chamilo / **chamilo-lms** Public

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

Commit `078d7e5`

ywarnier committed 3 weeks ago · ✕ 1 / 6

```
Security: Fix Weak Password Recovery Mechanism - refs GHSA-f27g-66gq-g7v2
```

`master` · v2.0.0-RC.3

1 parent c1eeb23 commit 078d7e5

1 file changed +63 -16 lines changed

Top

Filter files...

public/main/inc/lib

login.lib.php

1 file changed +63 -16 lines changed

Search within code

public/main/inc/lib/login.lib.php

```
@@ -39,9 +39,9 @@ public static function get_user_account_list($user, $reset = false, $by_username
39 39
40 40            if ($reset) {
41 41                if ($by_username) {
42    -                 $secret_word = self::get_secret_word($user['email']);
43    -                 if ($reset) {
44    -                     $reset_link = $portal_url."main/auth/lostPassword.php?reset=".$secret_word."&id=".$user['id'];
   42 +                 $token = self::generateResetToken($user['id']);
   43 +                 if ($token) {
   44 +                     $reset_link = $portal_url."main/auth/lostPassword.php?reset=".$token."&id=".$user['id'];
45 45                      $reset_link = Display::url($reset_link, $reset_link);
```

```
46  46                      } else {
47  47                          $reset_link = get_lang('Pass')." : $user[password]";
    @@ -55,9 +55,9 @@ public static function get_user_account_list($user, $reset = false, $by_username
55  55                      }
56  56                  } else {
57  57                      foreach ($user as $this_user) {
58      -                       $secret_word = self::get_secret_word($this_user['email']);
59      -                       if ($reset) {
60      -                           $reset_link = $portal_url."main/auth/lostPassword.php?reset=".$secret_word."&id=".$this_user['id'];
    58  +                       $token = self::generateResetToken($this_user['id']);
    59  +                       if ($token) {
    60  +                           $reset_link = $portal_url."main/auth/lostPassword.php?reset=".$token."&id=".$this_user['id'];
61  61                          $reset_link = Display::url($reset_link, $reset_link);
62  62                      } else {
63  63                          $reset_link = get_lang('Pass')." : $this_user[password]";
    @@ -237,18 +237,44 @@ public static function sendResetEmail(User $user)
237 237             Display::addFlash(Display::return_message(get_lang('Check your e-mail and follow the instructions.')));
238 238         }
239 239
    240 +     /**
    241 +      * Generates a cryptographically secure reset token for the given user,
    242 +      * stores it in the user's confirmationToken field with a timestamp,
    243 +      * and returns the token.
    244 +      *
    245 +      * @param int $userId
    246 +      *
    247 +      * @return string|null the generated token, or null on failure
    248 +      */
    249 +     public static function generateResetToken($userId)
    250 +     {
    251 +         $userEntity = api_get_user_entity((int) $userId);
    252 +         if (!$userEntity) {
    253 +             return null;
    254 +         }
```

```
      255 +
      256 +         $token = api_get_unique_id();
      257 +         $userEntity->setConfirmationToken($token);
      258 +         $userEntity->setPasswordRequestedAt(new \DateTime());
      259 +
      260 +         Database::getManager()->persist($userEntity);
      261 +         Database::getManager()->flush();
      262 +
      263 +         return $token;
      264 +     }
      265 +
240   266          /**
241   267           * Gets the secret word.
242   268           *
243       -       * @author Olivier Cauberghe <olivier.cauberghe@UGent.be>, Ghent University
      269 +       * @deprecated Use generateResetToken() instead
244   270           */
245   271          public static function get_secret_word($add)
246   272          {
247       -           return $secret_word = sha1($add);
      273 +           return sha1($add);
248   274          }
249   275
250   276          /**
251       -       * Resets a password.
      277 +       * Resets a password using a secure, time-limited token.
252   278           *
253   279           * @author Olivier Cauberghe <olivier.cauberghe@UGent.be>, Ghent University
254   280           */
      @@ -276,16 +302,37 @@ public static function reset_password($secret, $id,
      $by_username = false)
276   302                   return get_lang('Could not reset password');
277   303               }
278   304
279       -           if (self::get_secret_word($user['email']) == $secret) {
280       -               // OK, secret word is good. Now change password and mail it.
281       -               $user['password'] = api_generate_password();
282       -
283       -               UserManager::updatePassword($userEntity->getId(),
      $user['password']);
```

```
                305 +         $storedToken = $userEntity->getConfirmationToken();
284             306
285             -             return self::send_password_to_user($user, $by_username);
286             -         } else {
                307 +         if (empty($storedToken) || !hash_equals($storedToken, $secret)) {
287             308               return get_lang('You are not allowed to see this page. Either your
                      connection has expired or you are trying to access a page for which you do not
                      have the sufficient privileges.');
288             309           }
                310 +
                311 +         $ttl = (int) api_get_setting('user_reset_password_token_limit');
                312 +         if (empty($ttl)) {
                313 +             $ttl = 3600;
                314 +         }
                315 +
                316 +         if (!$userEntity->isPasswordRequestNonExpired($ttl)) {
                317 +             $userEntity->setConfirmationToken(null);
                318 +             $userEntity->setPasswordRequestedAt(null);
                319 +             Database::getManager()->persist($userEntity);
                320 +             Database::getManager()->flush();
                321 +
                322 +             return get_lang('Link expired, please try again.');
                323 +         }
                324 +
                325 +         // Token is valid. Change password and mail it.
                326 +         $user['password'] = api_generate_password();
                327 +         UserManager::updatePassword($userEntity->getId(), $user['password']);
                328 +
                329 +         // Invalidate the token so it cannot be reused.
                330 +         $userEntity->setConfirmationToken(null);
                331 +         $userEntity->setPasswordRequestedAt(null);
                332 +         Database::getManager()->persist($userEntity);
                333 +         Database::getManager()->flush();
                334 +
                335 +         return self::send_password_to_user($user, $by_username);
289             336       }
290             337
291             338       /**
```

Comments 0

Please sign in to comment.