USER GUIDE

rezilion

Rezilion in GitLab CI

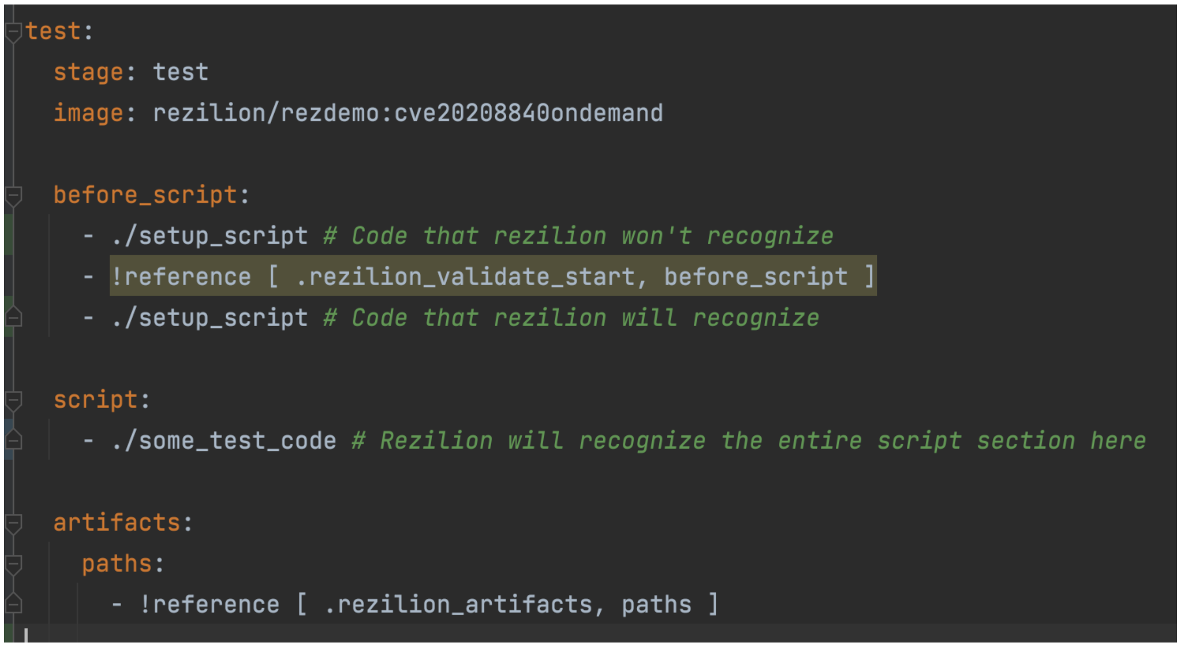
USER GUIDE

Requirements

1) Ultimate GitLab license
2) You are testing a container in your CI tests.

Limitations

1) Supported runtime environments:
    a) Native
    b) Java
    c) Node
    d) Ruby
    e) Bash
    f) Python
    g) Dotnet
    h) Perl
2) Supported package managers:
    a) rpm
    b) dpkg
    c) Maven
    d) NPM
    e) Gem
    f) pip
3) Rezilion only validates code that runs after its **validation_start** command, as shown in the example below.



```yaml
test:
  stage: test
  image: rezilion/rezdemo:cve20208840ondemand

  before_script:
    - ./setup_script # Code that rezilion won't recognize
    - !reference [ .rezilion_validate_start, before_script ]
    - ./setup_script # Code that rezilion will recognize

  script:
    - ./some_test_code # Rezilion will recognize the entire script section here

  artifacts:
    paths:
      - !reference [ .rezilion_artifacts, paths ]
```

4) Edge Case 1:
    a) What: Missing events (False Negative) - Mark loaded components as unloaded.
    b) When: Loading shared libraries to short processes using **dlopen**.
    c) Probability: Very rare.
    d) Risk: Very low.
    e) Technical Description: Shared libraries that are loaded using **dlopen** into processes that are being executed for less than 0.5 seconds.

Setup

Initial Setup

- *Please notice the use of accurate indentation when writing yaml files - A YAML file uses spaces as indentation, you can use 2 or 4 spaces for indentation, but **no tabs**.*

1) Commit “rezilion.yml” into your git repository (received by mail).
   For example:

```
git add -A rezilion.yml
git commit -m "Added rezilion yml to source control"
```

2) Add into your **include** section in ".gitlab-ci.yml" rezilion's template yml.
For example:

```
include:
    - template: Security/Container-Scanning.gitlab-ci.yml # for gitlab scanner
    - local: rezilion.yml # for rezilion
```

3) Declare Rezilion variables, inside gitlab.yml, as global variables (at the outer scope):

```
variables:
  REZILION_SCANNER: "<scanner_name>"
  LICENSE_KEY: "<license_key>"
  DOCKER_IMAGE: "<image_name>"
  CS_DISABLE_LANGUAGE_VULNERABILITY_SCAN: "false"
```

Available variables

- **REZILION_SCANNER**

  Available options:

  - gitlab_trivy (Use GitLab's built in Trivy scanner, which already have container scanning enabled for the pipeline)

- **LICENSE_KEY**

  Rezilion Validate's license key

- **DOCKER_IMAGE**

  An optional variable to specify the image to be scanned. If not specified, defaults to "$CI_APPLICATION_REPOSITORY:$CI_APPLICATION_TAG"

- **DOCKER_USER**

  An optional variable to specify the username of the registry the image contains on. If not specified, defaults to "$CI_REGISTRY_USER"

- **DOCKER_PASSWORD**

  An optional variable to specify the username of the registry the image contains on. If not specified, defaults to "$CI_REGISTRY_PASSWORD"

- **CS_DISABLE_LANGUAGE_VULNERABILITY_SCAN**
  - An optional flag, that allows the gitlab trivy scanner to show vulnerabilities for non OS package managers (such as NPM, GEM, PIP)

Setup per Test

1) Start Rezilion's validation process: Add the following line to your "before_script" section:

```
- !reference [ .rezilion_start, before_script ]
```

For example:

```
before_script:
    - !reference [ .rezilion_start, before_script ]
```

2) Add the Rezilion artifact path to your test artifacts.

For example:

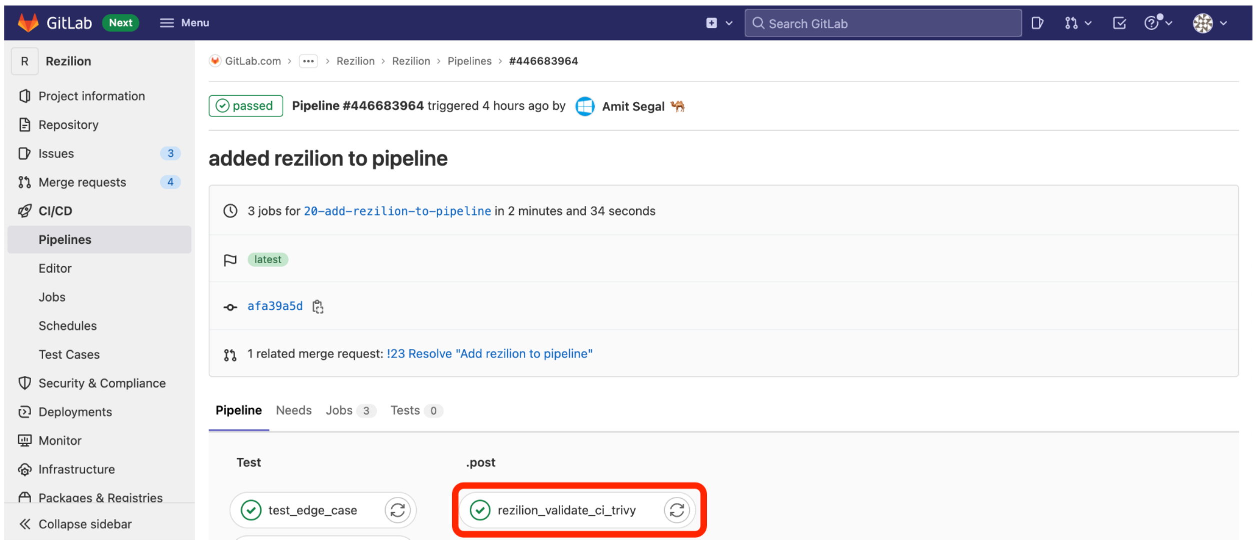
```
artifacts:
    paths:
        - !reference [ .rezilion_artifacts, paths ]
```

Run the Pipeline

- Rezilion's validation process is executed automatically for every test Rezilion is enabled on (as configured in section B).
- A new job is added to the pipeline, which contains the results of Rezilion.
- From our testing, scanning a very large image (~14K files) should only take about 10 seconds. Not including the installation of our package which depends on the speed of your network (size - 23mb).

Explore the Result by Artifact

The results can be viewed by going into the “**rezilion_validate**” job which runs at the end of the pipeline:



Pipeline #446683964 triggered 4 hours ago by Amit Segal

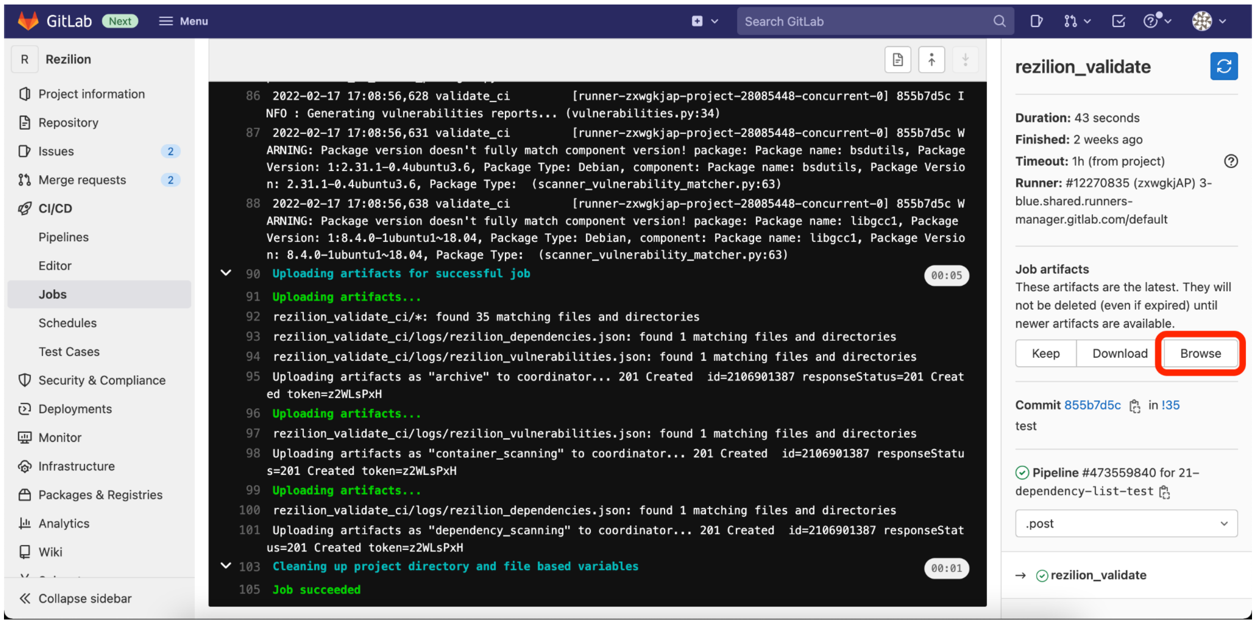
added rezilion to pipeline

3 jobs for 20-add-rezilion-to-pipeline in 2 minutes and 34 seconds

afa39a5d

1 related merge request: !23 Resolve "Add rezilion to pipeline"

And then clicking “Browse” in the artifacts section



```
2022-02-17 17:08:56,628 validate_ci         [runner-zxwgkjap-project-28085448-concurrent-0] 855b7d5c I
NFO : Generating vulnerabilities reports... (vulnerabilities.py:34)
2022-02-17 17:08:56,631 validate_ci         [runner-zxwgkjap-project-28085448-concurrent-0] 855b7d5c W
ARNING: Package version doesn't fully match component version! package: Package name: bsdutils, Package
Version: 1:2.31.1-0.4ubuntu3.6, Package Type: Debian, component: Package name: bsdutils, Package Versio
n: 2.31.1-0.4ubuntu3.6, Package Type:  (scanner_vulnerability_matcher.py:63)
2022-02-17 17:08:56,638 validate_ci         [runner-zxwgkjap-project-28085448-concurrent-0] 855b7d5c W
ARNING: Package version doesn't fully match component version! package: Package name: libgcc1, Package
Version: 1:8.4.0-1ubuntu1~18.04, Package Type: Debian, component: Package name: libgcc1, Package Versio
n: 8.4.0-1ubuntu1~18.04, Package Type:  (scanner_vulnerability_matcher.py:63)
Uploading artifacts for successful job                                                        00:05
Uploading artifacts...
rezilion_validate_ci/*: found 35 matching files and directories
rezilion_validate_ci/logs/rezilion_dependencies.json: found 1 matching files and directories
rezilion_validate_ci/logs/rezilion_vulnerabilities.json: found 1 matching files and directories
Uploading artifacts as "archive" to coordinator... 201 Created  id=2106901387 responseStatus=201 Creat
ed token=z2WLsPxH
Uploading artifacts...
rezilion_validate_ci/logs/rezilion_vulnerabilities.json: found 1 matching files and directories
Uploading artifacts as "container_scanning" to coordinator... 201 Created  id=2106901387 responseStatu
s=201 Created token=z2WLsPxH
Uploading artifacts...
rezilion_validate_ci/logs/rezilion_dependencies.json: found 1 matching files and directories
Uploading artifacts as "dependency_scanning" to coordinator... 201 Created  id=2106901387 responseStat
us=201 Created token=z2WLsPxH
Cleaning up project directory and file based variables                                       00:01
Job succeeded
```

Duration: 43 seconds
Finished: 2 weeks ago
Timeout: 1h (from project)
Runner: #12270835 (zxwgkjAP) 3-blue.shared.runners-manager.gitlab.com/default

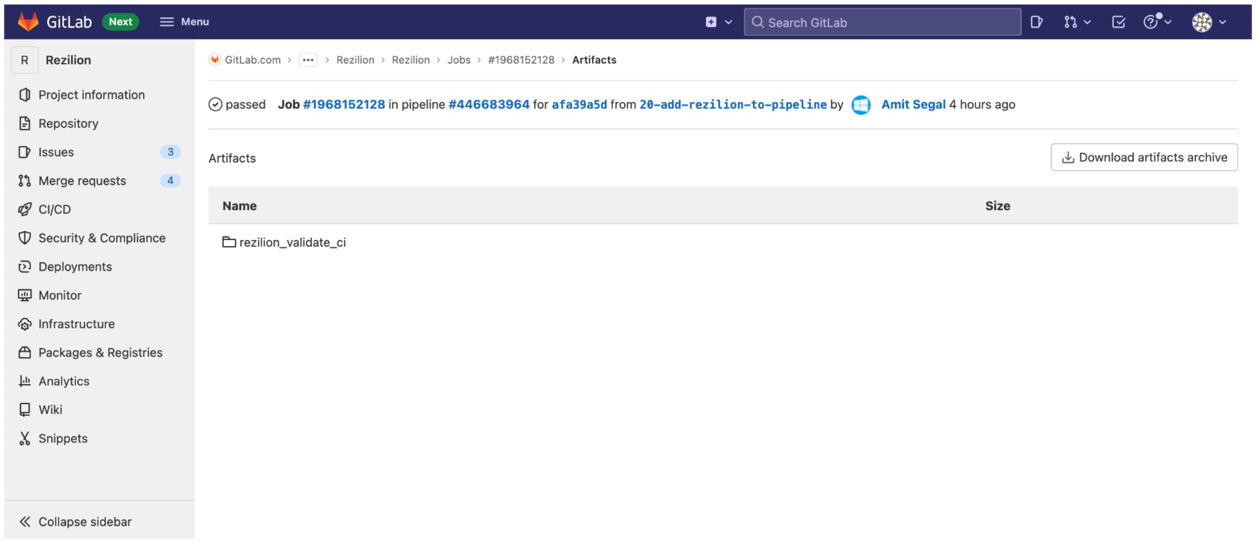
Job artifacts
These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.

Commit 855b7d5c in !35
test

Pipeline #473559840 for 21-dependency-list-test

→ rezilion_validate

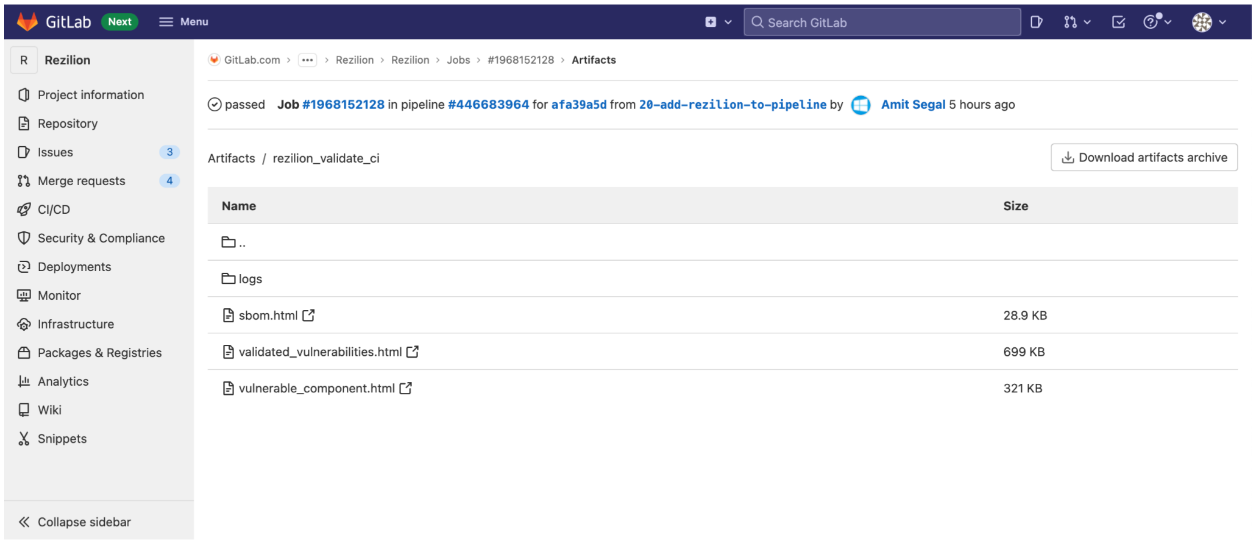
In the folder named “rezilion_validate_ci”, you’ll be able to find these 3 html files:



passed Job #1968152128 in pipeline #446683964 for afa39a5d from 20-add-rezilion-to-pipeline by Amit Segal 4 hours ago

Artifacts

Name	Size
rezilion_validate_ci	



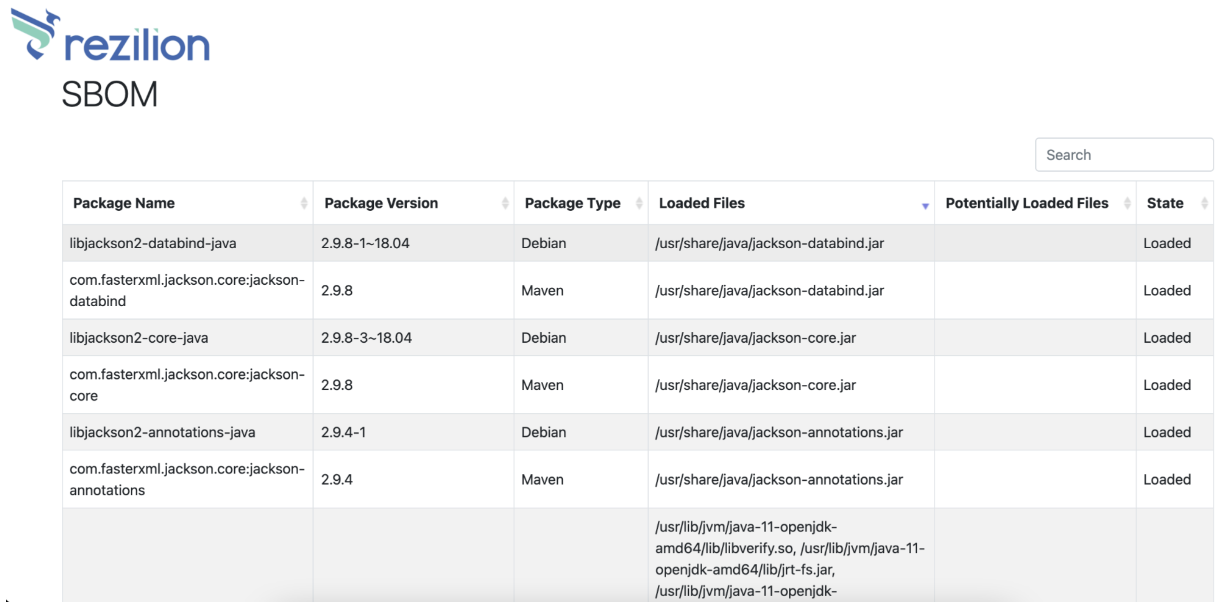
passed Job #1968152128 in pipeline #446683964 for afa39a5d from 20-add-rezilion-to-pipeline by Amit Segal 5 hours ago

Artifacts / rezilion_validate_ci

Name	Size
..	
logs	
sbom.html	28.9 KB
validated_vulnerabilities.html	699 KB
vulnerable_component.html	321 KB

SBOM (Software Bill Of Materials)

Each component Rezilion found on the scanning will be there.
Potentially loaded files happen when a nested jar/war/etc runs and finishes execution very fast (less than a second). In this scenario, the main jar/war/etc will be marked as exploitable, but the inner jars will be marked as potentially exploitable.

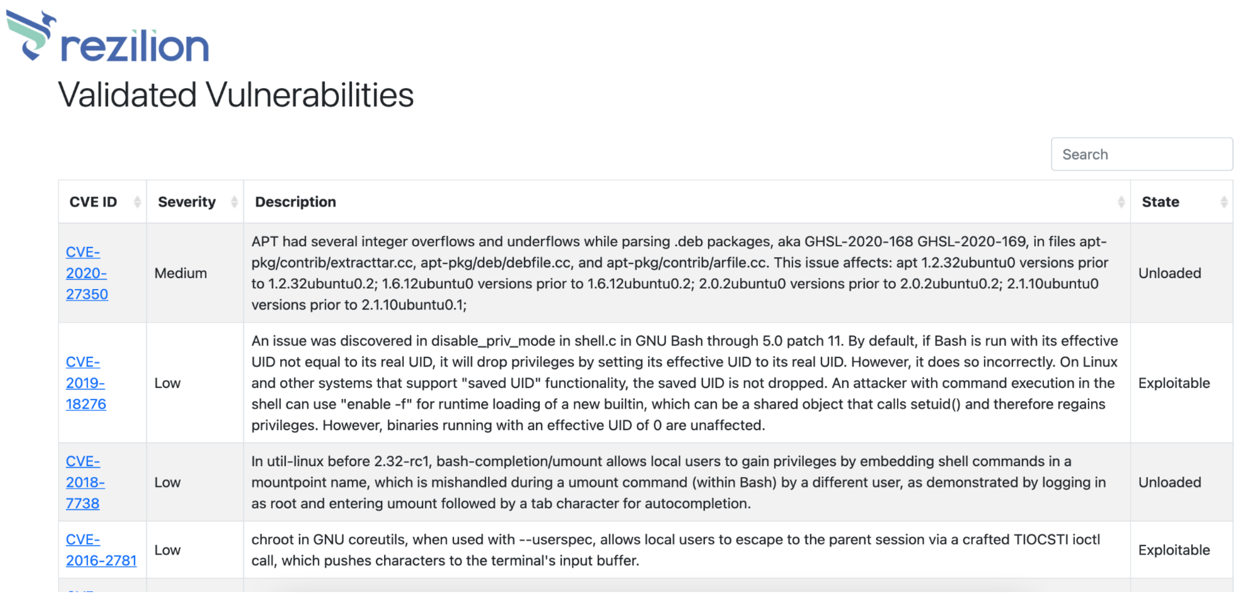


Package Name	Package Version	Package Type	Loaded Files	Potentially Loaded Files	State
libjackson2-databind-java	2.9.8-1~18.04	Debian	/usr/share/java/jackson-databind.jar		Loaded
com.fasterxml.jackson.core:jackson-databind	2.9.8	Maven	/usr/share/java/jackson-databind.jar		Loaded
libjackson2-core-java	2.9.8-3~18.04	Debian	/usr/share/java/jackson-core.jar		Loaded
com.fasterxml.jackson.core:jackson-core	2.9.8	Maven	/usr/share/java/jackson-core.jar		Loaded
libjackson2-annotations-java	2.9.4-1	Debian	/usr/share/java/jackson-annotations.jar		Loaded
com.fasterxml.jackson.core:jackson-annotations	2.9.4	Maven	/usr/share/java/jackson-annotations.jar		Loaded
			/usr/lib/jvm/java-11-openjdk-amd64/lib/libverify.so, /usr/lib/jvm/java-11-openjdk-amd64/lib/jrt-fs.jar, /usr/lib/jvm/java-11-openjdk-		

Validated Vulnerabilities

Each vulnerability the scanner found, with an **Exploitable**/**Unexploitable** state. When clicking on a vulnerability, the affected components will be shown.

- The table can be sorted by one of those fields: **CVE ID**, **Severity**, **Description**, **State.**
    - By clicking on the CVE ID you will redirect to more details about it in NVD site (NATIONAL VULNERABILITY DATABASE)
- There is an option to search by a given string in the table, using the search textbox.



Validated Vulnerabilities

Search

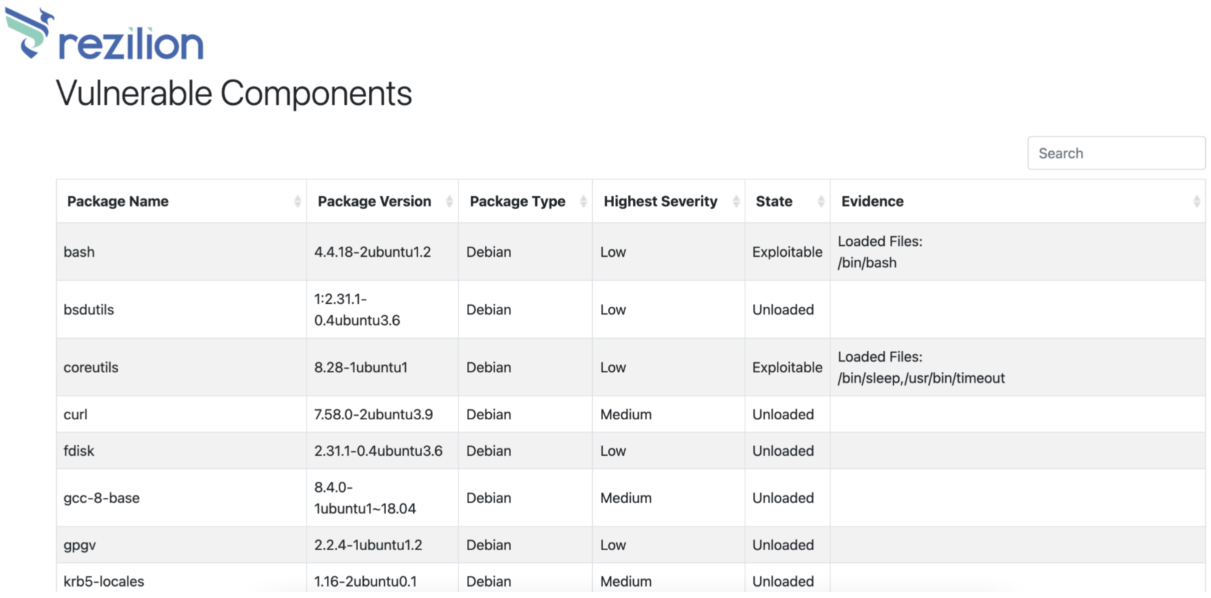
CVE ID	Severity	Description	State
CVE-2020-27350	Medium	APT had several integer overflows and underflows while parsing .deb packages, aka GHSL-2020-168 GHSL-2020-169, in files apt-pkg/contrib/extracttar.cc, apt-pkg/deb/debfile.cc, and apt-pkg/contrib/arfile.cc. This issue affects: apt 1.2.32ubuntu0 versions prior to 1.2.32ubuntu0.2; 1.6.12ubuntu0 versions prior to 1.6.12ubuntu0.2; 2.0.2ubuntu0 versions prior to 2.0.2ubuntu0.2; 2.1.10ubuntu0 versions prior to 2.1.10ubuntu0.1;	Unloaded
CVE-2019-18276	Low	An issue was discovered in disable_priv_mode in shell.c in GNU Bash through 5.0 patch 11. By default, if Bash is run with its effective UID not equal to its real UID, it will drop privileges by setting its effective UID to its real UID. However, it does so incorrectly. On Linux and other systems that support "saved UID" functionality, the saved UID is not dropped. An attacker with command execution in the shell can use "enable -f" for runtime loading of a new builtin, which can be a shared object that calls setuid() and therefore regains privileges. However, binaries running with an effective UID of 0 are unaffected.	Exploitable
CVE-2018-7738	Low	In util-linux before 2.32-rc1, bash-completion/umount allows local users to gain privileges by embedding shell commands in a mountpoint name, which is mishandled during a umount command (within Bash) by a different user, as demonstrated by logging in as root and entering umount followed by a tab character for autocompletion.	Unloaded
CVE-2016-2781	Low	chroot in GNU coreutils, when used with --userspec, allows local users to escape to the parent session via a crafted TIOCSTI ioctl call, which pushes characters to the terminal's input buffer.	Exploitable

Vulnerable Components

A list of components found by the vulnerability scanner.
Each row represents a component, with relevant evidence in case it is loaded.
In case when a nested jar/war/etc runs and finishes execution very fast (less than a second). The main jar/war/etc will be marked as exploitable, but the inner jars will be marked as potentially exploitable.

When clicking on a component, a list of its associated vulnerabilities will be displayed.

- The table can be sorted by one of those fields: **CVE ID**, **Severity**, **Description**, **State.**
    - By clicking on the CVE ID you will redirect to more details about it in NVD site (NATIONAL VULNERABILITY DATABASE)
- There is an option to search by a given string in the table, using the search textbox.


Vulnerable Components

Search

Package Name	Package Version	Package Type	Highest Severity	State	Evidence
bash	4.4.18-2ubuntu1.2	Debian	Low	Exploitable	Loaded Files: /bin/bash
bsdutils	1:2.31.1-0.4ubuntu3.6	Debian	Low	Unloaded	
coreutils	8.28-1ubuntu1	Debian	Low	Exploitable	Loaded Files: /bin/sleep,/usr/bin/timeout
curl	7.58.0-2ubuntu3.9	Debian	Medium	Unloaded	
fdisk	2.31.1-0.4ubuntu3.6	Debian	Low	Unloaded	
gcc-8-base	8.4.0-1ubuntu1~18.04	Debian	Medium	Unloaded	
gpgv	2.2.4-1ubuntu1.2	Debian	Low	Unloaded	
krb5-locales	1.16-2ubuntu0.1	Debian	Medium	Unloaded	

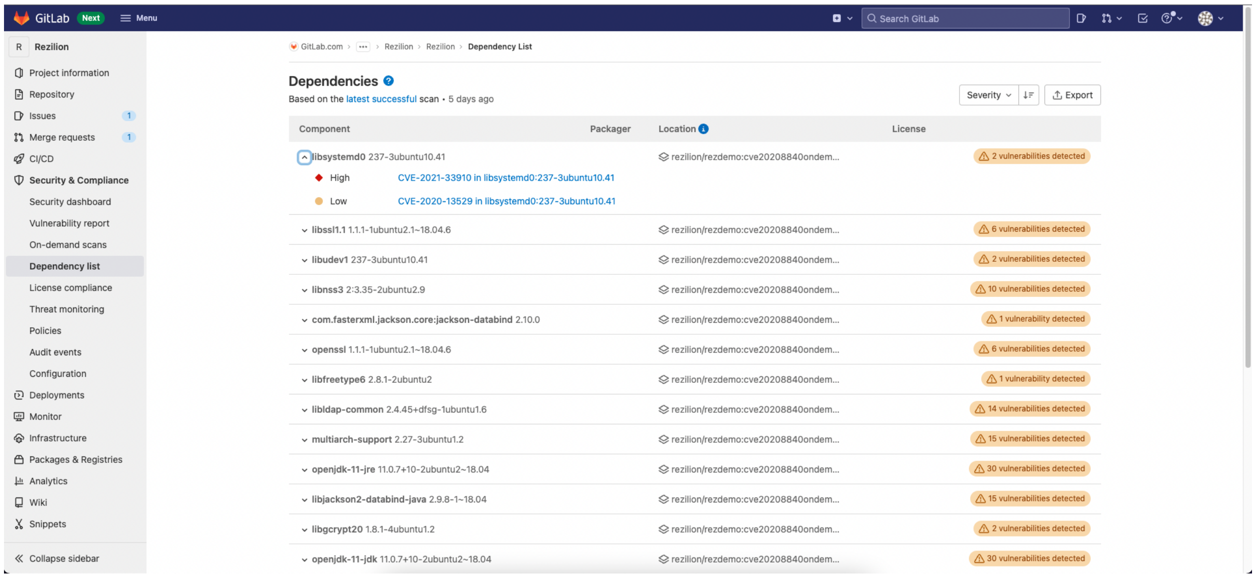
Explore the Results by GitLab UI

The results can also be viewed through the GitLab security dashboard:

Dependency List (SBOM)

The dependency list to review your project's dependencies and key details about those dependencies, including their known vulnerabilities. It is a collection of dependencies in your project, including existing and new findings. The dependency list only shows the results of the last successful pipeline to run on the default branch.

-



GitLab.com › ⋯ › Rezilion › Rezilion › Dependency List

Dependencies

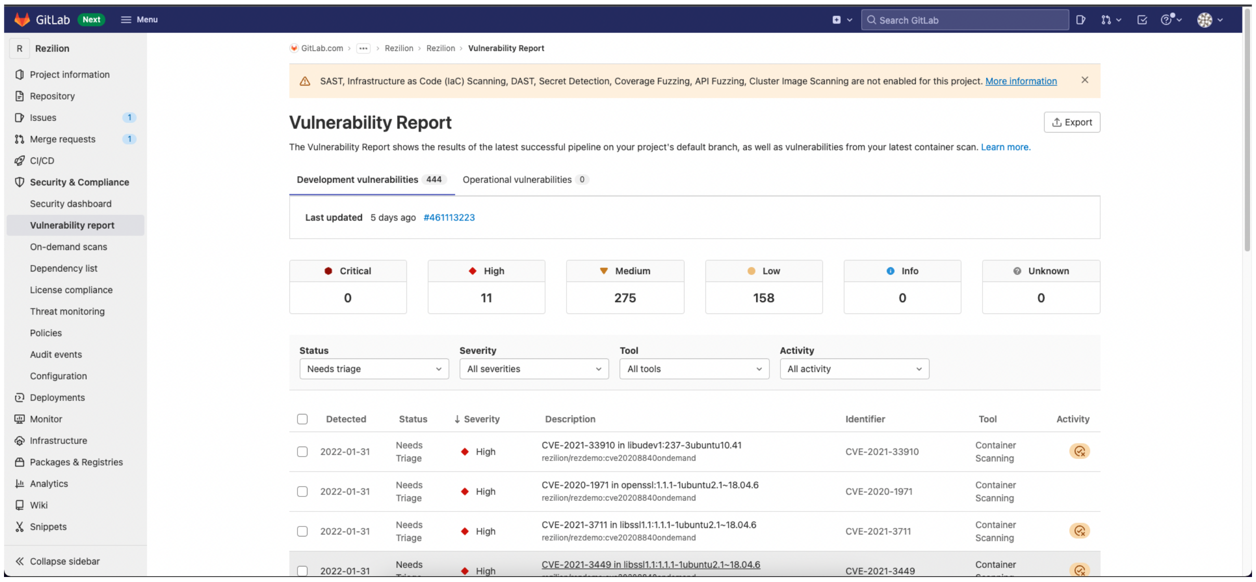
Based on the latest successful scan • 5 days ago

Component	Packager	Location	License	
libsystemd0 237-3ubuntu10.41		rezilion/rezdemo:cve20208840ondem...		2 vulnerabilities detected
◆ High	CVE-2021-33910 in libsystemd0:237-3ubuntu10.41			
● Low	CVE-2020-13529 in libsystemd0:237-3ubuntu10.41			
libssl1.1 1.1.1-1ubuntu2.1~18.04.6		rezilion/rezdemo:cve20208840ondem...		6 vulnerabilities detected
libudev1 237-3ubuntu10.41		rezilion/rezdemo:cve20208840ondem...		2 vulnerabilities detected
libnss3 2:3.35-2ubuntu2.9		rezilion/rezdemo:cve20208840ondem...		10 vulnerabilities detected
com.fasterxml.jackson.core:jackson-databind 2.10.0		rezilion/rezdemo:cve20208840ondem...		1 vulnerability detected
openssl 1.1.1-1ubuntu2.1~18.04.6		rezilion/rezdemo:cve20208840ondem...		6 vulnerabilities detected
libfreetype6 2.8.1-2ubuntu2		rezilion/rezdemo:cve20208840ondem...		1 vulnerability detected
libldap-common 2.4.45+dfsg-1ubuntu1.6		rezilion/rezdemo:cve20208840ondem...		14 vulnerabilities detected
multiarch-support 2.27-3ubuntu1.2		rezilion/rezdemo:cve20208840ondem...		15 vulnerabilities detected
openjdk-11-jre 11.0.7+10-2ubuntu2~18.04		rezilion/rezdemo:cve20208840ondem...		30 vulnerabilities detected
libjackson2-databind-java 2.9.8-1~18.04		rezilion/rezdemo:cve20208840ondem...		15 vulnerabilities detected
libgcrypt20 1.8.1-4ubuntu1.2		rezilion/rezdemo:cve20208840ondem...		2 vulnerabilities detected
openjdk-11-jdk 11.0.7+10-2ubuntu2~18.04		rezilion/rezdemo:cve20208840ondem...		30 vulnerabilities detected

Vulnerability Report

The Vulnerability Report provides information about vulnerabilities from scans of the default branch. At all levels, the Vulnerability Report contains:

- Totals of vulnerabilities per severity level.
- Filters for common vulnerability attributes.
- Details of each vulnerability, presented in tabular layout.
- The Activity column contains an icon to indicate if the vulnerability is exploitable .



GitLab.com › ⋯ › Rezilion › Rezilion › Vulnerability Report

SAST, Infrastructure as Code (IaC) Scanning, DAST, Secret Detection, Coverage Fuzzing, API Fuzzing, Cluster Image Scanning are not enabled for this project. More information

Vulnerability Report

The Vulnerability Report shows the results of the latest successful pipeline on your project's default branch, as well as vulnerabilities from your latest container scan. Learn more.

Development vulnerabilities 444 | Operational vulnerabilities 0

Last updated 5 days ago #461113223

Critical	High	Medium	Low	Info	Unknown
0	11	275	158	0	0

Status: Needs triage
Severity: All severities
Tool: All tools
Activity: All activity

Detected	Status	↓ Severity	Description	Identifier	Tool	Activity
2022-01-31	Needs Triage	◆ High	CVE-2021-33910 in libudev1:237-3ubuntu10.41 rezilion/rezdemo:cve20208840ondemand	CVE-2021-33910	Container Scanning	●
2022-01-31	Needs Triage	◆ High	CVE-2020-1971 in openssl:1.1.1-1ubuntu2.1~18.04.6 rezilion/rezdemo:cve20208840ondemand	CVE-2020-1971	Container Scanning	
2022-01-31	Needs Triage	◆ High	CVE-2021-3711 in libssl1.1:1.1.1-1ubuntu2.1~18.04.6 rezilion/rezdemo:cve20208840ondemand	CVE-2021-3711	Container Scanning	●
2022-01-31	Needs	◆ High	CVE-2021-3449 in libssl1.1:1.1.1-1ubuntu2.1~18.04.6	CVE-2021-3449	Container	●