chamilo / **chamilo-lms** Public

Code | Issues 415 | Pull requests 12 | Discussions | Actions | Projects

Commit e3790c5

AngelFQC committed on Mar 10 Verified

```
Security: Add URL safety checks to prevent SSRF attacks

Introduce `isUrlSafe` method to validate URLs, blocking private IP ranges, non-HTTP schemes,
and unresolvable hosts. Update `verifyUrl` and cURL configurations accordingly.

See advisory GHSA-q74c-mx8x-489h

Proposed by @elliSzAt

# Conflicts:
#       main/inc/lib/opengraph/OpenGraph.php
#       main/inc/lib/social.lib.php
```

master · v2.0.0-RC.3

1 parent 7df4cc1 commit e3790c5

1 file changed +47 -2 lines changed

Top

Filter files…

public/main/inc/lib

social.lib.php

1 file changed +47 -2 lines changed

Search within code

public/main/inc/lib/social.lib.php

```
@@ -11,6 +11,7 @@
11 11   use Chamilo\CoreBundle\Framework\Container;
12 12   use Chamilo\CourseBundle\Entity\CForumPost;
13 13   use Chamilo\CourseBundle\Entity\CForumThread;
   14 + use GuzzleHttp\Client;
14 15
```

```
15  16      /**
16  17       * Class SocialManager.
```

```
@@ -908,19 +909,63 @@ public static function formatWallMessages($messages)
```

```
908 909             return $data;
909 910         }
910 911
```

```
    912 +     /**
    913 +      * Check if a URL is safe to fetch server-side (not targeting internal resources).
    914 +      *
    915 +      * Blocks private/reserved IP ranges, non-HTTP schemes, and unresolvable hosts
    916 +      * to prevent SSRF attacks (CWE-918).
    917 +      */
    918 +     public static function isUrlSafe(string $url): bool
    919 +     {
    920 +         $parsed = parse_url($url);
    921 +
    922 +         // Allow only http and https schemes
    923 +         if (!isset($parsed['scheme']) || !in_array($parsed['scheme'], ['http', 'https'], true)) {
    924 +             return false;
    925 +         }
    926 +
    927 +         $host = $parsed['host'] ?? '';
    928 +         if (empty($host)) {
    929 +             return false;
    930 +         }
    931 +
    932 +         // Resolve hostname to IP
    933 +         $ip = gethostbyname($host);
    934 +         if ($ip === $host) {
    935 +             // DNS resolution failed
    936 +             return false;
    937 +         }
    938 +
    939 +         // Block private and reserved IP ranges
    940 +         if (false === filter_var(
    941 +                 $ip,
```

```
                                FILTER_VALIDATE_IP,
                                FILTER_FLAG_NO_PRIV_RANGE | FILTER_FLAG_NO_RES_RANGE
            )) {
            return false;
        }

        return true;
    }

    /**
     * verify if Url Exist - Using Curl.
     */
    public static function verifyUrl(string $uri): bool
    {
        if (!self::isUrlSafe($uri)) {
            return false;
        }

        $client = new Client();

        try {
            $response = $client->request('GET', $uri, [
-               'timeout' => 15,
+               'timeout' => 10,
                'verify' => false,
+               'allow_redirects' => ['max' => 3],
                'headers' => [
-                   'User-Agent' => $_SERVER['HTTP_USER_AGENT'],
+                   'User-Agent' => $_SERVER['HTTP_USER_AGENT'] ?? 'Chamilo',
                ],
            ]);
```

Comments 0

Please sign in to comment.