```
<h1>
  <small>Reproduction of AngularJS vulnerability:</small><br />
  XSS via SCE resource URL 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-2026-11998">CVE-2026-11998</a></td>
    </tr>
    <tr>
      <th>Type:</th>
      <td><a href="https://owasp.org/www-community/attacks/xss">Cross-Site Scripting (XSS)</a></td>
    </tr>
    <tr>
      <th>Affected versions:</th>
      <td>>=1.2.0-rc.3</td>
    </tr>
```

Reproduction of AngularJS vulnerability:

XSS via SCE resource URL sanitization bypass

Vulnerability information

CVE:	CVE-2026-11998
Type:	Cross-Site Scripting (XSS)
Affected versions:	>=1.2.0-rc.3
Fixed in version:	AngularJS NES v1.9.12, v1.5.29 and v1.4.17
Description:	AngularJS uses the Strict Contextual Escaping (SCE) mode (via its `$sce` and `$sceDelegate` services) to only render trusted or safe values for certain security-sensitive contexts. One of these contexts is resource URLs, such as those used to load JavaScript scripts, `<iframe>` documents, etc. To designate certain URLs as safe resource URLs, one can specify a list of URL matchers using the `trustedResourceUrlList()` method. Each matcher can be either a string (with some wildcards) or a regular expression, which is supposed to be “matched against the **entire** *normalized / absolute URL* of the resource being tested (even when the RegExp did not have the `^` and