10 NPM SECURITY BEST PRACTICES

BYTESAFE

1. Add an extra layer of security by using a private registry

Add an npm proxy within your control by using a private registry

- ☑ Use a Bytesafe private registry to cache packages and centralize dependency management
- ☑ Scan for potential issues
- ☑ Control what packages and versions are allowed

2. Continuously scan and monitor all packages for security issues

Don't just rely on triggered scans (like 'npm audit'!). **Continuously scan and monitor your packages and dependencies for security issues**

- ☑ Enable security scanning and monitoring with Bytesafe
- ☑ Only allow scanned and secure packages to automatically be added to your registry
- ☑ Get notifications where you are working. Get notifications directly in your Slack

3. Stay on top of your open source licenses

Using a package with the wrong license could have catastrophic consequences, so **don't leave licenses as an afterthought!** Scan packages for license issues

- ☑ Use tools like Bytesafe to identify license information in all files. Packages can have multiple licenses
- ☑ Unlicensed packages are a problem
- ☑ Scan packages for license issues and get notified when issues are identified.

4. Enable a dependency firewall to block packages at the door

Only add dependencies that you trust!

- ☑ Enable Scanned & Secure Policies to only allow scanned and secure packages to be added to your registry
- ☑ Enable and configure a Blacklist to block specific packages or versions

5. Shift responsibility for dependencies from individuals to teams

Take responsibility for project dependencies as a team. Review and exercise caution when adding new dependencies to not add unnecessary complexity.

- ☑ Add only packages when needed and with intention

6. Don't run scripts by default when installing packages

When installing packages there are often scripts executed as part of the installation process. The feature is convenient and useful, but executing random scripts is also a major risk. Make sure you know what is executed when installing packages.

```
npm install <package> --ignore-scripts
```

7. Be aware of typosquatting risks

The devil is in the details. Make sure to only install intended packages!

- ☑ Double check what you install. Review before you run!
- ☑ Be aware of typosquatting risks

8. Keep your tokens and passwords secure and secret

Centralize token management and avoid accidental publishing of secrets

- ☑ Store maintainer tokens and publish directly with Bytesafe
- ☑ Update ignore files to avoid accidentally publishing secrets

9. Build applications using the exact same package versions

Use the right tools to get deterministic results across different environments

- ☑ Understand and use lock files (package-lock, yarn-lock etc.)
- ☑ Use `npm ci` and not `npm install` if you want to reproduce a specific node_module state
- ☑ Make an entire registry read-only with Bytesafe Freeze policy

10. Make sure the whole team uses the private registry

Your team's code supply chain is only as strong as its weakest link. Source all packages from a central source.

- ☑ Connect additional package sources to Bytesafe as upstreams (git repositories or other private/public registries)
- ☑ Make sure that the whole team uses the private registry

```
npm config set registry 'https://example.bytesafe.dev/r/default/'
```

We are here to help!

If you need any guidance or consultation you can email me at daniel@bytesafe.dev. We are here to help you.