



HTML

CSS

JS

Result



```
<h1>
  <small>Reproduction of AngularJS vulnerability:</small><br />
  <code>(ng(Attr))Href</code> SVG image source sanitization bypass
</h1>

<section class="vulnerability-info-section">
  <h2>Vulnerability information</h2>
  <table class="vulnerability-info-table">
    <tr>
      <th>CVE:</th>
      <td><a href="https://www.cve.org/CVERecord?id=CVE-2025-0716">CVE-2025-0716</a></td>
    </tr>
    <tr>
      <th>Type:</th>
      <td>Image source sanitization bypass (a form of <a href="https://owasp.org/www-community/attacks/Content_Spoofing">Content Spoofing</a>)</td>
    </tr>
    <tr>
      <th>Affected versions:</th>
      <td>>=0.0.0</td>
    </tr>
  </table>
</section>
```

Reproduction of AngularJS vulnerability: (ng(Attr))Href SVG image source sanitization bypass

Vulnerability information

CVE: [CVE-2025-0716](#)

Type: Image source sanitization bypass (a form of [Content Spoofing](#))

Affected versions: >=0.0.0

Fixed in version: [AngularJS NES](#) v1.9.8, v1.5.24 and v1.4.16

Description: Setting an `<image>` SVG element's `href` attribute value via the `ngHref` and `ngAttrHref` directives or using [interpolation](#) is not subject to [image source sanitization](#). As a result, no restrictions are applied to the images that can be shown, which can also lead to a form of [Content Spoofing](#). Similarly, the app's performance and behavior can be negatively affected by using too large or slow-to-load images.

NOTE

Console

Assets

Comments

Export