GnuPG Setup

CBOM Secure · PGP Sensor (Discover_PGP)

Overview

The CBOM Secure Discover_PGP sensor performs read-only discovery of OpenPGP key material in GnuPG keyring files on a host, inventorying metadata only:

- Public key metadata: key ID, fingerprint, algorithm, key size, creation/expiry dates, and UIDs
- Secret key stubs: presence of secret material, keygrip, and subkeys
- Subkey relationships: fingerprints, algorithms, usage flags, and validity periods

Private key material, passphrases, and raw key bytes are never read, transmitted, or stored.

Prerequisites

- GnuPG installed (2.1+ recommended; 1.x supported read-only for .gpg keyrings).
- The CBOM agent installed on the host (or a mounted path to the GPG home).
- The agent OS user has read permission on ~/.gnupg/ or $GNUPGHOME.
- Python 3.8+ for the CBOM agent runtime.

Step-by-Step Guide

Step 1: Identify the GnuPG Home Directory

```
gpg --version
echo "GNUPGHOME=${GNUPGHOME:-~/.gnupg}"
ls -lh "${GNUPGHOME:-$HOME/.gnupg}/"
```

Modern GnuPG 2.1+ uses pubring.kbx (plus private-keys-v1.d/); legacy uses pubring.gpg / secring.gpg.

Step 2: Verify Read Access and List Keys

As the CBOM agent OS user, confirm keyring metadata is readable (the sensor uses --batch --no-tty):

```
gpg --homedir "${GNUPGHOME:-$HOME/.gnupg}" \
    --list-keys --with-fingerprint --with-keygrip --batch --no-tty
gpg --homedir "${GNUPGHOME:-$HOME/.gnupg}" \
    --list-secret-keys --with-fingerprint --with-keygrip --batch --no-tty
```

Note A # after sec/ssb (e.g. sec#) means the private key is a stub (not locally present). The sensor records only metadata.

Step 3: Grant Read Permission (if required)

```
sudo usermod -aG <keyring_owner_group> cbom-agent
chmod 750 /home/appuser/.gnupg
```

```
chmod 640 /home/appuser/.gnupg/pubring.kbx
```

Do not make .gnupg world-readable — that exposes trust data to all local users.

Step 4: Configure the CBOM Secure Sensor

```
sensors:
  - name: Discover_PGP
    enabled: true
    schedule: "0 2 * * *"
    config:
      gpg_home: "/home/appuser/.gnupg"
      gpg_binary: "gpg"
      discover_secret_key_stubs: true
      discover_subkeys: true
      stale_key_threshold_days: 365
      tags: ["gnupg", "pgp", "env:production"]
      cbom_api_endpoint: "https://cbom.example.com/api/v1"
      cbom_api_token: "${CBOM_API_TOKEN}"
sudo systemctl restart cbom-agent
```

Step 5: Validate

```
sudo journalctl -u cbom-agent -n 100 --no-pager | grep -i "Discover_PGP"
cbom-agent run-sensor --name Discover_PGP --once
```

Confirm public keys and secret-key stubs are listed and that key counts match a manual gpg --list-keys. Verify fingerprints, algorithms, sizes, and dates under Assets → PGP Keys, with no private material present.

Common Errors

Permission denied opening keyring file

Cause: The agent OS user cannot read the keyring file.

Resolution: chmod 640 the keyring, set group ownership to a shared group, and add cbom-agent to it.

Unsafe permissions on homedir (gpg aborts)

Cause: GnuPG refuses a group/world-writable home directory (e.g. 777/775).

Resolution: chmod 700 the .gnupg directory and 600 on pubring.kbx / trustdb.gpg.

CBOM platform rejects report (HTTP 401)

Cause: The API token is missing/expired or the variable was not injected.

Resolution: Set CBOM_API_TOKEN in the systemd unit, reload/restart, and rotate the token if expired.

Security Recommendations

- Grant the agent OS user read-only access to the GPG home; never run it as the keyring owner or root.

- Never make .gnupg broader than 750.
- Inject the API token via a systemd EnvironmentFile (chmod 600), not the YAML.
- Alert on keys approaching expiry via stale_key_threshold_days.
- Use auditd on the .gnupg directory to detect access outside scan windows.

Conclusion

The Discover_PGP sensor delivers automated, read-only discovery of OpenPGP key metadata from GnuPG keyrings — giving continuous visibility into PGP key lifecycle, algorithms, and expiry without ever reading private material.