Bitbucket Setup

CBOM Secure · Source Code Sensor (Discover_SourceCode)

Overview

The CBOM Secure Discover_SourceCode sensor scans Bitbucket repository contents to discover cryptographic material embedded in source code and configuration files:

- Hard-coded private keys, public keys, and key pairs (RSA, EC, DSA, Ed25519)
- X.509 certificates and chains stored in repository files
- API tokens, secret keys, and access credentials committed to repositories
- Symmetric keys, passphrases, and initialization vectors
- SSH keys and known-hosts entries
- Cryptographic artifacts such as .pem, .crt, .key, .jks, and .p12 files

Covers both Bitbucket Cloud and Bitbucket Server / Data Center. All access is read-only.

Prerequisites

- A Bitbucket Cloud workspace admin account, or a Bitbucket Server / Data Center 7.x+ user with Project Read and Repository Read.
- Connectivity from the sensor host to the API endpoint (api.bitbucket.org for Cloud; your host for Server).
- The Discover_SourceCode sensor binary on the sensor host.
- Outbound HTTPS (443) allowed to the Bitbucket API endpoint.
- The workspace slug (Cloud) or project key (Server) for the target repositories.
- Python 3.8+ or the CBOM sensor runtime dependencies installed.

Step-by-Step Guide

Step 1 (Cloud): Create an App Password

Log in to Bitbucket Cloud → user avatar → Personal settings → App passwords → Create app password. Label it CBOM-SourceCode-Sensor, enable only Repositories: Read, and Create. Copy the password immediately (shown once) and note the account username (not email).

Verify Cloud API access

```
curl -u "your_username:your_app_password" \
  "https://api.bitbucket.org/2.0/repositories/{workspace}"
```

An HTTP 200 with a values array confirms connectivity and credential validity.

Step 2 (Server/Data Center): Create an HTTP Access Token

Log in → user avatar → Manage account → HTTP access tokens → Create token. Name it CBOM-SourceCode-Sensor, set permission level Project read (Repository read is included), leave Write/Admin unselected, optionally set an expiry, and Create. Copy the token immediately.

Verify Server API access

```
curl -H "Authorization: Bearer your_access_token" \
  "https://<your-bitbucket-host>/rest/api/1.0/projects/{projectKey}/repos"
```

Step 3: Configure the Sensor (Cloud)

```
sensor:
  name: Discover_SourceCode
  enabled: true
  provider: bitbucket_cloud
connection:
  base_url: "https://api.bitbucket.org/2.0"
  workspace: "your-workspace-slug"
  auth:
    type: basic
    username: "your_bitbucket_username"
    password: "your_app_password"
scan:
  repositories: ["*"]              # or specific repo slugs
  branches: ["main", "master", "develop"]
  file_extensions: [".pem", ".crt", ".cer", ".key", ".p12", ".jks",
    ".env", ".yaml", ".yml", ".json", ".xml", ".conf", ".properties"]
  exclude_paths: ["node_modules/", ".git/", "vendor/", "dist/"]
  max_file_size_kb: 512
  depth: full
output:
  format: cbom_json
  destination: "/var/cbom/output/bitbucket_cloud"
schedule:
  interval: "24h"
```

Step 4: Configure the Sensor (Server / Data Center)

```
sensor:
  name: Discover_SourceCode
  enabled: true
  provider: bitbucket_server
connection:
  base_url: "https://your-bitbucket-host.example.com"
  auth:
    type: bearer_token
    token: "your_http_access_token"
  ssl_verify: true
  ssl_ca_bundle: "/etc/ssl/certs/ca-certificates.crt"    # custom CA if needed
scan:
  projects: ["PROJ"]               # your project key(s)
  repositories: ["*"]
  branches: ["main", "master", "develop"]
  file_extensions: [".pem", ".crt", ".cer", ".key", ".p12", ".jks",
    ".env", ".yaml", ".yml", ".json", ".xml", ".conf", ".properties"]
  exclude_paths: ["node_modules/", ".git/", "vendor/", "dist/"]
  max_file_size_kb: 512
output:
  format: cbom_json
  destination: "/var/cbom/output/bitbucket_server"
```

Step 5: Validate

```
cbom-sensor run --config /etc/cbom/sensors/bitbucket-setup.yaml
ls -lh /var/cbom/output/bitbucket_cloud/
```

Confirm Authenticated to Bitbucket successfully and a discovered-repository count, then inspect the generated CBOM JSON report. For Cloud, watch the X-RateLimit-Remaining header and adjust the schedule if needed.

Common Errors

HTTP 401 Unauthorized

Cause: Wrong/revoked credential, or (Cloud) an email used instead of the username.

Resolution: Use the account username (not email), regenerate the app password/token if revoked, and strip whitespace from the config values.

HTTP 403 Forbidden — insufficient permissions

Cause: The credential lacks Repository Read, or the account cannot access the target workspace/project.

Resolution: Recreate the app password with Repositories: Read (Cloud) or confirm Project/Repository read (Server); verify the workspace/projects values.

SSL CERTIFICATE_VERIFY_FAILED

Cause: The Server instance uses an internal CA not in the host trust store.

Resolution: Set ssl_ca_bundle to the internal CA, or append it to the system bundle and run update-ca-certificates. Never set ssl_verify: false in production.

Security Recommendations

- Use a dedicated Bitbucket service account for the sensor — not a personal account.
- Grant only Read permissions; never Write, Admin, or Webhook.
- Rotate app passwords / tokens on schedule (typically every 90 days); use token expiry on Server.
- Store credentials in a secrets manager and reference by path — never plaintext YAML.
- Restrict the sensor host's outbound access to only the Bitbucket API endpoint.

Conclusion

With read-only API credentials in Bitbucket Cloud or Server/Data Center and least-privilege configuration, the Discover_SourceCode sensor safely and continuously inventories cryptographic material across your repositories, feeding results into CBOM Secure for centralized analysis and compliance reporting.