Improper sanitize of SVG files during content upload ('Cross-site Scripting') in Sylius/Sylius

Low lchrusciel published GHSA-4qrp-27r3-66fj on Mar 14

Package

php sylius/sylius (Composer)

Affected versions

<1.9 <1.9.10 || >=1.10 <1.10.11 || >=1.11 <1.11.2

Patched versions

1.9.10, 1.10.11, 1.11.2

Description

Impact

There is a possibility to upload an SVG file containing XSS code in the admin panel. In order to perform an XSS attack, the file itself has to be opened in a new card (or loaded outside of the IMG tag). The problem applies both to the files opened on the admin panel and shop pages.

Patches

The issue is fixed in versions: 1.9.10, 1.10.11, 1.11.2, and above.

Workarounds

If there is a need to upload an SVG image type, on-upload sanitization has to be added. The way to achieve this is to require a library that will do the trick:

```
composer require enshrined/svg-sanitize
```

The second step is all about performing a file content sanitization before writing it to the filesystem. It can be done by overwriting the service:

```php
<?php

declare(strict_types=1);

namespace App\Uploader;

use enshrined\svgSanitize\Sanitizer;
use Gaufrette\Filesystem;
use Sylius\Component\Core\Generator\ImagePathGeneratorInterface;
use Sylius\Component\Core\Generator\UploadedImagePathGenerator;
use Sylius\Component\Core\Model\ImageInterface;
use Sylius\Component\Core\Uploader\ImageUploaderInterface;
use Symfony\Component\HttpFoundation\File\File;
use Webmozart\Assert\Assert;

final class ImageUploader implements ImageUploaderInterface
{
    private const MIME_SVG_XML = 'image/svg+xml';
    private const MIME_SVG = 'image/svg';

    /** @var Filesystem */
    protected $filesystem;

    /** @var ImagePathGeneratorInterface */
    protected $imagePathGenerator;

    /** @var Sanitizer */
    protected $sanitizer;

    public function __construct(
        Filesystem $filesystem,
        ?ImagePathGeneratorInterface $imagePathGenerator = null
    ) {
        $this->filesystem = $filesystem;

        if ($imagePathGenerator === null) {
            @trigger_error(sprintf(
                'Not passing an $imagePathGenerator to %s constructor is deprecated since Sylius
            ), \E_USER_DEPRECATED);
        }

        $this->imagePathGenerator = $imagePathGenerator ?? new UploadedImagePathGenerator();
        $this->sanitizer = new Sanitizer();
    }

    public function upload(ImageInterface $image): void
    {
        if (!$image->hasFile()) {
            return;
        }

        /** @var File $file */
        $file = $image->getFile();

        Assert::isInstanceOf($file, File::class);
```

```
        $fileContent = $this->sanitizeContent(file_get_contents($file->getPathname()), $file->ge

        if (null !== $image->getPath() && $this->has($image->getPath())) {
            $this->remove($image->getPath());
        }

        do {
            $path = $this->imagePathGenerator->generate($image);
        } while ($this->isAdBlockingProne($path) || $this->filesystem->has($path));

        $image->setPath($path);

        $this->filesystem->write($image->getPath(), $fileContent);
    }

    public function remove(string $path): bool
    {
        if ($this->filesystem->has($path)) {
            return $this->filesystem->delete($path);
        }

        return false;
    }

    protected function sanitizeContent(string $fileContent, string $mimeType): string
    {
        if (self::MIME_SVG_XML === $mimeType || self::MIME_SVG === $mimeType) {
            $fileContent = $this->sanitizer->sanitize($fileContent);
        }

        return $fileContent;
    }

    private function has(string $path): bool
    {
        return $this->filesystem->has($path);
    }

    /**
     * Will return true if the path is prone to be blocked by ad blockers
     */
    private function isAdBlockingProne(string $path): bool
    {
        return strpos($path, 'ad') !== false;
    }
}
```

After that, register service in the container:

```
services:
    sylius.image_uploader:
        class: App\Uploader\ImageUploader
```

```
        arguments:
            - '@gaufrette.sylius_image_filesystem'
            - '@Sylius\Component\Core\Generator\ImagePathGeneratorInterface'
```

For more information

If you have any questions or comments about this advisory:

- Open an issue in Sylius issues
- Email us at security@sylius.com

Severity

Low

CVE ID

CVE-2022-24749

Weaknesses

CWE-80 CWE-434