Package: oysteR (via r-universe)

September 13, 2024

Title Scans R Projects for Vulnerable Third Party Dependencies

Version 0.1.3.9001

Maintainer Colin Gillespie <csgillespie@gmail.com>

Description Collects a list of your third party R packages, and scans them with the 'OSS' Index provided by 'Sonatype', reporting back on any vulnerabilities that are found in the third party packages you use.

License Apache License 2.0 | file LICENSE

URL https://github.com/sonatype-nexus-community/oysteR

BugReports https://github.com/sonatype-nexus-community/oysteR/issues

Depends R (>= 3.5.0)

Imports cli, dplyr, glue, httr, jsonlite, purrr, rjson, rlang, stringr, tibble, tidyr, utils, yaml

Suggests covr, httptest, knitr, rmarkdown, testthat (>= 2.1.0)

VignetteBuilder knitr

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Repository https://sonatype-nexus-community.r-universe.dev

RemoteUrl https://github.com/sonatype-nexus-community/oyster

RemoteRef HEAD

RemoteSha 236e746a1f3f374ea22f30271ec784d32d8fcc6f

Contents

audit 2
audit_conda 3
audit_deps 3

audit_description 4
audit_installed_r_pkgs 5
audit_renv_lock 5
audit_req_txt 6
expect_secure 7
get_vulnerabilities 7
remove_cache 8

Index **9**

`audit` *Search for Package Vulnerabilities*

Description

Search the OSS Index for known package vulnerabilities in any of the supported ecosystems— e.g. CRAN, PyPI, Conda, NPM, Maven, etc. see https://ossindex.sonatype.org/ecosystems for full list.

Usage

```
audit(pkg, version, type, verbose = TRUE)
```

Arguments

`pkg`	A vector of package names to search in the OSS Index.
`version`	The specific package version to search for. By default it will search all known versions. If not `*`, must be the same length as pkg.
`type`	The package management environment. For R packages, set equal to "cran". This defaults to `"cran"`. See https://ossindex.sonatype.org/ecosystems.
`verbose`	Default `TRUE`.

Examples

```
pkg = c("abind", "acepack")
version = c("1.4-5", "1.4.1")
audit(pkg, version, type = "cran")
```

audit_conda *Audit a conda environment file*

Description

This function searches the OSS index for vulnerabilities recorded for packages listed in a Conda environment file typically called `environment.yml` but are subject to varied names. Conda environment can contain packages from both Conda and PyPI. All packages will be audited.

Usage

```
audit_conda(dir = ".", fname = "environment.yml", verbose = TRUE)
```

Arguments

`dir`	The directory containing a Conda environment yaml file.
`fname`	The file name of conda environment yaml file.
`verbose`	Default `TRUE`.

Examples

```
## Not run:
# Looks for a environment.yml file in dir
audit_conda(dir = ".")

## End(Not run)
```

audit_deps *Check Package Dependencies*

Description

Collects R dependencies and checks them against OSS Index. Returns a tibble of results.

Usage

```
audit_deps(pkgs = NULL, verbose = TRUE)
```

Arguments

`pkgs`	Default `NULL`. See details for further information.
`verbose`	Default `TRUE`.

Details

This function is deprecated. See

By default, packages listed in `installed.packages()` are scanned by sonatype. However, you can pass your own data frame of packages. This data frame should have two columns, `version` and `package`.

Value

A tibble/data.frame.

`audit_description` *Audits Packages Listed in a DESCRIPTION file*

Description

Looks for a DESCRIPTION file in `dir`, then extract the packages in the fields & calculates the dependency tree.

Usage

```
audit_description(
  dir = ".",
  fields = c("Depends", "Imports", "Suggests"),
  verbose = TRUE
)
```

Arguments

`dir`	The file path of an renv.lock file.
`fields`	The DESCRIPTION field to parse. Default is Depends, Import, & Suggests.
`verbose`	Default `TRUE`.

Examples

```
## Not run:
# Looks for a DESCRIPTION file in dir
audit_description(dir = ".")

## End(Not run)
```

audit_installed_r_pkgs *Audit Installed Packages*

Description

Audits all installed packages by calling `installed.packages()` and checking them against the OSS Index.

Usage

```
audit_installed_r_pkgs(verbose = TRUE)
```

Arguments

verbose	Default TRUE.

Value

A tibble/data.frame.

Examples

```
## Not run:
# Audit installed packages
# This calls installed.packages()
pkgs = audit_installed_r_pkgs()

## End(Not run)
```

audit_renv_lock *Audit an renv.lock File*

Description

This function searches the OSS index for vulnerabilities recorded for packages listed in an `renv.lock` file. An `renv.lock` file is created by the `{renv}` package which is used for project level package management in R.

Usage

```
audit_renv_lock(dir = ".", verbose = TRUE)
```

Arguments

dir	The file path of an renv.lock file.
verbose	Default TRUE.

Examples

```
## Not run:
# Looks for renv.lock file in dir
audit_renv_lock(dir = ".")

## End(Not run)
```

`audit_req_txt` *Audit a requirements.txt File*

Description

This function searches the OSS index for vulnerabilities recorded for packages listed in a requirements.txt file based on PyPi.

Usage

```
audit_req_txt(dir = ".", verbose = TRUE)
```

Arguments

`dir`	The file path of a requirements.txt file.
`verbose`	Default `TRUE`.

Details

pip is a standard of python package management based on the Python Package Index (PyPI). pip uses a requirements.txt file to manage of Python libraries. The requirements.txt file contains package names and versions (often used to manage a virtual environment).

Examples

```
## Not run:
# Looks for a requirements.txt file in dir
audit_description(dir = ".")

## End(Not run)
```

expect_secure *Vulnerability Detection via Testthat*

Description

A `testthat` version for detecting vulnerabilities. This function is used within the `testthat` framework. As testthat strips out the repositories from options, we have to set the value locally in the function, i.e. the value you have in `getOption("repos")` is not used.

Usage

```
expect_secure(pkg, repo = "https://cran.rstudio.com", verbose = FALSE)
```

Arguments

`pkg`	The pkg to check
`repo`	The CRAN repository, used to get version numbers
`verbose`	Default `TRUE`.

Details

An important proviso is that we are only testing packages for specific versions. By default, this will be the latest version on CRAN. This may differ for users or if you are using a CRAN snapshot. For the latter, simply change the `repo` parameter.

Examples

```
## Not run:
 # Typically used inside testthat
 oysteR::expect_secure("oysteR")

## End(Not run)
```

get_vulnerabilities *Extract vulnerabilities*

Description

Parse the audit data frame (obtained via `audit_deps`), and extract the vulnerabilities.

Usage

```
get_vulnerabilities(audit)
```

Arguments

audit	Output from audit_deps.

Examples

```
## Not run:
# Audit installed packages
# This calls installed.packages()
# pkgs = audit_deps()

# Or pass your own packages
pkgs = data.frame(package = c("abind", "acepack"),
                  version = c("1.4-5", "1.4.1"))
#deps = audit_deps(pkgs)
#get_vulnerabilities(deps)

## End(Not run)
```

---

remove_cache *Remove cache*

Description

The OSS cache is located at `tools::R_user_dir("oysteR", which = "cache")`. The function `R_user_dir()` is only available for R >= 4.0.0. Packages are cached for 12 hours, then refreshed at the next audit

Usage

```
remove_cache()
```

Index

audit, 2
audit_conda, 3
audit_deps, 3
audit_description, 4
audit_installed_r_pkgs, 5
audit_renv_lock, 5
audit_req_txt, 6

expect_secure, 7

get_vulnerabilities, 7

remove_cache, 8