Red Hat Trusted Application Pipeline 1.0

Inspecting your SBOM using Red Hat Trusted Profile Analyzer

Learn how to scan your SBOM to gain actionable information about the security posture of your application.

Last Updated: 2024-07-01

Red Hat Trusted Application Pipeline 1.0 Inspecting your SBOM using Red Hat Trusted Profile Analyzer

Learn how to scan your SBOM to gain actionable information about the security posture of your application.

Legal Notice

Abstract

This document provides information about how to review SBOM to gain actionable information about the security posture of your application.

Table of Contents

PREFACE

When Red Hat Trusted Application Pipeline builds your application images, it also provides a software bill of materials (SBOM). The SBOM lists all the software libraries that the image uses. You can use the SBOM to identify security vulnerabilities.

However, the SBOM is long and difficult to read. To turn the raw SBOM into actionable information, you can use Trusted Profile Analyzer (TPA). For example, TPA can identify dependencies in your image that are targets of known Common Vulnerabilities and Exploits (CVEs).

CHAPTER 1. DOWNLOADING, CONVERTING, AND ANALYZING YOUR SBOM

The following procedure explains how to inspect your SBOM with TPA. Specifically, it outlines how to download an SBOM, convert the SBOM into a compatible format, and analyze the SBOM with TPA.

Prerequisites:

- Cosign
- Syft
- jq

Procedure:

1. In your container registry, find the full address of the container image whose SBOM you want to inspect. The address has the format registry/namespace/image:tag. For example, quay.io/app/app-image:ff59e21cc...

NOTE

Do not use the address of the SBOM image, which ends with **.sbom**. Use the address of the image for the actual application.

2. In your CLI, use cosign to download the SBOM. Redirect the output to a file you can reference later. Make sure the new filename ends with **.json**.

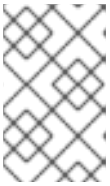
```
cosign download sbom quay.io/redhat/rhtap-
app:8d34c03188cf294a77339b2a733b1f6811263a369b309e6b170d9b489abc0334 >
/tmp/sbom.json
```

3. (Optional) Your SBOM ultimately appears in the TPA UI with a name listed in this .json file. By default, Syft creates that name based on the filepath of the SBOM. If you want your SBOM to appear in the TPA UI with a more meaningful name, you must manually change it in the .json file you just downloaded. Specifically, you must replace the name in the **.metadata.component** object. You can optionally add a **version** field here, if you wish.

```
$ vim /tmp/sbom.json
"component": {
     "bom-ref": "fdef64df97f1d419",
     "type": "file",
     "name":
"/var/lib/containers/storage/vfs/dir/3b3009adcd335d2b3902c5a7014d22b2beb6392b1958f1d9c
7aabe24acab2deb" #Replace this with a meaningful name
   }
```

4. Run the following command to store the Bombastic API URL as an environment variable.

```
$ bombastic_api_url="https://$(oc -n rhtap get route --selector
app.kubernetes.io/name=bombastic-api -o jsonpath='{.items[].spec.host}')"
```



NOTE

In this command and the next command, after **-n**, be sure to enter the namespace in which you installed RHTAP. The examples assume you used a namespace called **rhtap**.

5. In your CLI, create a new **token_issuer_url** environment variable with the following value.

```
$ token_issuer_url=https://$(oc -n rhtap get route --selector
app.kubernetes.io/name=keycloak -o
jsonpath='{.items[].spec.host}')/realms/chicken/protocol/openid-connect/token
```

6. Next, you need to set the TPA__OIDC__WALKER_CLIENT_SECRET environment variable. If you have access to the private.env file, which your organization generated while installing RHTAP, you can simply source that file. If you do not have access to that file, ask whomever installed RHTAP to provide your with the TPA OIDC Walker client secret.

   a. If you have access to the **private.env** file:

   ```
   $ source private.env
   ```

   b. Or, once you have obtained the secret from whomever installed RHTAP:

   ```
   $ TPA__OIDC__WALKER_CLIENT_SECRET=<secret value>
   ```

7. Run the following command to obtain a token for the BOMbastic API. The token allows you to upload the SBOM.

```
$ tpa_token=$(curl \
    -d 'client_id=walker' \
    -d "client_secret=$TPA__OIDC__WALKER_CLIENT_SECRET" \
    -d 'grant_type=client_credentials' \
    "$token_issuer_url" \
| jq -r .access_token)
```

8. Try to upload the SBOM.

```
curl \
    -H "authorization: Bearer $tpa_token" \
    -H "transfer-encoding: chunked" \
    -H "content-type: application/json" \
    --data @/tmp/sbom.json \
    "$bombastic_api_url/api/v1/sbom?id=my-sbom"
```

   a. If you receive the error message **storage error: invalid storage content**, use Syft to convert your SBOM to an earlier CycloneDX, 1.4. You can disregard warnings about merging packages with different pURLs; they indicate that Syft might discard some data from the original SBOM, but that data is not crucial.

   ```
   $ syft convert /tmp/sbom.json -o cyclonedx-json@1.4=/tmp/sbom-1-4.json
   ```

   b. Then try to upload the SBOM again:

```
$ curl \
    -H "authorization: Bearer $tpa_token" \
    -H "transfer-encoding: chunked" \
    -H "content-type: application/json" \
    --data @/tmp/sbom-1-4.json \
    "$bombastic_api_url/api/v1/sbom?id=my-sbom"
```

9. Access your cluster that is running RHTAP through the OpenShift Console.
10. In the rhtap project, navigate to Networking > Routes. Open the URL listed on the same row as the **spog-ui** service.
11. Use the Register button to create a new account and authenticate to TPA.
12. Select your SBOM (the most recent upload) and see what insights TPA has provided about your application based on that SBOM.
    a. Go to the Dependency Analytics Report tab to view vulnerabilities and remediations.

Additional resources

- Parts of this document are based on the Trustification documentation for SBOMs.

Revised on 2024-07-01 13:32:01 UTC