chamilo / **chamilo-lms** Public

Code | Issues 399 | Pull requests 13 | Discussions | Actions | Projects

Commit da671d6

AngelFQC committed 3 weeks ago · ✕ 1 / 6 · Verified

```
Security: Ensure SVG files are sanitized and properly served download

See advisory GHSA-273p-jw9w-3g22
```

master · v2.0.0 v2.0.0-RC.3

1 parent 7c4965e commit da671d6

1 file changed +22 -22 lines changed

Top

Filter files…

src/CoreBundle/Controller

ResourceController.php

1 file changed +22 -22 lines changed

Search within code

...oreBundle/Controller/ResourceController.php

```
@@ -602,6 +602,28 @@ private function processFile(Request $request, ResourceNode $resourceNode, Resou
602 602         // This covers files uploaded before the MIME-type allowlist was introduced.
603 603         $isSocialAttachment = 'social_post_attachments' === (string) $request->attributes->get('type');
604 604
+ 605         // SVG: sanitize before serving in any mode (view or download).
+ 606         // Glide is raster-only and cannot process SVG; sanitization strips embedded scripts regardless of how the file was stored.
+ 607         if ('image/svg+xml' === $mimeType) {
+ 608             $raw = $resourceNodeRepo->getResourceNodeFileContent($resourceNode, $resourceFile);
```

```
+             $content = (new SvgSanitizer())->sanitize((string) $raw);
+
+             if (false === $content || '' === $content) {
+                 throw new BadRequestHttpException('Invalid SVG file');
+             }
+
+             $response = new Response($content);
+             $dispositionMode = 'download' === $mode
+                 ? ResponseHeaderBag::DISPOSITION_ATTACHMENT
+                 : ResponseHeaderBag::DISPOSITION_INLINE;
+             $disposition = $response->headers->makeDisposition($dispositionMode, $fileName);
+             $response->headers->set('Content-Disposition', $disposition);
+             $response->headers->set('Content-Type', 'image/svg+xml');
+             $response->headers->set('X-Content-Type-Options', 'nosniff');
+
+             return $response;
+         }
+
605 627          switch ($mode) {
606 628              case 'download':
607 629                  $forceDownload = true;
    @@ -612,28 +634,6 @@ private function processFile(Request $request, ResourceNode $resourceNode, Resou
612 634              default:
613 635                  $forceDownload = false;
614 636
615     -                // SVG must not go through Glide (raster-only library); serve directly after sanitization.
616     -                // Sanitizing at serve time ensures scripts are stripped regardless of how the file was stored.
617     -                if ('image/svg+xml' === $mimeType) {
618     -                    $raw = $resourceNodeRepo->getResourceNodeFileContent($resourceNode, $resourceFile);
619     -                    $content = (new SvgSanitizer())->sanitize($raw);
620     -
621     -                    if (false === $content || '' === $content) {
622     -                        throw new BadRequestHttpException('Invalid SVG file');
623     -                    }
624     -
```

```
625       -                     $response = new Response($content);
626       -                     $disposition = $response->headers->makeDisposition(
627       -                         ResponseHeaderBag::DISPOSITION_INLINE,
628       -                         $fileName
629       -                     );
630       -                     $response->headers->set('Content-Disposition',
          $disposition);
631       -                     $response->headers->set('Content-Type', 'image/svg+xml');
632       -                     $response->headers->set('X-Content-Type-Options',
          'nosniff');
633       -
634       -                     return $response;
635       -                 }
636       -
637   637                 // If it's an image then send it to Glide.
638   638                 if (str_contains($mimeType, 'image')) {
639   639                     $glide = $this->getGlide();
```

Comments 0

Please sign in to comment.