Written by Antonio
on March 27, 2019

Analyzing PHPKB v9: Part two

This article has been split into three parts; other parts can be found below:

Part 1: Part 1

Part 3: Part 3

UPDATE: the vulnerabilities have been fixed in PHPKB v9 P1-202005.

Reflected Cross-Site Scripting when editing a custom field (CVE-2020-10462)

Exploitable by: Superuser

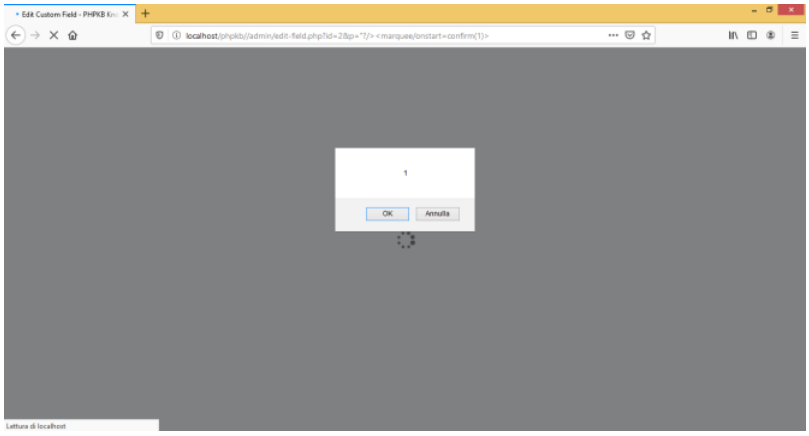
Vulnerable file: admin/edit-field.php

Steps required in order to find the injection point:

- log-in as a Superuser;
- go under “articles” and then on “custom fields”;
- click on “add new”, fill random parameters and click on “save custom field”;
- select the “custom field” and click on “edit”.

The vulnerability lies in the &p= parameter; proof of concept:

```
[PHPKB]/admin/edit-field.php?id=2&p="'/><marquee/onstart=confirm(1)>
```



Reflected Cross-Site Scripting when editing a template (CVE-2020-10463)

Exploitable by: Superuser

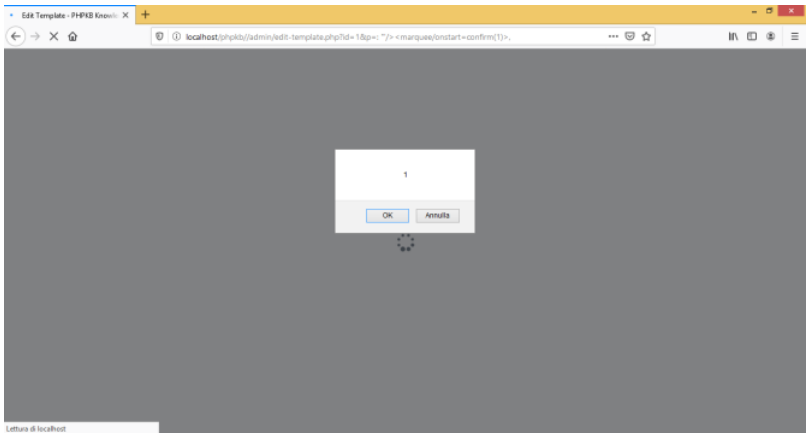
Vulnerable file: admin/edit-template.php

Steps required in order to find the injection point:

- log-in as a Superuser;
- go under “articles” and then on “article templates”;
- click on “add new” fill random parameters and click on “save article template”;
- select the “article template” and click on “edit”.

The vulnerability lies in the &p= parameter; proof of concept:

```
[PHPKB]/admin/edit-template.php?id=1&p="'/><marquee/onstart=confirm(1)>
```



Reflected Cross-Site Scripting when editing an article (CVE-2020-10464)

Exploitable by: Superuser/Editor

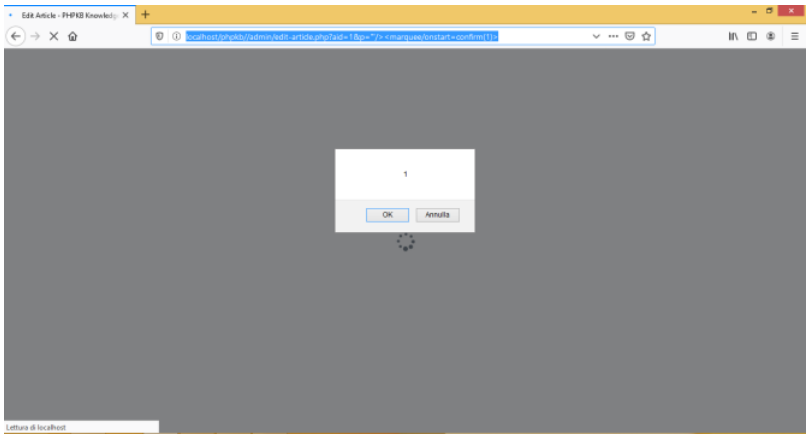
Vulnerable file: admin/edit-article.php

Steps required in order to find the injection point:

- log-in as a Superuser/Editor;
- go under “articles” and then on “add article”;
- fill random parameters and click on “save article”;
- select the “article” and click on “edit”.

The vulnerability lies in the &p= parameter; proof of concept:

```
[PHPKB]/admin/edit-article.php?aid=1&p="'/><marquee/onstart=confirm(1)>
```



Reflected Cross-Site Scripting when editing a category (CVE-2020-10465)

Exploitable by: Superuser

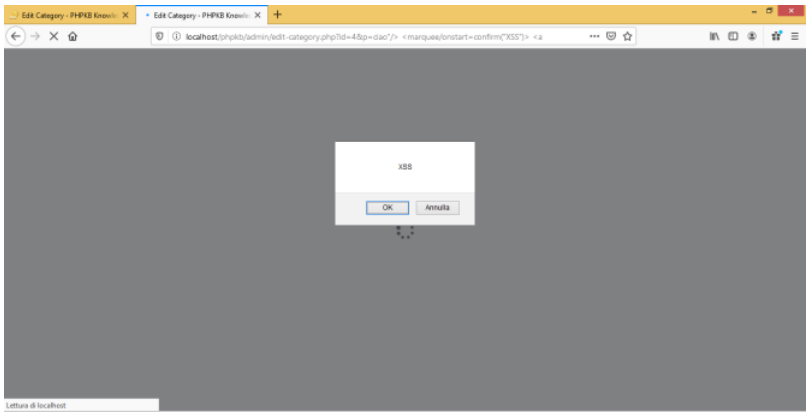
Vulnerable file: admin/edit-category.php

Steps required in order to find the injection point:

- log-in as a Superuser;
- go under “categories” and click on “add new”;
- fill random parameters and click “save category”;
- click on manage categories”;
- slect the “category” and click on “edit”.

The vulnerability lies in the &p= parameter; proof of concept:

```
[PHPKB]/admin/edit-category.php?id=1&p=ciao"/><marquee/onstart=confirm("1")>
```



Reflected Cross-Site scripting when editing a glossary term (CVE-2020-10466)

Exploitable by: Superuser

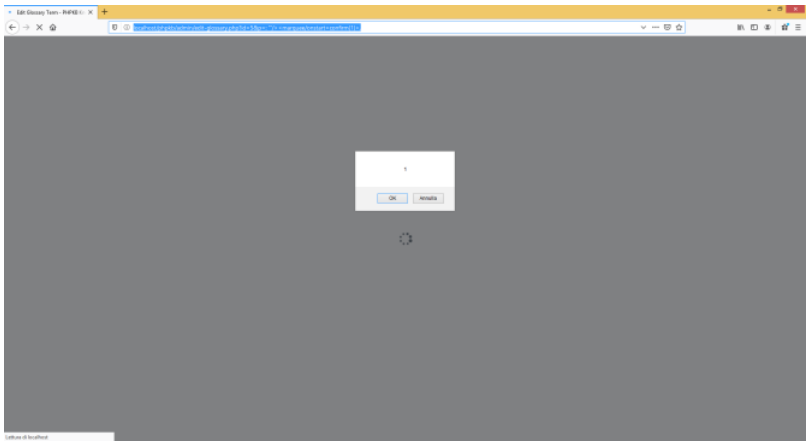
Vulnerable file: admin/edit-glossary.php

Steps required in order to find the injection point:

- log-in as a Superuser;
- go under “glossary” and click on “add new”;
- fill random parameters and click “save”;
- click on “manage”;
- click on “edit”.

The vulnerability lies in the &p= parameter; proof of concept:

```
[PHPKB]/admin/edit-glossary.php?id=5&p= "'/><marquee/onstart=confirm(1)>
```



Reflected Cross-Site Scripting when editing a comment (CVE-2020-10467)

Exploitable by: Superuser

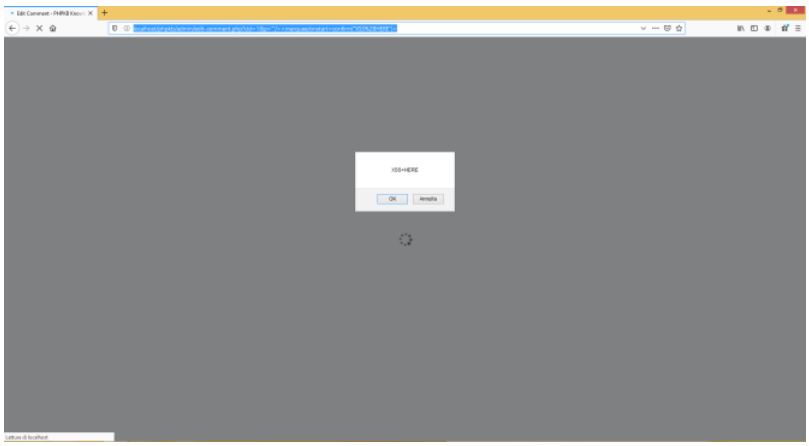
Vulnerable file: admin/edit-comment.php

Steps required in order to find the injection point:

- post a comment on a random article;
- log-in as a Superuser;
- go under “comments”;
- click on “pending”;
- select the “comment” and click on “edit”.

The vulnerability lies in the &p= parameter; proof of concept:

```
[PHPKB]/admin/edit-comment.php?cid=1&p="'/><marquee/onstart=confirm("XSS+HERE")>
```



Reflected Cross-Site Scripting when editing a news article (CVE-2020-10468)

Exploitable by: Superuser

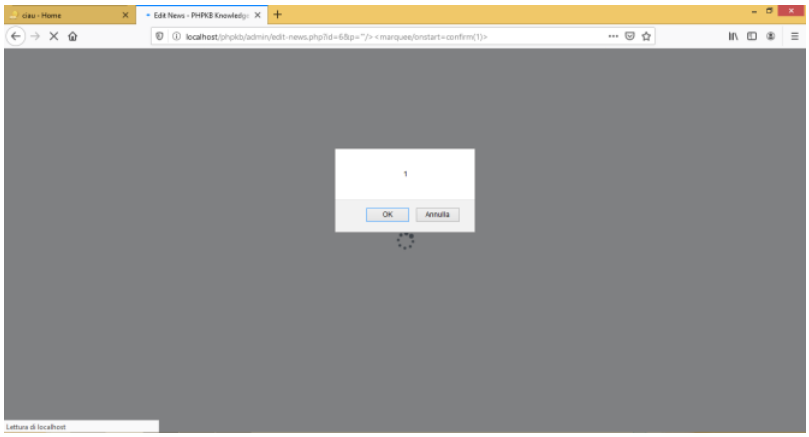
Vulnerable file: admin/edit-news.php

Steps required in order to find the injection point:

- log-in a Superuser;
- go under “news” and click on “add new”;
- fill random parameters and click on “save”;
- click on “manage”;
- click on “edit”.

The vulnerability lies in the &p= parameter; proof of concept:

```
[PHPKB]/admin/edit-news.php?id=6&p="'/><marquee/onstart=confirm(1)>
```



Reflected Cross-Site Scripting when editing a department (CVE-2020-10469)

Exploitable by: Superuser

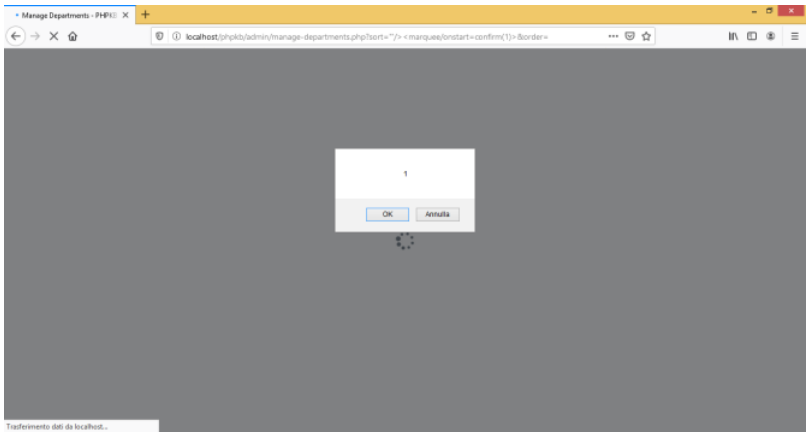
Vulnerable file: admin/manage-departments.php

Steps required in order to find the injection point:

- log-in as a Superuser;
- go under “tickets” and click on “departments”;
- fill random parameters and click on “save”;
- click on “edit”.

The vulnerability lies in the &sort= parameter; proof of concept:

```
[PHPKB]/admin/manage-departments.php?sort="'/><marquee/onstart=confirm(1)>&order=
```



Reflected Cross-Site Scripting when sorting custom fields (CVE-2020-10470)

Exploitable by: Superuser

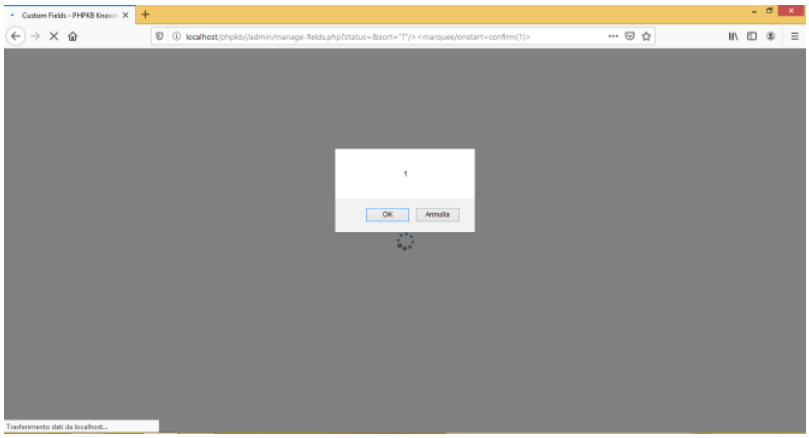
Vulnerable file: admin/manage-fields.php

Steps required in order to find the injection point:

- log-in as a Superuser;
- go under “articles” and click on “custom fields”;
- add a new “field” and then click on the dropbox, used to show 10 or 25 or 50 or 100 results.

The vulnerability lies in the &sort= parameter; proof of concept:

```
[PHPKB]/admin/manage-fields.php?sort="'/><marquee/onstart=confirm(1)>&order=
```



Reflected Cross-Site Scripting when sorting articles (CVE-2020-10471)

Exploitable by: Superuser

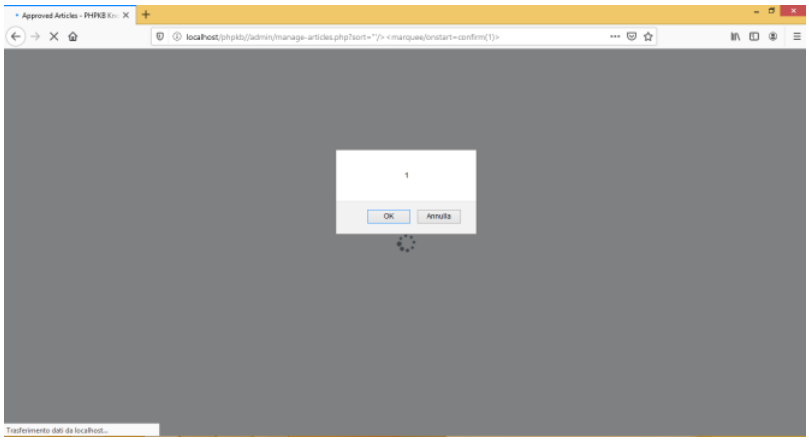
Vulnerable file: admin/manage-articles.php

Steps required in order to find the injection point:

- log-in as a Superuser;
- go under “articles” and click on “my articles”;
- add a new “article” and then click on the dropbox, used to show 10 or 25 or 50 or 100 results.

The vulnerability lies in the &sort= parameter; proof of concept:

```
[PHPKB]/admin/manage-articles.php?sort="'/><marquee/onstart=confirm(1)>
```



Reflected Cross-Site Scripting when sorting templates (CVE-2020-10472)

Exploitable by: Superuser

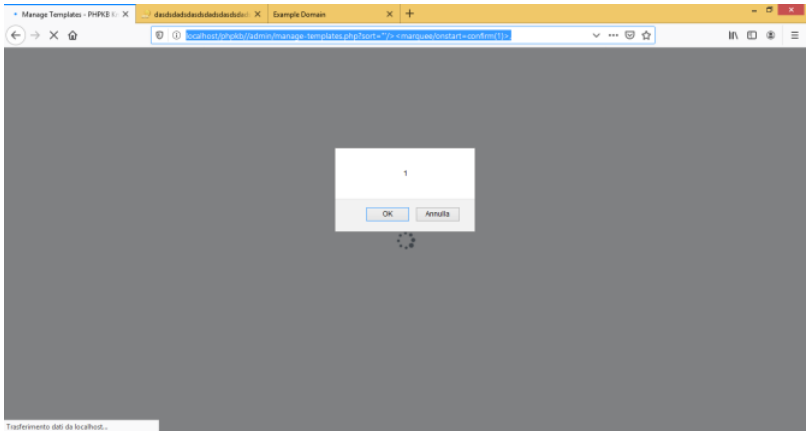
Vulnerable file: admin/manage-templates.php

Steps required in order to find the injection point:

- log-in as a Superuser;
- go under “articles” and click on “article templates”;
- add a new field and then click on the dropbox, used to show 10 or 25 or 50 or 100 results.

The vulnerability lies in the &sort= parameter; proof of concept:

```
[PHPKB]/admin/manage-templates.php?sort="'/><marquee/onstart=confirm(1)>
```



Reflected Cross-Site Scripting when deleting a category (CVE-2020-10473)

Exploitable by: Superuser

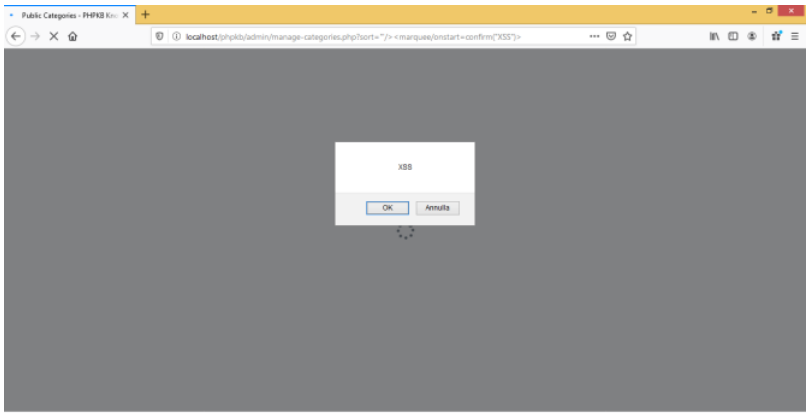
Vulnerable file: admin/manage-categories.php

Steps required in order to find the injection point:

- log-in as a Superuser;
- go under “categories” and click on “add new”;
- fill random parameters and click on “save category”;
- click on “manage categories”;
- click on “delete category”.

The vulnerability lies in the &sort= parameter; proof of concept:

```
[PHPKB]/admin/manage-categories.php?sort="'/>
<marquee/onstart=confirm("XSS")>&order=&status=public&action=status&status=public&id=1&action=delete
```



Reflected Cross-Site Scripting when deleting a comment (CVE-2020-10474)

Exploitable by: Superuser

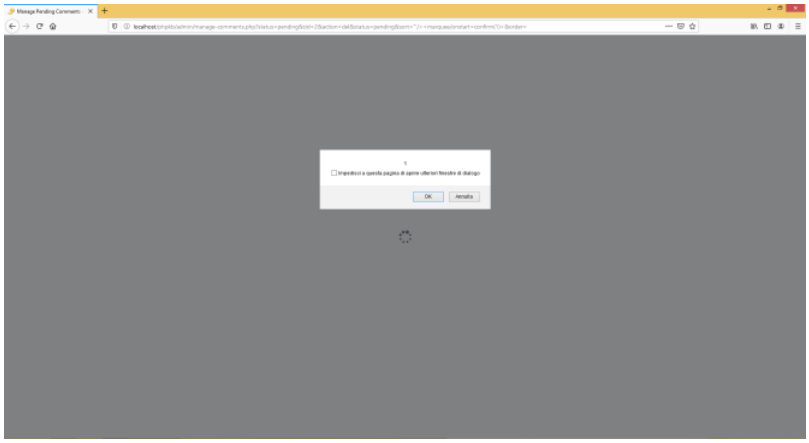
Vulnerable file: admin/manage-comments.php

Steps required in order to find the injection point:

- post a comment on a random article;
- log-in as a Superuser;
- go under “comments”;
- click on “pending”;
- select the “comment” and click on “delete”.

The vulnerability lies in the &sort= parameter; proof of concept:

```
[PHPKB]/admin/manage-comments.php?status=pending&cid=2&action=del&status=pending&sort="'/>
<marquee/onstart=confirm(1)>&order=
```



Reflected Cross-Site Scripting when deleting a ticket (CVE-2020-10475)

Exploitable by: Superuser

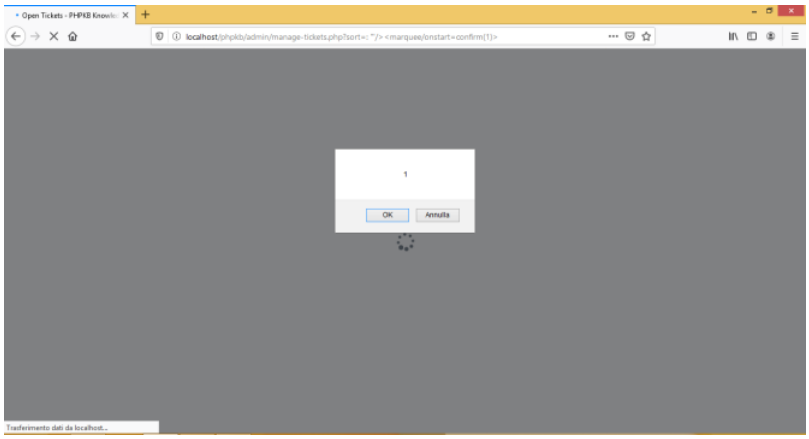
Vulnerable file: admin/manage-tickets.php

Steps required in order to find the injection point:

- ask a question on the PHPKB website;
- log-in as a Superuser;
- go under “tickets” and click on “delete”.

The vulnerability lies in the &sort= parameter; proof of concept:

```
[PHPKB]/admin/manage-tickets.php?sort="'/><marquee/onstart=confirm(1)>
```



Reflected Cross-Site Scripting when editing a glossary term #2 (CVE-2020-10476)

Exploitable by: Superuser

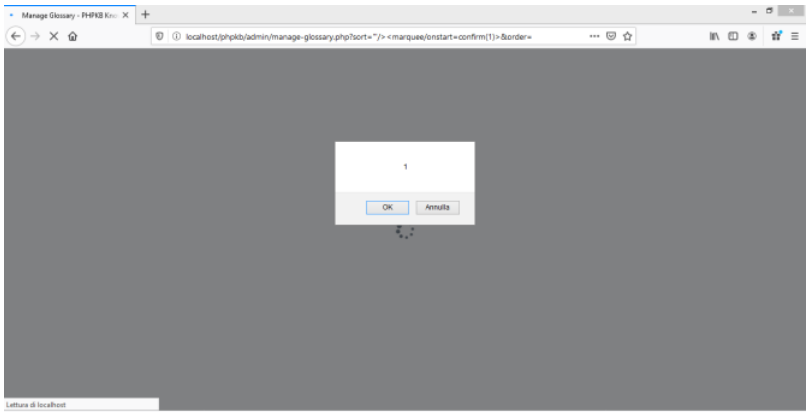
Vulnerable file: manage-glossary.php

Steps required in order to find the injection point:

- log-in as a Superuser;
- go under “glossary” and click on “add new”;
- fill random parameters and click “save”;
- click on “edit”;
- click on “save”.

The vulnerability lies in the &sort= parameter; proof of concept:

```
[PHPKB]/admin/manage-glossary.php?sort="'/><marquee/onstart=confirm(1)>&order=
```



Reflected Cross-Site Scripting when editing a news article (CVE-2020-10477)

Exploitable by: Superuser

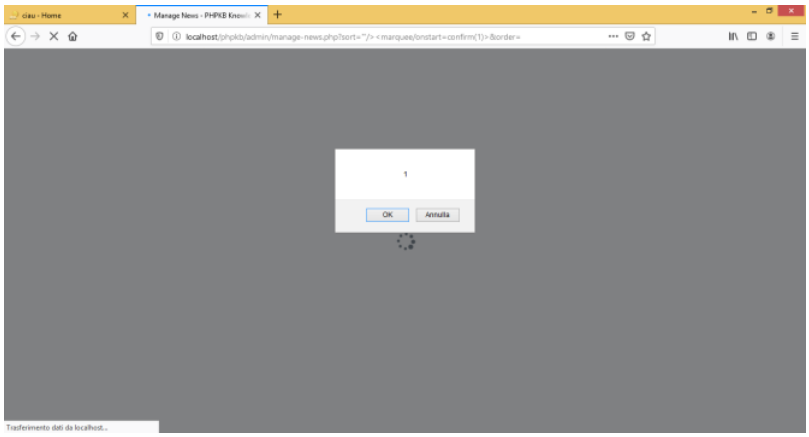
Vulnerable file: admin/manage-news.php

Steps required in order to find the injection point:

- log-in as a Superuser;
- go under “news” and click on “add new”;
- fill random parameters and click “save”;
- click on “edit”;
- click on “save”.

The vulnerability lies in the &sort= parameter; proof of concept:

```
[PHPKB]/admin/manage-news.php?sort="'/><marquee/onstart=confirm(1)>&order=
```



That's all for part two!

←

Top

→