Cheat sheet:

Angular security best practices

snyk

1. The “Angular way” safeguards you from XSS

Use Angular’s built-in curly braces string interpolation `({{ }})` to safely encode potentially dangerous characters inside a template expression. It allows you to use native syntax to escape potentially malicious user input that may put your web application at risk and lead to **Cross-site Scripting (XSS) vulnerabilities.**

2. Use innerHTML with caution

If you must add HTML in a component, bind it to use `[innerHTML]` which Angular will be cautious enough to sanitize for any malicious Cross-site Scripting (XSS) strings. This ensures data is interpreted as HTML in its context and sanitized. In the case of resource URLs (URLs that point to an executable resource), sanitizing is not possible!

3. Never use templates generated by concatenating user input

Never concatenate any input that is potentially provided by a useras a string to a template.

Avoid this pattern:

```
@Component({
 template: `
   <div> ... </div>
 ` + potentialUserInputString
})
```

Authors

@liran_tal
Liran TNode.js Security WG &
Developer Advocate at Snyk

@AnfibiaCreativa
Principal FrontEnd Software Engineer &
Architect at Netcentric

4. Never use native DOM APIs to interact with HTML elements

Never use native DOM APIs to interact with HTML elements on the page.

Avoid directly interacting with the DOM and instead use Angular templates, and Angular’s own APIs to manipulate the DOM or otherwise interact with it. As a general guideline, avoid the following:

- `node.appendChild();`
- using document to interact with the page
- using jQuery APIs

5. Avoid template engines on server-side templates

- Avoid template engines to create or add templates data on Angular server-side rendered.
- Don’t mix Angular’s own templating engine along with Node.js’s template engines like Handlebars, Pug, EJS or others.

6. Scan your Angular project for components which introduce security vulnerabilities

Angular components with millions of downloads are still vulnerable, and AngularJS has over 20 vulnerabilities to date - most are untracked with CVEs.

If you’re using npm audit already that’s a great start. To further increase security for CI integration and deployment monitoring:

- **Get a free Snyk scan token** to connect your GitHub or Bitbucket projects and receive a security webhook..
- Snyk will find vulnerabilities in 3rd party Angular and JavaScript modules in your package.json and opens fix pull requests so you can merge the upgraded version..