

✓ Nimbus skin: XSS via the "Advertise" link interface messages (CVE-2022-29907)

Actions

✓ Closed, Resolved

Public

SECURITY

Assigned To

ashley

Authored By

ashley

2022-04-25 16:10:35 (UTC+0)

Tags

Security

Nimbus (Backlog)

Vuln-XSS

SecTeam-Processed (Completed)

Referenced Files

None

Subscribers

Aklapper

ashley

Bawolff

lcawte

sbassett

Description

When `[[MediaWiki:Nimbus-advertise-url]]` exists and is not disabled (i.e. its value is not `-`), an "Advertise" link (`[[MediaWiki:Nimbus-advertise]]`) will show up in the skin's footer, and the target of this link is the URL configured in `[[MediaWiki:Nimbus-advertise-url]]`.

Unfortunately both messages are currently vulnerable to the easiest possible XSS you can think of: `">`

```
<script>alert('XSS')</script>
```

Here's a quick, tested patch to fix that:

```
@@ -76,10 +104,10 @@ class SkinNimbus extends SkinTemplate {  
    */
```


```

function advertiseLink() {
    $link = '';
-    $adMsg = wfMessage( 'nimbus-advertise-url' )->inContentLanguage();
-    if ( !$adMsg->isDisabled() ) {
+    $adMsg = $this->msg( 'nimbus-advertise-url' )->inContentLanguage();
+    if ( !$adMsg->isDisabled() && filter_var( $adMsg->text(), FILTER_VALIDATE_URL ) ) {
        $link = '<a href="' . $adMsg->text() . '"' rel="nofollow">' .
-            wfMessage( 'nimbus-advertise' )->plain() . '</a>';
+            $this->msg( 'nimbus-advertise' )->escaped() . '</a>';
    }
    return $link;
}

```

(The RequestContext-ification is not related to the security aspect, but I figured I might as well do that while I'm editing this portion of the code. Also, line numbers etc. are probably off, given that my local copy of Nimbus has a lot of uncommitted changes.)

Details

Project	Subject
 mediawiki/skins/Nimbus	[SECURITY] Fix XSS in the "Advertise" link messages
Customize query in Gerrit	

Related Objects


Mentions

Mentioned In

~~T305209: Write and send supplementary release announcement for extensions and skins with security patches (1.35.7/1.37.3/1.38.2)~~

Mentioned Here

~~T305209: Write and send supplementary release announcement for extensions and skins with security patches (1.35.7/1.37.3/1.38.2)~~

 **ashley** created this task. 2022-04-25 16:10:35 (UTC+0)

  Restricted Application added a subscriber: **Aklapper**. · [View Herald Transcript](#) 2022-04-25 16:10:36 (UTC+0)

 **ashley** claimed this task. 2022-04-25 16:11:09 (UTC+0)

 **ashley** added projects: **Nimbus**, **Vuln-XSS**.



 **ashley** added a subscriber: **Bawolff**. 2022-04-26 15:01:37 (UTC+0)

 **Bawolff** added a comment. Edited · 2022-04-26 15:18:28 (UTC+0)

\$adMsg should probably also be ->escaped() as well for the href. FILTER_VALIDATE_URL allows urls like `https://example.com/"><script>alert(1)</script>`. Additionally, if you assume the message is not


trusted, you can have javascript scheme urls to get xss e.g. `javascript://%0aalert(1)` passes `FILTER_VALIDATE_URL`.

You could potentially use `$url = Sanitizer::validateAttributes(['href' => $adMsg->text()])` `['href'] ?? false;` and then `htmlspecialchars` the result if its not false (I did not test that).

 **ashley** added a comment. 2022-04-27 13:16:35 (UTC+0) 


In **T306815#7881305**, **@Bawolff** wrote:

\$adMsg should probably also be `->escaped()` as well for the href. `FILTER_VALIDATE_URL` allows urls like `https://example.com/"><script>alert(1)</script>`. Additionally, if you assume the message is not trusted, you can have javascript scheme urls to get xss e.g. `javascript://%0aalert(1)` passes `FILTER_VALIDATE_URL`.

Thank you for this, I learned something new! 

You could potentially use `$url = Sanitizer::validateAttributes(['href' => $adMsg->text()])` `['href'] ?? false;` and then `htmlspecialchars` the result if its not false (I did not test that).

This seems to have done the trick for good, though I had to pass in `['href']` as a 2nd param to the `validateAttributes` method because it apparently uses a whitelist-based approach, so you have to tell it explicitly what attributes are OK and what are not.

Here's the current patch that I have, would definitely appreciate your thoughts on it! 

```
@@ -76,10 +104,13 @@ class SkinNimbus extends SkinTemplate {
    */
    function advertiseLink() {
        $link = '';
-        $adMsg = wfMessage( 'nimbus-advertise-url' )->inContentLanguage();
+        $adMsg = $this->msg( 'nimbus-advertise-url' )->inContentLanguage();
        if ( !$adMsg->isDisabled() ) {
-            $link = '<a href="' . $adMsg->text() . '" rel="nofollow">' .
-                wfMessage( 'nimbus-advertise' )->plain() . '</a>';
+            $url = Sanitizer::validateAttributes( [ 'href' => $adMsg->text() ], [ 'href'
+ ] )['href'] ?? false;
+            if ( $url ) {
+                $link = '<a href="' . htmlspecialchars( $url, ENT_QUOTES ) . '"
+ rel="nofollow">' .
+                $this->msg( 'nimbus-advertise' )->escaped() . '</a>';
+            }
        }
        return $link;
    }
}
```

 **Bawolff** added a comment. 2022-04-27 13:21:07 (UTC+0) 

Looks good :)

✓ **ashley** closed this task as *Resolved*. 2022-04-27 14:16:46 (UTC+0) ▼

Many thanks for your help with this, [@Bawolff](#), I really appreciate it! Not only did we get the XSS fixed *properly*, I learned a thing or two while doing it. 😊

🔒 **Legoktm** changed the visibility from "**Custom Policy**" to "Public (No Login Required)". 2022-04-27 18:26:00 (UTC+0)

🔒 **Legoktm** changed the edit policy from "**Custom Policy**" to "All Users".

🔗 **sbassett** mentioned this in ~~**T305209: Write and send supplementary release announcement for extensions and skins with security patches (1.35.7/1.37.3/1.38.2)**~~. 2022-04-27 19:29:14 (UTC+0) ▼

🔗 **sbassett** edited projects, added **SecTeam-Processed**; removed **Security-Team**.

👤 **sbassett** added a subscriber: **sbassett**.

Thanks, all. Tracking for the next supplemental security release: [T305209](#).

✎ **Mstyles** renamed this task from *Nimbus skin: XSS via the "Advertise" link interface messages* to *Nimbus skin: XSS via the "Advertise" link interface messages (CVE-2022-29907)*. 2022-07-06 17:53:47 (UTC+0)