

chamilo / chamilo-lms Public

<> Code Issues 415 Pull requests 12 Discussions Actions Projects

# Commit 73ae629



christianbeeznest committed on Nov 29, 2025 · ✖ 3 / 7

Security: Restrict login redirect to same-origin URLs

master (#7122) · v2.0.0-beta.3 ··· v2.0.0-RC.1

1 parent [4564c09](#) commit 73ae629

3 files changed +27 -13 lines changed

↑ Top

🔍 Filter files...

- ✓ assets/vue
  - ✓ components
    - Login.vue
  - ✓ composables/auth
    - login.js
- ✓ src/CoreBundle/EventListener
  - ExceptionListener.php

3 files changed +27 -13 lines changed

🔍 Search within code

assets/vue/components/Login.vue

```

  ↑   @@ -100,8 +100,12 @@ const isInIframe = window.self !== window.top
100  100      if (isInIframe) {
101  101          try {
102  102              const parentUrl = window.top.location.href
103  -         window.top.location.href = "/login?redirect=" +
           encodeURIComponent(parentUrl)
103  +         const parent = new URL(parentUrl)

```

```

104 + // Only keep path + query + hash so redirect stays internal
105 + const redirectPath = parent.pathname + parent.search + parent.hash
106 + window.top.location.href = "/login?redirect=" +
    encodeURIComponent(redirectPath)
104 107 } catch (e) {
108 + // Cross-origin or other error: just go to login without redirect
105 109 window.top.location.href = "/login"
106 110 }
107 111 }

```

assets/vue/composables/auth/login.js

```

@@ -52,13 +52,13 @@ function normalizeRedirectUrl(rawRedirect) {
52 52 try {
53 53     const currentOrigin = window.location.origin
54 54
55 - // root-relative path ("/resources/pages/edit?id=...")
55 + // Root-relative path ("/resources/pages/edit?id=...")
56 56     if (rawRedirect.startsWith("/")) {
57 57         const url = new URL(rawRedirect, currentOrigin)
58 58         return url.pathname + url.search + url.hash
59 59     }
60 60
61 - // absolute URL - validate protocol first
61 + // Absolute URL - validate protocol first
62 62     if (!isValidHttpUrl(rawRedirect)) {
63 63         return null
64 64     }
@@ -102,10 +102,11 @@ export function useLogin() {
102 102     totp,
103 103     }
104 104
105 - // Add returnUrl if exists in query param
106 - const returnUrl = route.query.redirect?.toString() || null
107 - if (returnUrl) {
108 -     payload.returnUrl = returnUrl
105 + // Add returnUrl if exists in query param, but sanitize it first
106 + const rawReturnUrl = route.query.redirect?.toString() || null

```

```

107 +     const safeReturnUrl = rawReturnUrl ? normalizeRedirectUrl(rawReturnUrl) :
      null
108 +     if (safeReturnUrl) {
109 +         payload.returnUrl = safeReturnUrl
109 110     }
110 111
111 112     const responseData =
@@ -122,7 +123,8 @@ export function useLogin() {
122 123         // If backend forces password rotation, still apply locale before
      redirect
123 124         if (responseData.rotate_password && responseData.redirect) {
124 125             applyUserLocale(responseData)
125 -         window.location.href = responseData.redirect
126 +         const safeRedirect =
      normalizeRedirectUrl(responseData.redirect.toString())
127 +         window.location.href = safeRedirect || "/"
126 128         return { success: true, rotate: true }
127 129     }
128 130
@@ -135,8 +137,10 @@ export function useLogin() {
135 137         // Terms and conditions redirect (apply locale before navigating)
136 138         if (responseData.load_terms && responseData.redirect) {
137 139             applyUserLocale(responseData)
138 -         window.location.href = responseData.redirect
139 -         return { success: true, redirect: responseData.redirect }
140 +         const safeRedirect =
      normalizeRedirectUrl(responseData.redirect.toString())
141 +         const target = safeRedirect || "/"
142 +         window.location.href = target
143 +         return { success: true, redirect: target }
140 144     }
141 145
142 146         // External redirect param (apply locale before navigating)
@@ -147,7 +151,6 @@ export function useLogin() {
147 151         const safeRedirect = normalizeRedirectUrl(rawRedirect)
148 152
149 153         if (safeRedirect) {
150 -         // Keep query params (e.g. "?id=/api/pages/6") intact on same origin
151 154         // Full reload here is intentional so the full app shell is rebuilt.
152 155         window.location.href = safeRedirect

```

```

153 156         return { success: true }
@@ -157,7 +160,8 @@ export function useLogin() {
157 160         // Fallback backend redirect (apply locale before navigating)
158 161         if (responseData.redirect) {
159 162             applyUserLocale(responseData)
160 -         window.location.href = responseData.redirect
163 +         const safeRedirect =
164 +             normalizeRedirectUrl(responseData.redirect.toString())
161 165         return { success: true }
162 166     }
163 167

```

...eBundle/EventListener/ExceptionListener.php

```

@@ -52,9 +52,15 @@ public function __invoke(ExceptionEvent $event): void
52 52         ) {
53 53             // If no token (not logged in), redirect to login with "redirect"
54 54             back param
54 54             if (null === $this->tokenStorage->getToken()) {
55 +             // Use only a relative path (path + query) for the redirect
56 +             parameter
56 +             $redirectPath = $request->getRequestUri();
57 +             if (!\is_string($redirectPath) || '' === $redirectPath) {
58 +                 $redirectPath = '/';
59 +             }
60 +
55 61             $loginUrl = $this->router->generate(
56 62                 'login',
57 -                 ['redirect' => $request->getSchemeAndHttpHost().$request-
58 +                 ['redirect' => $redirectPath],
59 +                 UrlGeneratorInterface::ABSOLUTE_URL
60 66                 $event->setResponse(new RedirectResponse($loginUrl));

```

Comments 0



Please [sign in](#) to comment.