

chamilo / chamilo-lms Public

<> Code Issues 405 Pull requests 14 Discussions Actions Projects

Commit ea6b7b7



AngelFQC committed on Mar 9 · ✖ 1/3 · 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 [@e11iSzAt](#)

1.11.x · v1.11.38

1 parent [74fd1df](#) commit ea6b7b7

2 files changed

+51 -4 ●●●●● ●

Top



- ▼ main/inc/lib
 - ▼ opengraph
 - OpenGraph.php
 - social.lib.php



▼ main/inc/lib/opengraph/OpenGraph.php ⋮

```

@@ -52,11 +52,14 @@ static public function fetch($URI) {
52 52
53 53         curl_setopt($curl, CURLOPT_FAILONERROR, true);
54 54         curl_setopt($curl, CURLOPT_FOLLOWLOCATION, true);
55 +         curl_setopt($curl, CURLOPT_MAXREDIRS, 3);

```

```

56 + curl_setopt($curl, CURLOPT_PROTOCOLS, CURLPROTO_HTTP | CURLPROTO_HTTPS);
57 + curl_setopt($curl, CURLOPT_REDIR_PROTOCOLS, CURLPROTO_HTTP |
    CURLPROTO_HTTPS);
55 58 curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
56 - curl_setopt($curl, CURLOPT_TIMEOUT, 15);
59 + curl_setopt($curl, CURLOPT_TIMEOUT, 10);
57 60 curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
58 61 curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
59 - curl_setopt($curl, CURLOPT_USERAGENT, $_SERVER['HTTP_USER_AGENT']);
62 + curl_setopt($curl, CURLOPT_USERAGENT, $_SERVER['HTTP_USER_AGENT'] ??
    'Chamilo');
60 63
61 64 $response = curl_exec($curl);
62 65

```

main/inc/lib/social.lib.php

```

@@ -2111,19 +2111,63 @@ public static function
readContentWithOpenGraph(string $link): string

2111 2111     return $html;
2112 2112     }
2113 2113
2114 + /**
2115 +  * Check if a URL is safe to fetch server-side (not targeting internal
resources).
2116 +  *
2117 +  * Blocks private/reserved IP ranges, non-HTTP schemes, and unresolvable
hosts
2118 +  * to prevent SSRF attacks (CWE-918).
2119 +  */
2120 + public static function isUrlSafe(string $url): bool
2121 + {
2122 +     $parsed = parse_url($url);
2123 +
2124 +     // Allow only http and https schemes
2125 +     if (!isset($parsed['scheme']) || !in_array($parsed['scheme'],
['http', 'https'], true)) {
2126 +         return false;
2127 +     }
2128 +

```

```

2129 +     $host = $parsed['host'] ?? '';
2130 +     if (empty($host)) {
2131 +         return false;
2132 +     }
2133 +
2134 +     // Resolve hostname to IP
2135 +     $ip = gethostbyname($host);
2136 +     if ($ip === $host) {
2137 +         // DNS resolution failed
2138 +         return false;
2139 +     }
2140 +
2141 +     // Block private and reserved IP ranges
2142 +     if (false === filter_var(
2143 +         $ip,
2144 +         FILTER_VALIDATE_IP,
2145 +         FILTER_FLAG_NO_PRIV_RANGE | FILTER_FLAG_NO_RES_RANGE
2146 +     )) {
2147 +         return false;
2148 +     }
2149 +
2150 +     return true;
2151 + }
2152 +

```

```

2114 2153     /**
2115 2154     * verify if Url Exist - Using Curl.
2116 2155     */
2117 2156     public static function verifyUrl(string $uri): bool
2118 2157     {

```

```

2158 +         if (!self::isUrlSafe($uri)) {
2159 +             return false;
2160 +         }
2161 +

```


```

2119 2162         $client = new Client();
2120 2163
2121 2164         try {
2122 2165             $response = $client->request('GET', $uri, [
2123 -                 'timeout' => 15,
2166 +                 'timeout' => 10,
2124 2167                 'verify' => false,

```

2168	+	'allow_redirects' => ['max' => 3],
2125	2169	'headers' => [
2126	-	'User-Agent' => \$_SERVER['HTTP_USER_AGENT'],
2170	+	'User-Agent' => \$_SERVER['HTTP_USER_AGENT'] ?? 'Chamilo',
2127	2171],
2128	2172]);
2129	2173	
↓		

Comments 0



Please [sign in](#) to comment.