.
5. Store the API key securely in your secrets manager.

> **Note:** Do not assign the TPP `Master Admin`, `PKI Administrator`, or VaaS `System Admin` roles to the sensor's identity. `Discover_PKI` requires no enrollment, approval, revocation, or configuration permissions.

---

4. Generate Access Credentials

4.1 TPP: OAuth2 Password Grant

1. Confirm the API Integration created in Section 3.1 is authorized for the `cbom-sensor-svc` service account under **Support > API Integrations > CBOM Discover PKI > Authorize**.
2. Note the OAuth token endpoint, typically `https://tpp.example.com/vedauth/authorize/oauth`.
3. Record the service account's password and store it in your secrets manager, it is exchanged for short lived access and refresh tokens at runtime, never stored by the sensor itself.

4.2 VaaS: API Key

The API key generated in Section 3.2 is used directly as a request header (`tppl-api-key`) and does not require a token exchange step. Confirm the key is scoped to the correct team/zone before proceeding.

---

5. Verify API Access

Before configuring the sensor, validate that credentials can authenticate and retrieve certificate data.

5.1 TPP: Obtain an OAuth Token and Query Certificates

```
# Step 1: Obtain an access token via the password grant
TOKEN_RESPONSE=$(curl -s -X POST "https://tpp.example.com/vedauth/authorize/oauth" \
  -H "Content-Type: application/json" \
  -d '{
    "client_id": "cbom-discover-pki",
    "username": "cbom-sensor-svc",
    "password": "<password>",
    "scope": "certificate:manage,discover"
  }')

ACCESS_TOKEN=$(echo "$TOKEN_RESPONSE" | python3 -c "import sys,json;
print(json.load(sys.stdin)['access_token'])")

# Step 2: Query the certificate inventory
curl -s -X GET \
  "https://tpp.example.com/vedsdk/certificates/?limit=5" \
  -H "Authorization: Bearer $ACCESS_TOKEN" \
  | python3 -m json.tool
```

> **Note:** TPP access tokens are short lived (typically 4 hours). The response also includes a `refresh_token`, the sensor uses this to obtain new access tokens between discovery cycles without re sending the service account's password. Refresh tokens should still be treated as sensitive and stored in a secrets manager.

5.2 VaaS: Query Certificates with an API Key

```
curl -s -X GET \
  "https://api.venafi.cloud/outagedetection/v1/certificates?limit=5" \
  -H "tppl-api-key: <api-key>" \
  -H "Accept: application/json" \
  | python3 -m json.tool
```

A successful response returns a JSON array of certificate objects. Confirm fields such as `certificateName`, `validTo`, `issuerCA`, and `keyStrength` (TPP) or `subjectCN`, `validityEnd`, and `keySize` (VaaS) 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/venafi_pki.yaml
sensor:
  name: Discover_PKI
  vendor: Venafi
  product: "Venafi Trust Protection Platform"   # or "Venafi as a Service"
  category: PKI
  enabled: true
  schedule: "0 4 * * *"           # Daily at 04:00 local time (cron syntax)
  timeout_seconds: 120

connection:
  platform: "tpp"                 # Options: "tpp", "vaas"
  base_url: "https://tpp.example.com"   # or "https://api.venafi.cloud" for VaaS
  verify_ssl: true
  ca_bundle: "/etc/ssl/certs/internal-root-ca.pem"  # Path to CA bundle; set to true
for system store (VaaS)

auth:
  method: "oauth2_password"        # Options: "oauth2_password" (TPP),
"api_key" (VaaS)

  # --- TPP OAuth2 Password Grant ---
  token_endpoint: "https://tpp.example.com/vedauth/authorize/oauth"
  client_id: "cbom-discover-pki"
  username: "cbom-sensor-svc"
  password: "${VENAFI_PASSWORD}"       # Inject from secrets manager
  scope: "certificate:manage,discover"

  # --- VaaS API Key (uncomment if platform: vaas) ---
  # api_key: "${VENAFI_VAAS_API_KEY}"

discovery:
  endpoint: "/vedsdk/certificates/"    # TPP; VaaS uses "/outagedetection/v1/
certificates"
  page_size: 100                       # Number of certificates per paginated request
  max_pages: 500                       # Safety cap; remove or increase for large
inventories
  include_expired: true
  include_revoked: false               # Revoked certificates excluded from active
CBOM by default
  filters:
    # Optional Venafi query filters (passed as query parameters).
    # Refer to the Venafi WebSDK / VaaS API reference for supported filter fields.
    # policy_dn: "\\VED\\Policy\\CBOM Discovery Scope"
    # zone: "Web Servers\\Production"

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

logging:
```

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

Replace all placeholder values before running the sensor:

Placeholder	Replace With
`tpp.example.com`	Actual TPP hostname (omit for VaaS)
`cbom-discover-pki`	Client ID / service account name created in Section 3
`${VENAFI_PASSWORD}` / `${VENAFI_VAAS_API_KEY}`	Environment variable or secrets manager reference
`/etc/ssl/certs/internal-root-ca.pem`	Path to CA bundle, or `true` for system store
`\\VED\\Policy\\CBOM Discovery Scope`	Policy folder (TPP) or zone (VaaS) to restrict discovery to, if scoping is required

---

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. Set the required environment variable:

```
export VENAFI_PASSWORD="your-actual-password-here"
# or, for VaaS:
export VENAFI_VAAS_API_KEY="your-actual-api-key-here"
```

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

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

1. Confirm the output contains log lines similar to:

```
[INFO] Discover_PKI: Authenticated successfully to https://tpp.example.com
[INFO] Discover_PKI: Page 1 -- retrieved 100 certificates
[INFO] Discover_PKI: Page 2 -- retrieved 100 certificates
...
[INFO] Discover_PKI: Discovery complete. Total certificates found: 1,204
[INFO] Discover_PKI: Output written to /var/cbom/output/venafi_pki_discovery.json
```

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

```
python3 -c "
import json
with open('/var/cbom/output/venafi_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: `invalid_grant` on Token Request (TPP)

Symptom:

```
[ERROR] Discover_PKI: HTTP 400 Bad Request -- POST /vedauth/authorize/oauth
[ERROR] Discover_PKI: Response body:
{"error":"invalid_grant","error_description":"Authentication failed"}
```

Cause: The service account password is incorrect, has expired under a domain password policy, or the API Integration created in Section 3.1 has not been authorized for `cbom-sensor-svc`.

Resolution:

1. Verify the password stored in the environment variable matches the current `cbom-sensor-svc` credential.
2. In Aperture, confirm under **Support > API Integrations > CBOM Discover PKI > Authorize** that the service account is listed as an authorized user of the integration.
3. Re run the curl test from Section 5.1 manually to isolate whether the issue is credential- or authorization related.

Error 2: `403 Forbidden`, Insufficient Scope

Symptom:

```
[ERROR] Discover_PKI: HTTP 403 Forbidden -- GET /vedsdk/certificates/
[ERROR] Discover_PKI: Response body: {"Error":"Insufficient permissions to access
this resource"}
```

Cause: The service account can authenticate, but its role does not grant read access to the policy folder(s) or zone(s) being queried, or the OAuth token's scope does not include `discover`.

Resolution:

1. Confirm the token request in Section 4.1 includes `discover` in the requested `scope`.
2. In Aperture, check the service account's role assignment under **Identity > Users > cbom sensor svc > Roles** and confirm it has read access to the target policy folder tree.

3. For VaaS, confirm the service account's role includes read access to the relevant application/zone under **Settings > Service Accounts > cbom discover pki**.

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 TPP server presents a TLS certificate issued by an internal CA that is not trusted by the sensor host's system certificate store. (VaaS uses a publicly trusted certificate and should not require this fix.)

Resolution:

1. Export the root CA certificate from your PKI in PEM format.
2. Copy it to the sensor host, for example `/etc/ssl/certs/internal-root-ca.pem`.
3. Update the YAML configuration to reference the bundle:

```
connection:
  verify_ssl: true
  ca_bundle: "/etc/ssl/certs/internal-root-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 OAuth2 tokens (TPP) or scoped API keys (VaaS) over long lived static passwords.** Rely on the refresh token flow for TPP so the service account password is only ever exchanged once at initial setup, not on every discovery cycle.
- **Rotate the API key (VaaS) or service account password (TPP) on a regular schedule.** Automate rotation through your secrets manager rather than manual updates to the YAML configuration.
- **Restrict the service account and API Integration to the minimum required scope.** `certificate:manage,discover` (TPP) or a read only zone role (VaaS) is sufficient for `Discover_PKI`. Never assign enrollment, revocation, or administrative roles to the sensor's identity.
- **Store all secrets in an external secrets manager** (e.g. HashiCorp Vault, AWS Secrets Manager, Azure Key Vault, or CyberArk's own vault) and inject them as environment variables at runtime. Never write plaintext credentials in the YAML configuration file or commit them to source control.
- **Review Venafi's own audit trail** for the `cbom-sensor-svc` identity (TPP: **Reports > Audit Log**; VaaS: **Activity Log**) and confirm it performs only certificate read/discover operations. Any

unexpected enrollment, approval, or revocation activity associated with this identity should be investigated immediately.

---

10. Conclusion

This guide has walked through registering a dedicated, read only Venafi API integration and service account, obtaining OAuth2 access credentials (TPP) or an API key (VaaS), and configuring the `Discover_PKI` sensor YAML to query the Venafi certificate inventory across either deployment model. With scheduled execution, the CBOM sensor maintains a continuously updated inventory of certificate metadata including subject, SAN, validity, issuer CA, policy folder/zone, and key details. For questions about multi zone scoping, VaaS team boundaries, or integrating discovery output with the CBOM aggregator pipeline, refer to the CBOM platform documentation or contact your CBOM administrator.