snyk Snyk CLI Cheat Sheet

www.snyk.io

Installation

```
$ npm install -g snyk
```

Once installed, you need to authenticate with your Snyk account:

`$ snyk auth` (will take you to a browser)

`$ snyk auth [api-token]` (for local testing)

For CI testing we recommend creating an environment variable called `SNYK_TOKEN` that is set to your auth token, and running `snyk auth`

CLI commands

All the following CLI commands must be run in the project folder:

`$ snyk test` To test your project for known vulnerabilities

`$ snyk wizard` Runs snyk test with an interactive wizard for fixing issues locally

`$ snyk monitor` Tests against known dependencies, uploads to snyk UI, snapshots current dependencies (manifests only), enables continuous monitoring

`$ snyk ignore` Ignore a vulnerability for a certain period of time

```
snyk ignore --id=npm:tough-cookie:20160722 --expiry=2019-04-30
--reason='Not currently exploitable'
```

`$ snyk protect` Applies the patches specified in your `.snyk` file to the local file system

`$ snyk policy` Displays your snyk policy file

Docker

`$ snyk test --docker myapp:mytag`
Test the image for vulnerabilities and receive remediation advice per vulnerability

`$ snyk test --docker myapp:mytag --file=path/to/Dockerfile`
Test the image for vulnerabilities and receive remediation advice per vulnerability and as alternative base images for your Dockerfile

`$ snyk monitor --docker ubuntu:latest`
Create a snapshot of the image's dependencies for continuous monitoring

Common Snyk CLI options

`$ snyk [cmd] --org=my-team`
Associate a test, a snapshot or a wizard snapshot with a specific organization

`$ snyk [cmd] --file=package.json`
If you have multiple manifest files, you can specify a manifest file using `--file`

`$ snyk [cmd] --file=req.txt --package-manager=pip`
Custom named manifests require you specify the package manager and the manifest file

`$ snyk [cmd] --dev`
Dev dependencies are disabled by default. To enable them, use `--dev`

`$ snyk test https://github.com/snyk/goof`
Test a public GitHub repo

`$ snyk test lodash` (latest)
`$ snyk test ionic@1.6.5`
Test a public npm package

`$ snyk test -- -Dpkg_version=1.4`
Test maven/gradle with properties

`$ snyk monitor --project-name=myapp`
Overriding a project name

`$ snyk test --json`
Get JSON output from snyk test commands

`$ snyk test --json | jq '. | (.vulnerabilities[] | select(.CVSSv3 | contains("AV:N")))'`
Use the jq JSON processor to filter the JSON output for just those vulnerabilities that have a CVSSv3 network attack vector

Also, check out our Snyk JSON to HTML mapper

`--ignore-policy`
Ignores and resets the state of your policy file

`--trust-policies`
Applies and uses ignore rules from your dependencies' Snyk policies; otherwise ignore policies are only shown as a suggestion

`--show-vulnerable-paths`
Display the dependency paths from the top level dependencies down to the vulnerable packages (defaults to true). Applicable to `snyk test`

`--dry-run`
Don't apply updates or patches during protect

Advanced CLI usage

`$ snyk config clear`
Flush out the API key

Point to the `gradle.build.kts` file if using a Gradle script with Kotlin DSL. Also, pass a Gradle configuration file

```
$ snyk test --file=build.gradle.kts --package-manager=gradle
-- --configuration <configuration>
```

Troubleshooting

Running out of tests on an OS project?

1. Run `snyk monitor`
2. On the Snyk UI go to the settings of the project
3. Enter the URL to your OS repo in the “Git remote URI” field

Failing to install Snyk CLI?
This might be a permissions issue, try installing with `sudo`

Can’t find the snyk command after install?
Change the permissions of the snyk file using `chmod -R +x ./snyk`

Remediation commands not working on your project?
Check the docs to see the latest supported languages

Unexpected test results?
For the most accurate test results, download project dependencies before running snyk test, for example:

```
$ npm install
$ mvn install
$ dotnet restore
$ dep ensure
```