nix-community / vulnix

Code | Issues 17 | Pull requests 2 | Actions | Projects | Security | Insights

Vulnerability (CVE) scanner for Nix/NixOS [maintainer=@henrirosten]

BSD-3-Clause license

Code of conduct

Security policy

523 stars | 39 forks | 14 watching | Branches | Activity | Custom properties | Tags

Public repository

1 Branch | 35 Tags | Go to file t | Go to file | Add file | Code

henrirosten Apply ruff formatter changes 5e635e4 · 2 months ago

.github/workflows	Run nix flake check in github actions	2 months ago
doc	Add --profile option to scan user en...	4 years ago
nix	Use ruff as python linter	2 months ago
src/vulnix	Apply ruff formatter changes	2 months ago
.editorconfig	README.rst: Add syntax highlighting...	3 years ago
.gitignore	Initial man pages	7 years ago
CHANGES.rst	Back to development: 1.11.0	2 months ago
HACKING.rst	Fix incorrect processing of the 'until'...	6 years ago
LICENSE	remote (outdated) year from license	6 years ago
MANIFEST.in	Fix packaging bug	6 years ago
README.rst	Remove new maintainer needed ann...	3 months ago
VERSION	Back to development: 1.11.0	2 months ago
default.nix	Apply initial treefmt formatters	2 months ago
flake.lock	Add nix treefmt	2 months ago
flake.nix	Apply initial treefmt formatters	2 months ago
setup.cfg	Remove pytest-flake8 check	2 months ago
setup.py	Apply ruff formatter changes	2 months ago
shell.nix	Apply initial treefmt formatters	2 months ago

README Code of conduct BSD-3-Clause license Security

Nix(OS) vulnerability scanner

This is a utility that validates a Nix store for any packages that are reachable from live paths and likely to be affected by vulnerabilities listed in the NVD.

It implements a CLI utility to inspect the current status and a monitoring integration for Sensu.

Example output

```
2 derivations with active advisories

----------------------------------------------------------------------
binutils-2.31.1

/nix/store/zc1lbkaf9l9hlsp1jdiq3si56nsglymh-binutils-2.31.1.drv
CVE                                              CVSSv3
https://nvd.nist.gov/vuln/detail/CVE-2018-1000876  7.8
https://nvd.nist.gov/vuln/detail/CVE-2018-20657    7.5
https://nvd.nist.gov/vuln/detail/CVE-2018-20712    6.5

----------------------------------------------------------------------
libssh2-1.9.0

/nix/store/mfpfclry68r4sv4mcc9hb88z0lb9jk1c-libssh2-1.9.0.drv
CVE                                              CVSSv3
https://nvd.nist.gov/vuln/detail/CVE-2019-17498    8.1
```

Theory of operation

`vulnix` pulls all published CVEs from NIST and caches them locally. It matches name and version of all derivations referenced from the command line against known CVE entries. A *whitelist* is used to filter out unwanted results.

Matching Nix package names to NVD products is currently done via a coarse heuristic. First, a direct match is tried. If no product can be found, variations with lower case and underscore instead of hyphen are tried. It is clear that this mapping is too simplistic and needs to be improved in future versions.

System requirements

- Depends on common Nix tools like `nix-store`. These are expected to be in $PATH.
- Depends on being able to interact with the Nix store database (/nix/var/nix/db). This means that it must either run as the same user that owns the Nix store database or `nix-daemon` must be active.
- Parses `*.drv` files directly. Tested with Nix >=1.10 and 2.x.
- It refuses to work without some locale environment settings. Try `export LANG=C.UTF-8` if you see encoding errors.

Usage Example

- What vulnerabilities are listed for my current system

```
vulnix --system
```

- Check `nix-build` output together with its transitive closure

```
vulnix result/
```

- Check all passed derivations, but don't determine requisites

```
vulnix -R /nix/store/*.drv
```

- JSON output for machine post-processing

```
vulnix --json /nix/store/my-derivation.drv
```

See `vulnix --help` for a list of all options.

Whitelisting

`vulnix` output may contain false positives, unfixable packages or stuff which is known to be addressed. The *whitelist* feature allows to exclude packages matching certain criteria.

Usage

Load whitelists from either local files or HTTP servers

```
vulnix -w /path/to/whitelist.toml \
       -w https://example.org/published-whitelist.toml
```

Syntax

Whitelists are TOML files which contain the package to be filtered as section headers, followed by further per-package options.

Section headings - package selection

Exclude a package at a specific version

```
["openjpeg-2.3.0"]
```

Exclude a package regardless of version (additional CVE filters may apply, see below)

```
["openjpeg"]
```

Exclude all packages (see below for CVE filters, again)

```
["*"]
```

Options

cve

List of CVE identifiers to match. The whitelist rule is valid as long as the detected CVEs are a subset of the CVEs listed here. If additional CVEs are detected, this whitelist rule is not effective anymore.

until
Date in the form "YYYY-MM-DD" which confines this rule's lifetime. On the specified date and later, this whitelist rule is not effective anymore.

issue_url
URL or list of URLs that point to any issue tracker. Informational only.

comment
String or list of strings containing free text. Informational only.

Examples

Create a ticket on your favourite issue tracker. Estimate the time to get the vulnerable package fixed. Create whitelist entry:

```
["ffmpeg-3.4.2"]
cve = ["CVE-2018-6912", "CVE-2018-7557"]
until = "2018-05-01"
issue_url = "https://issues.example.com/29952"
comment = "need to backport patch"
```

This particular version of ffmpeg will be left out from reports until either another CVE gets published or the specified date is reached.

CVE patch auto-detection

`vulnix` will inspect derivations for patches which supposedly fix specific CVEs. When a patch filename contains one or more CVE identifiers, these will not reported anymore. Example Nix code:

```
{
  patches = [ ./CVE-2018-6951.patch ];
}
```

Patches which fix multiple CVEs should name them all with a non-numeric separator, e.g. `CVE-2017-14159+CVE-2017-17740.patch` .

Auto-detection even works when patches are pulled via `fetchpatch` and friends as long as there is a CVE identifier in the name. Example:

```
{
  patches = [
    (fetchpatch {
      name = "CVE-2018-9055.patch";
      url = http://paste.opensuse.org/view/raw/330751ce;
      sha256 = "0m798m6c4v9yyhql7x684j5kppcm6884n1rrb9ljz8p9aqq2jqnm";
    })
  ];
}
```

Releases 2

Release 1.10.2 Latest
on Dec 13, 2024

+ 1 release

Sponsor this project

opencollective.com/**nix-community**

Packages

No packages published

Contributors 19

+ 5 contributors

Languages

- **Python** 91.9%
- **Nix** 8.1%