1. Avoid publishing secrets to the npm registry

- Run `npm publish --dry-run` to review the package before publishing
- Put sensitive files in `.gitignore`
- Use the `files` property in `package.json` to whitelist files and directories

2. Enforce lockfile

Freeze lockfile and ensure the npm CLI installs per lockfile only, without changing it. In CI and build environments favor:

- `$ npm ci`
- `$ yarn install --frozen-lockfile`

3. Minimize attack surface—ignore run-scripts

Malicious packages take advantage of key lifecycle events when an npm install runs arbitrary commands.

To minimize this attack surface:

- Assess a project’s health status and credibility before installing a package
- Disable run-scripts during install such as:
  
  ```
  $ npm install <package> --ignore-scripts
  ```

4. Assess npm project health

Review a project for outdated dependencies, and assess environment health with CLI commands:

- `$ npm doctor`
- `$ npm outdated`

5. Scan and monitor for vulnerabilities in open source dependencies

Don’t let vulnerabilities in your project dependencies reduce the security of your application. Make sure to:

1. Connect Snyk to GitHub or other SCMs for optimal CI/CD integration with your projects
2. Run `snyk test` to scan a new project from the CLI
3. Run `snyk monitor` to track and open PRs to automatically fix security vulnerabilities in open source dependencies.

6. Use a local npm proxy

A local private registry such as Verdaccio will give you an extra layer of security, enabling you:

1. Full control of lightweight private package hosting
2. To cache packages and avoid being affected by network and external incidents

   Easily spin up verdaccio using docker:

   ```
   $ docker run verdaccio/verdaccio
   ```

7. Responsible disclosure

Publicly disclosed security vulnerabilities without prior warning and proper coordination pose a potentially serious threat.

We are happy to collaborate on responsible security disclosures for the npm community:

1. Report a security issue via the [vulnerability disclosure form](https://forms.gle/3C8L337tR1ZfZmWuA)
2. Email us at [security@snyk.io](mailto:security@snyk.io)

8. Enable 2FA

Enable two-factor authentication on npm with

```
$ npm profile enable-2fa auth-and-writes
```

9. Use npm author tokens

Make use of restricted tokens for querying npm packages and functionalities from CI by creating a read-only and IPv4 address range restricted token:

```
$ npm token create --read-only --cidr=192.0.2.0/24
```

10. Understand typosquatting risks

Typos in package installation can be deadly.

1. Be mindful when copy-pasting package install instructions to the terminal and verify authenticity.
2. Opt to have a logged-out npm user in your developer environment
3. Favor npm install with `--ignore-scripts`

Authors:

- [@liran_tal](https://twitter.com/liran_tal)
  Node.js Security WG & Developer Advocate at Snyk
- [@jotadeveloper](https://twitter.com/jotadeveloper)
  Core maintainer at Verdaccio

---

www.snyk.io

A local private registry such as Verdaccio will give you an extra layer of security, enabling you:

- Publicly disclosed security vulnerabilities without prior warning and proper coordination pose a potentially serious threat.
- We are happy to collaborate on responsible security disclosures for the npm community: