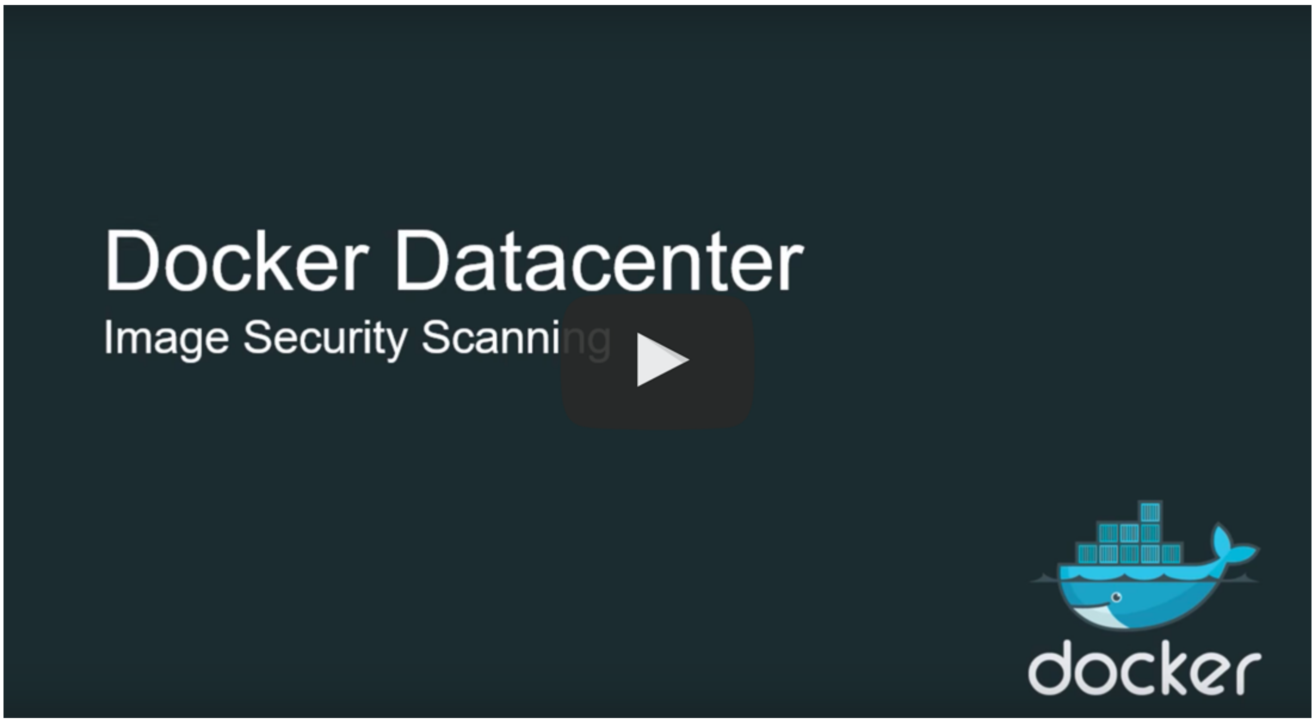
Scan images for vulnerabilities

Estimated reading time: 6 minutes

These are the docs for DTR version 2.3.11

To select a different version, use the selector below.

2.3.11



(https://www.youtube.com/watch?v=121poCB0Nn8)

Docker Trusted Registry can scan images in your repositories to verify that they are free from known security vulnerabilities or exposures, using Docker Security Scanning. The results of these scans are reported for each image tag.

Docker Security Scanning is available as an add-on to Docker Trusted Registry, and an administrator configures it for your DTR instance. If you do not see security scan results available on your repositories, your organization may not have purchased the Security Scanning feature or it may be disabled. See Set up Security Scanning in DTR (https://docs.docker.com/datacenter/dtr/2.3/guides/admin/configure/set-up-vulnerability-scans/) for more details.

> **Tip:** Only users with write access to a repository can manually start a scan. Users with read-only access can view the scan results, but cannot start a new scan.

The Docker Security Scan process

Scans run either on demand when a user clicks the **Start a Scan** links or **Scan** button (see Manual scanning (/datacenter/dtr/2.3/guides/user/manage-images/scan-images-for-vulnerabilities/#manual-scanning) below), or automatically on any `docker push` to the repository.

First the scanner performs a binary scan on each layer of the image, identifies the software components in each layer, and indexes the SHA of each component in a bill-of-materials. A binary scan evaluates the components on a bit-by-bit level, so vulnerable components are discovered even if they are statically-linked or under a different name.

The scan then compares the SHA of each component against the US National Vulnerability Database that is installed on your DTR instance. When this database is updated, DTR reviews the indexed components for newly discovered vulnerabilities.

DTR scans both Linux and Windows images, but by default Docker doesn't push foreign image layers for Windows images so DTR can't scan them. If you want DTR to scan your Windows images, configure Docker to always push image layers (https://docs.docker.com/datacenter/dtr/2.3/guides/user/manage-images/pull-and-push-images/), and it will scan the non-foreign layers.

Security scan on push

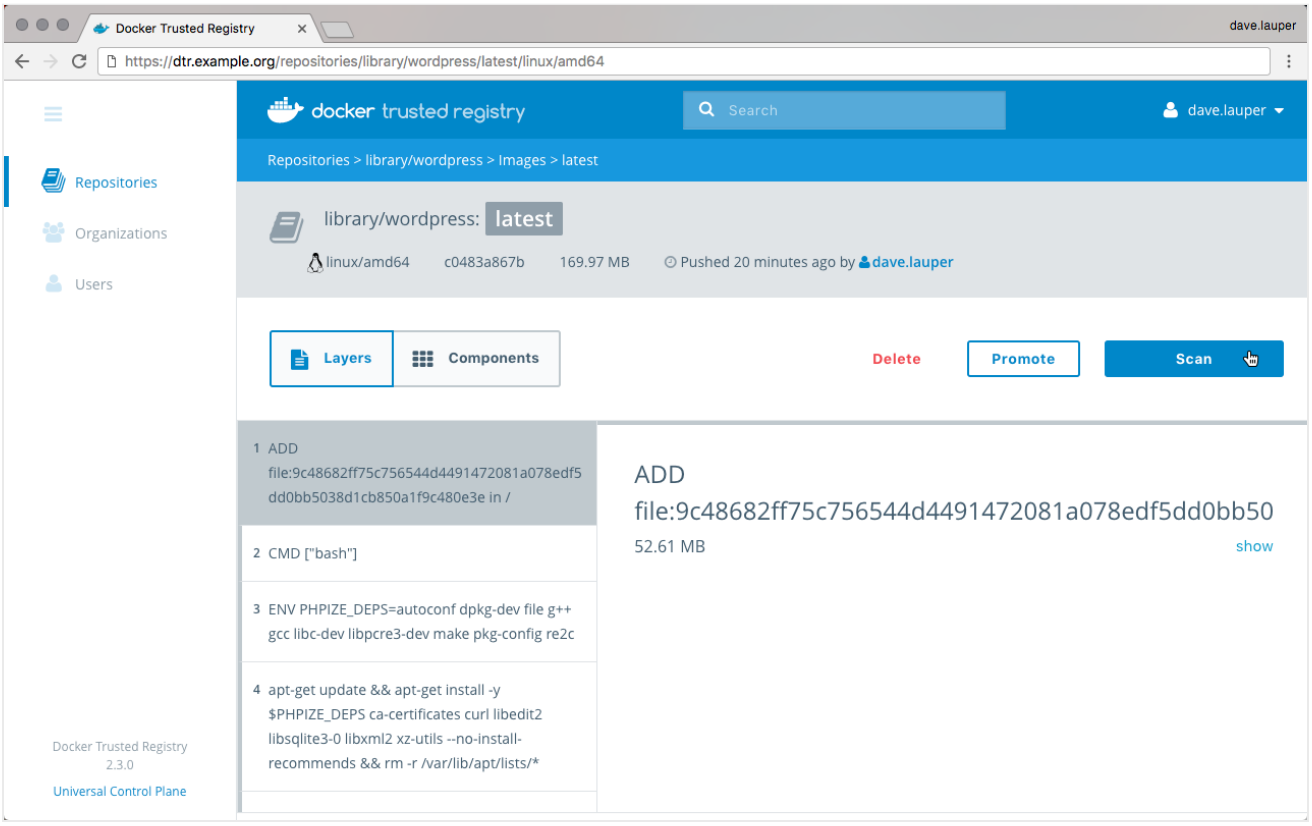
By default, Docker Security Scanning runs automatically on `docker push` to an image repository.

If your DTR instance is configured in this way, you do not need to do anything once your `docker push` completes. The scan runs automatically, and the results are reported in the repository's **Images** tab after the scan finishes.

Manual scanning

If your repository owner enabled Docker Security Scanning but disabled automatic scanning, you can manually start a scan for images in repositories to which you have `write` access.

To start a security scan, navigate to the **tag details**, and click the **Scan** button.


Docker Trusted Registry
dave.lauper
https://dtr.example.org/repositories/library/wordpress/latest/linux/amd64
docker trusted registry
Search
dave.lauper
Repositories
Organizations
Users
Repositories > library/wordpress > Images > latest
library/wordpress: latest
linux/amd64
c0483a867b
169.97 MB
Pushed 20 minutes ago by dave.lauper
Layers
Components
Delete
Promote
Scan
Docker Trusted Registry 2.3.0
Universal Control Plane
show

```
1 ADD file:9c48682ff75c756544d4491472081a078edf5dd0bb5038d1cb850a1f9c480e3e in /
2 CMD ["bash"]
3 ENV PHPIZE_DEPS=autoconf dpkg-dev file g++ gcc libc-dev libpcre3-dev make pkg-config re2c
4 apt-get update && apt-get install -y $PHPIZE_DEPS ca-certificates curl libedit2 libsqlite3-0 libxml2 xz-utils --no-install-recommends && rm -r /var/lib/apt/lists/*
```

```
ADD
file:9c48682ff75c756544d4491472081a078edf5dd0bb50
52.61 MB
```

DTR begins the scanning process. You will need to refresh the page to see the results once the scan is complete.

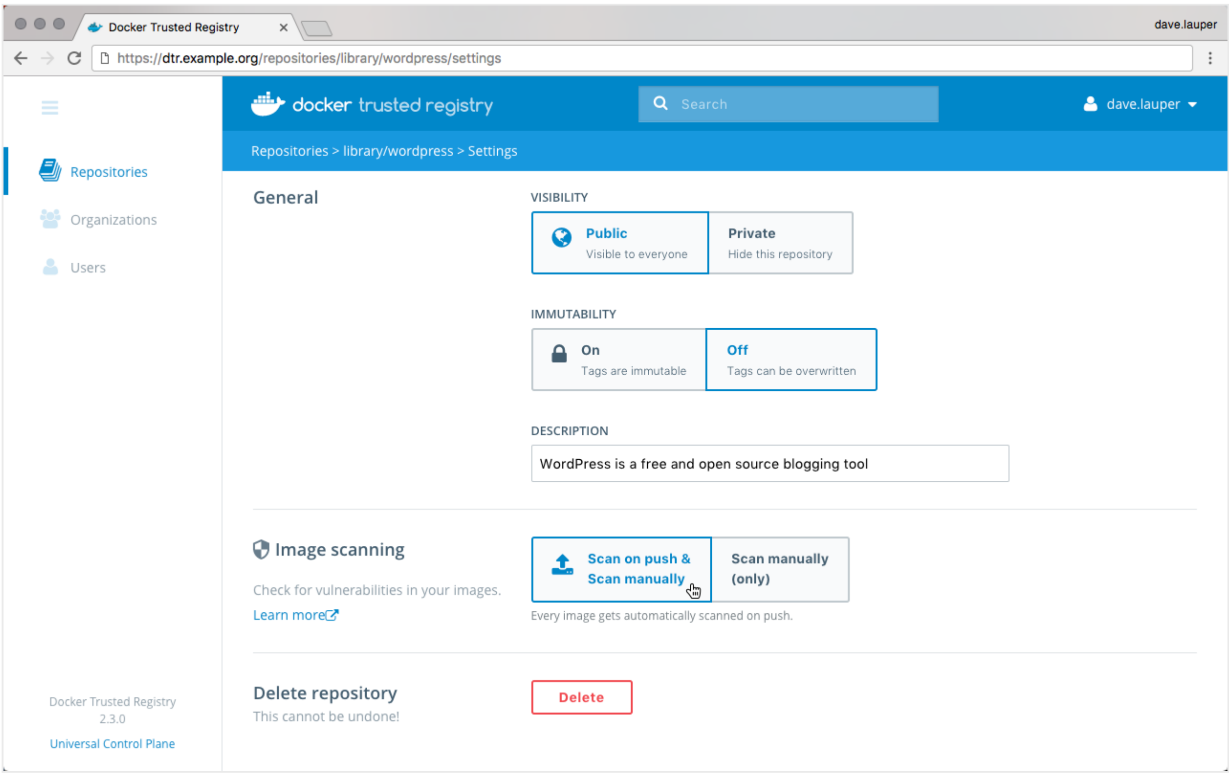
Change the scanning mode

You can change the scanning mode for each individual repository at any time. You might want to disable scanning if you are pushing an image repeatedly during troubleshooting and don't want to waste resources scanning and re-scanning, or if a repository contains legacy code that is not used or updated frequently.

> **Note**: To change an individual repository's scanning mode, you must have `write` or `administrator` access to the repo.

To change the repository scanning mode:

1. Navigate to the repository, and click the **Settings** tab.
2. Scroll down to the **Image scanning** section.
3. Select the desired scanning mode.


Docker Trusted Registry
https://dtr.example.org/repositories/library/wordpress/settings
docker trusted registry
Search
dave.lauper
Repositories
Organizations
Users
Repositories > library/wordpress > Settings

General

VISIBILITY

[✓ Public Visible to everyone] Private Hide this repository

IMMUTABILITY

On Tags are immutable [✓ Off Tags can be overwritten]

DESCRIPTION

WordPress is a free and open source blogging tool

Image scanning

Check for vulnerabilities in your images.
Learn more

[✓ Scan on push & Scan manually] Scan manually (only)

Every image gets automatically scanned on push.

Delete repository

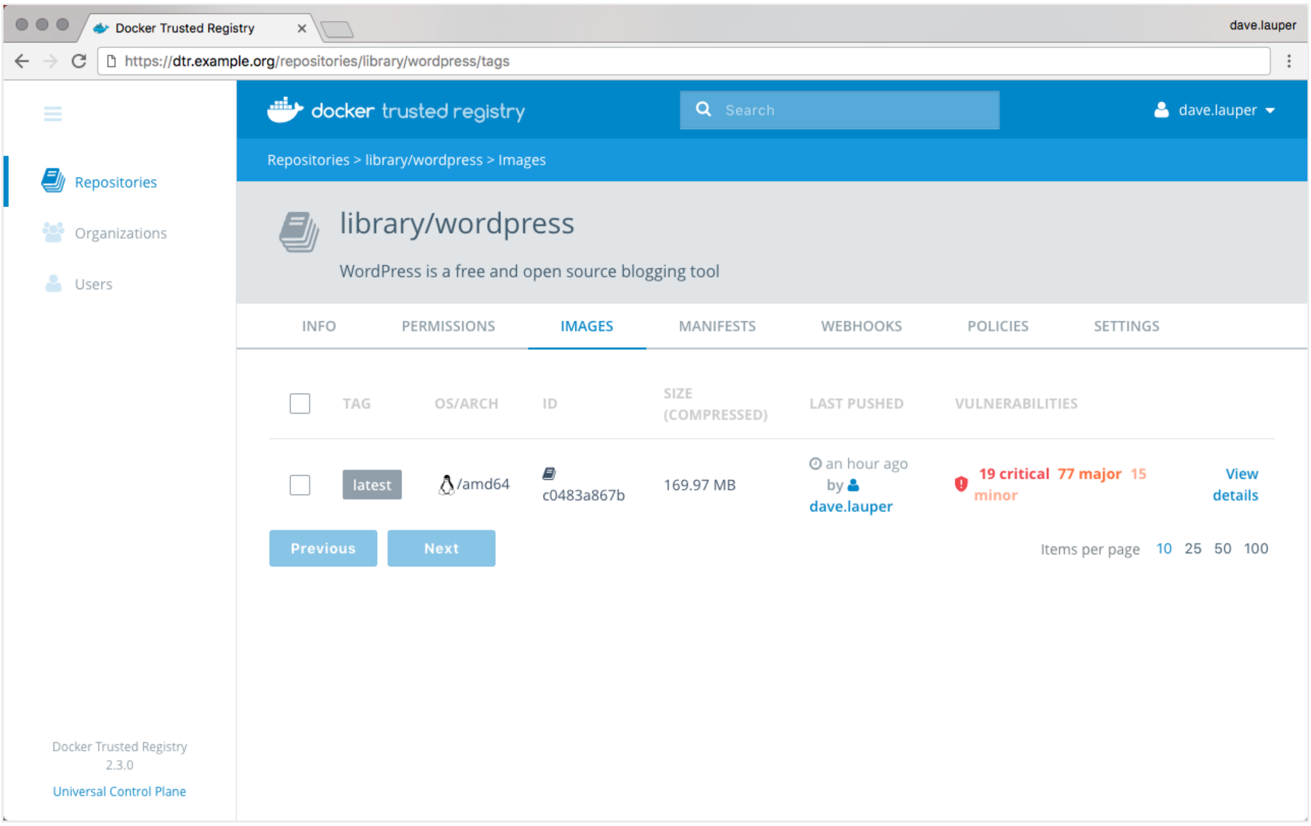
This cannot be undone!

Delete

View security scan results

Once DTR has run a security scan for an image, you can view the results.

The **Images** tab for each repository includes a summary of the most recent scan results for each image.


Docker Trusted Registry
https://dtr.example.org/repositories/library/wordpress/tags
docker trusted registry
Search
dave.lauper
Repositories
Organizations
Users
Repositories > library/wordpress > Images

library/wordpress

WordPress is a free and open source blogging tool

TAG	OS/ARCH	ID	SIZE (COMPRESSED)	LAST PUSHED	VULNERABILITIES	
latest	/amd64	c0483a867b	169.97 MB	an hour ago by dave.lauper	19 critical 77 major 15 minor	View details

- A green shield icon with a check mark indicates that the scan did not find any vulnerabilities.
- A red or orange shield icon indicates that vulnerabilities were found, and the number of vulnerabilities is included on that same line.

If the vulnerability scan can’t detect the version of a component, it reports the vulnerabilities for all versions of that component.

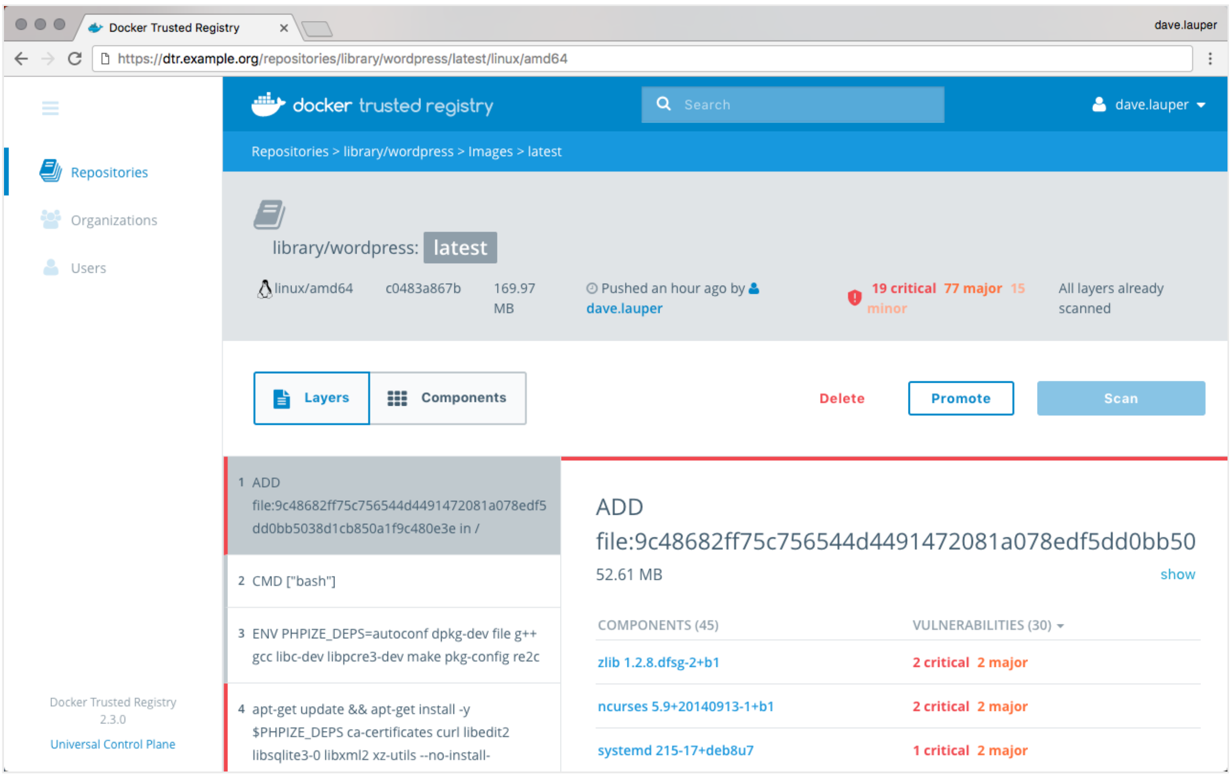
From the **Images** tab you can click **View details** for a specific tag to see the full scan results. The top of the page also includes metadata about the image, including the SHA, image size, date last pushed and user who last pushed, the security scan summary, and the security scan progress.

The scan results for each image include two different modes so you can quickly view details about the image, its components, and any vulnerabilities found.

- The **Layers** view lists the layers of the image in order as they are built by the Dockerfile.

  This view can help you find exactly which command in the build introduced the vulnerabilities, and which components are associated with that single command. Click a layer to see a summary of its components. You can then click on a component to switch to the Component view and get more details about the specific item.

  > **Tip**: The layers view can be long, so be sure to scroll down if you don’t immediately see the reported vulnerabilities.

  Repositories > library/wordpress > Images > latest

  library/wordpress: latest

  linux/amd64 c0483a867b 169.97 MB Pushed an hour ago by dave.lauper 19 critical 77 major 15 minor All layers already scanned

  1 ADD file:9c48682ff75c756544d4491472081a078edf5dd0bb5038d1cb850a1f9c480e3e in /

  2 CMD ["bash"]

  3 ENV PHPIZE_DEPS=autoconf dpkg-dev file g++ gcc libc-dev libpcre3-dev make pkg-config re2c

  4 apt-get update && apt-get install -y $PHPIZE_DEPS ca-certificates curl libedit2 libsqlite3-0 libxml2 xz-utils --no-install-

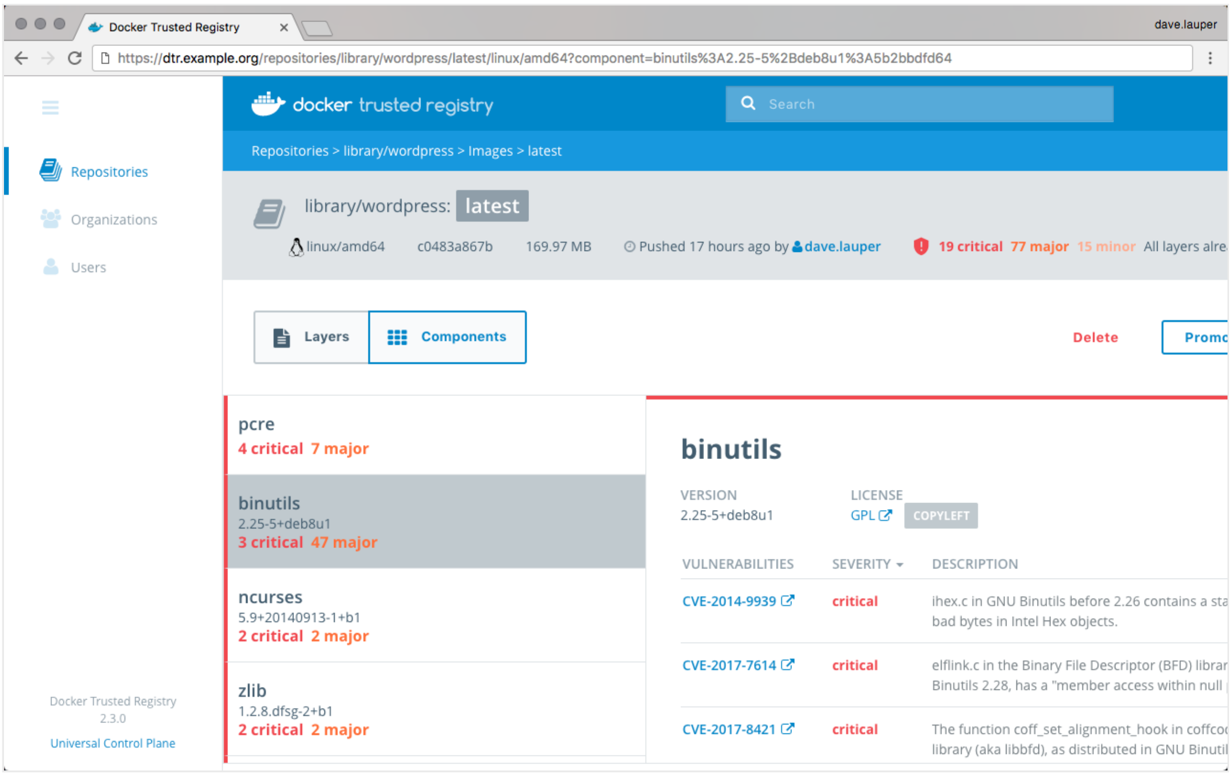
  ADD file:9c48682ff75c756544d4491472081a078edf5dd0bb50

  52.61 MB

COMPONENTS (45)	VULNERABILITIES (30)
zlib 1.2.8.dfsg-2+b1	2 critical 2 major
ncurses 5.9+20140913-1+b1	2 critical 2 major
systemd 215-17+deb8u7	1 critical 2 major

- The **Components** view lists the individual component libraries indexed by the scanning system, in order of severity and number of vulnerabilities found, most vulnerable first.

  Click on an individual component to view details about the vulnerability it introduces, including a short summary and a link to the official CVE database report. A single component can have multiple vulnerabilities, and the scan report provides details on each one. The component details also include the license type used by the component, and the filepath to the component in the image.


Docker Trusted Registry

https://dtr.example.org/repositories/library/wordpress/latest/linux/amd64?component=binutils%3A2.25-5%2Bdeb8u1%3A5b2bbdfd64

dave.lauper

docker trusted registry

Search

Repositories > library/wordpress > Images > latest

Repositories

Organizations

Users

library/wordpress: latest

linux/amd64 c0483a867b 169.97 MB Pushed 17 hours ago by dave.lauper 19 critical 77 major 15 minor All layers alre

Layers Components Delete Promo

pcre
4 critical 7 major

binutils
2.25-5+deb8u1
3 critical 47 major

ncurses
5.9+20140913-1+b1
2 critical 2 major

zlib
1.2.8.dfsg-2+b1
2 critical 2 major

Docker Trusted Registry
2.3.0
Universal Control Plane

binutils

VERSION	LICENSE
2.25-5+deb8u1	GPL COPYLEFT

VULNERABILITIES	SEVERITY	DESCRIPTION
CVE-2014-9939	critical	ihex.c in GNU Binutils before 2.26 contains a sta bad bytes in Intel Hex objects.
CVE-2017-7614	critical	elflink.c in the Binary File Descriptor (BFD) librar Binutils 2.28, has a "member access within null
CVE-2017-8421	critical	The function coff_set_alignment_hook in coffco library (aka libbfd), as distributed in GNU Binuti

What to do next

If you find that an image in your registry contains vulnerable components, you can use the linked CVE scan information in each scan report to evaluate the vulnerability and decide what to do.

If you discover vulnerable components, you should check if there is an updated version available where the security vulnerability has been addressed. If necessary, you might contact the component's maintainers to ensure that the vulnerability is being addressed in a future version or patch update.

If the vulnerability is in a `base layer` (such as an operating system) you might not be able to correct the issue in the image. In this case, you might switch to a different version of the base layer, or you might find an equivalent, less vulnerable base layer. You might also decide that the vulnerability or exposure is acceptable.

Address vulnerabilities in your repositories by updating the images to use updated and corrected versions of vulnerable components, or by using a different components that provide the same functionality. When you have updated the source code, run a build to create a new image, tag the image, and push the updated image to your DTR instance. You can then re-scan the image to confirm that you have addressed the vulnerabilities.

registry (https://docs.docker.com/glossary/?term=registry), scan (https://docs.docker.com/glossary/?term=scan), vulnerability (https://docs.docker.com/glossary/?term=vulnerability)