The following is a JavaScript security flaw:

<script>

 var str = "</script><script>alert('Pwned');</script>";

</script>

The browser ignores the fact that the<script> tags are inside a JavaScript String, invoking the alert()function.

The reason for this odd behavior is that the page gets rendered in various stages. First the HTML is parsed, and a render tree created. Only then, is the JavaScript actually executed. In the example above, the render tree see the <script> tags, and is oblivious to the fact that they’re inside a string; it has no concept of JavaScript. It strips these out, and evaluates the script nodes as usual with our injected message.

This behavior would be little more than a curiosity, were it not for the common pattern of injecting JSON into documents, say with ERB.

<script>

 var users = <%= @users.to\_json.html\_safe %>;

</script>

If you have the line above anywhere in your code, and @users includes some user submitted data, your application is vulnerable to a XSS attack.

**[SM-D01-R01]** If you’re using Rails, thwart this vulnerability by settingActiveSupport.escape\_html\_entities\_in\_json to true. The default isfalse.