Azure Key Vault & Container Registry

CBOM Secure · Integration Guide

Overview

Crypto Bill of Materials (CBOM) involves discovering, inventorying, and assessing cryptographic assets — keys, certificates, algorithms, and their usage — across an organization's infrastructure. When extended to Azure, CBOM Secure covers two complementary components:

- Azure Key Vault — discovered directly through Azure APIs/SDKs, optionally enriched with Azure Key Sensor telemetry for usage and risk context.
- Azure Container Registry (ACR) — container images that may embed certificates and keys, scanned at the image-layer level.

Prerequisites

- An Azure subscription with read access to the target Key Vaults and Container Registries.
- A discovery identity (service principal or managed identity) with Key Vault Get/List on keys, secrets, and certificates, and AcrPull on the target registries.
- Azure CLI and, for image inspection, a container runtime (for example Docker) on the scanning host.
- (Optional) A Log Analytics workspace with Microsoft Defender for Cloud / Azure Key Sensor telemetry enabled, if usage and risk correlation is required.
- The CBOM Secure Azure Discovery Sensor package ready for configuration.

Step-by-Step Guide

Step 1: Grant the Discovery Identity Access

Grant the discovery identity read access to both Key Vault and the container registry.

```
# Key Vault data-plane read
az keyvault set-policy \
  --name <kv-name> \
  --spn <client-id> \
  --key-permissions get list \
  --secret-permissions get list \
  --certificate-permissions get list

# Container registry pull access
az role assignment create \
  --assignee <client-id> \
  --role AcrPull \
  --scope $(az acr show --name <acr-name> --query id -o tsv)
```

Step 2: Discover Azure Key Vault Material

CBOM Secure enumerates the following from each Key Vault, using the Azure SDK / REST APIs as the active source of truth:

Asset	What is discovered
Keys	RSA, EC, and symmetric (e.g. AES) keys — HSM-backed or software.
Secrets	PEM / PKCS#12-encoded certificates or keys stored as Base64 blobs.
Certificates	X.509 objects, sometimes bundled with the private key.
Metadata	Key usage, creation and expiry dates, access policies, and tags.

Step 3: (Optional) Enrich with Azure Key Sensor Telemetry

Azure Key Sensor does not expose key or certificate material — it is a threat-detection system. For CBOM purposes it provides passive context rather than discovery:

- Access telemetry — who accessed which key and when, supporting usage analysis.
- Alert correlation — flags misused or at-risk cryptographic material.
- Context — links key/certificate usage to potential compromise or misconfiguration.

Note Best practice: use Azure REST APIs / SDKs for active discovery of material, and Key Sensor logs (via Defender for Cloud / Log Analytics) for passive telemetry.

Step 4: Discover Cryptographic Material in ACR Images

ACR does not store keys or certificates directly, but the container images it holds frequently embed cryptographic material. CBOM Secure inspects image layers for:

- Embedded TLS certificates and private keys in container layers (for example under /etc/ssl/ or application configs).
- Hard-coded secrets and keys in application files.
- SBOMs pushed as OCI artifacts, which may declare libraries with cryptographic dependencies.
- Code-signing metadata, where images are signed (for example with Cosign).

How discovery works

- Enumerate repositories and image digests with the ACR APIs or the az acr CLI.
- Pull and extract image layers to inspect files for .pem, .key, .crt, .pfx, and .jks material.
- For OCI artifacts or Sigstore (Cosign / Notary v2), parse signatures, annotations, and manifest metadata for cryptographic usage.

```
# Enumerate image manifests in a repository
az acr repository show-manifests \
 --name <acr-name> \
 --repository <repository>
```

Step 5: Configure the CBOM Secure Sensor

```
tenant_id: "<TENANT_ID>"
client_id: "<CLIENT_ID>"
client_secret: "<CLIENT_SECRET>"
```

```
subscription_id: "<SUBSCRIPTION_ID>"
targets:
  - type: key_vault
    name: "<kv-name>"
  - type: container_registry
    name: "<acr-name>"
scan_name: AzureKeyVaultACRDiscovery
```

Step 6: Validate

Run the discovery task and confirm Key Vault keys, secrets, and certificates are inventoried, and that any cryptographic material embedded in scanned ACR images is reported. A representative CBOM record:

```
{
  "discoverID": "cbom-azure-123",
  "type": "certificate",
  "location": "AzureKeyVault://kv-name/cert1",
  "publicKeySha256": "...",
  "createdAt": "2023-11-01T08:00:00",
  "expiresAt": "2025-11-01T08:00:00",
  "keyUsage": ["digitalSignature", "keyEncipherment"],
  "origin": "AzureKeyVault",
  "alerts": ["SuspiciousAccessPattern", "GeoAnomaly"]
}
```

Reference: Discovery Methods at a Glance

Component	Discovery method	Risk signals (Key Sensor / Defender)
Azure Key Vault	Azure SDK / API / CLI / REST	Brute-force alerts, anomalous access
Azure Key Sensor	Telemetry via Defender for Cloud	Suspicious access, privilege escalation
ACR images	Scan image layers or SBOMs	Hard-coded secrets, weak crypto
ACR signatures	Parse Notary / Cosign metadata	Signature mismatch, untrusted issuer