Title:https://gitee.com/hnaoyun/PbootCMS website building system has a cross-site scripting XSS vulnerability.

BUG_Author: @hnaoyun

Affected Version: PbootCMS <v3.2.5

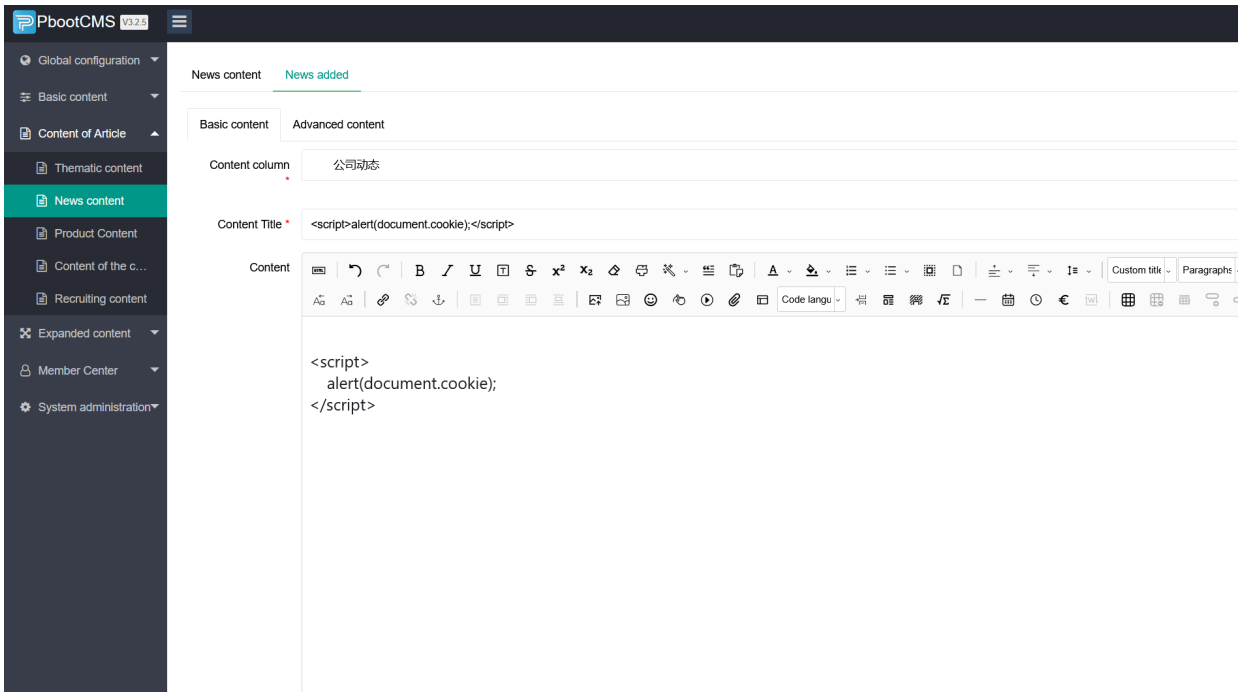
Vendor: PbootCMS: PbootCMS是全新内核且永久开源免费的PHP企业网站开发建设管理系统，是一套高效、简洁、强悍的可免费商用的PHP CMS源码，能够满足各类企业网站开发建设的需要。系统采用简单到想哭的模板标签，只要懂HTML就可快速开发企业网站。官方提供了大量网站模板免费下载和使用，将致力于为广大开发者和企业提供最佳的网站开发建设解决方案。(gitee.com)

Vulnerability Files:

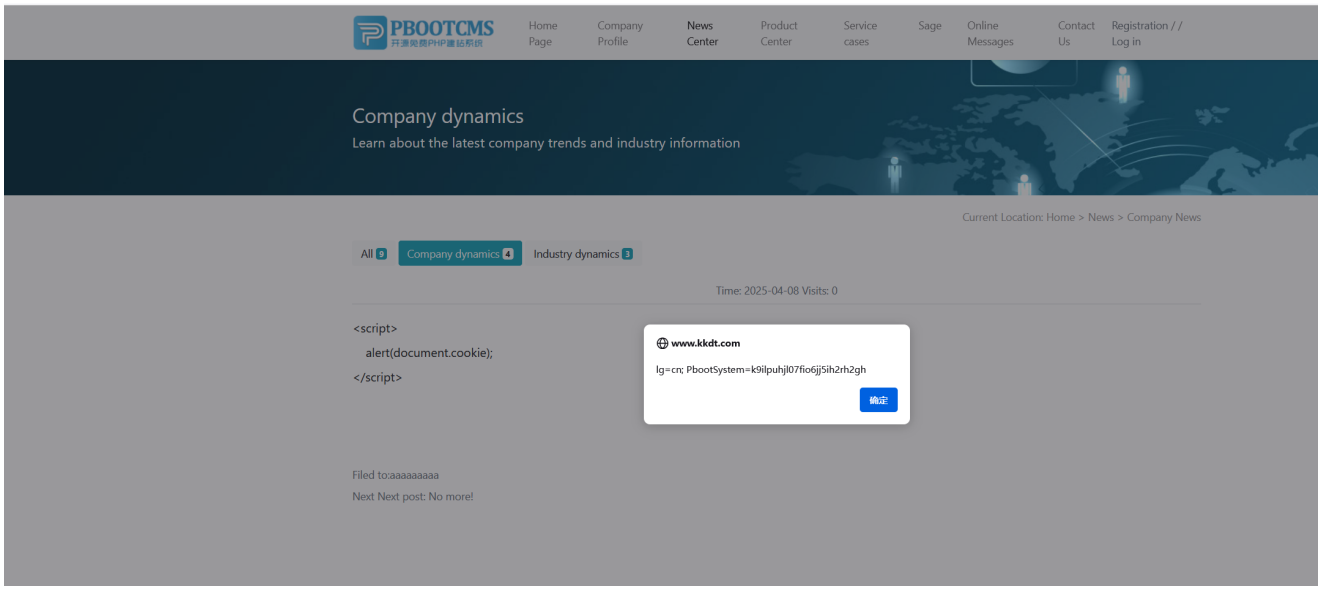
- apps/common/function.php
- core\extend\ueditor\php\Uploader.class.php

Description:

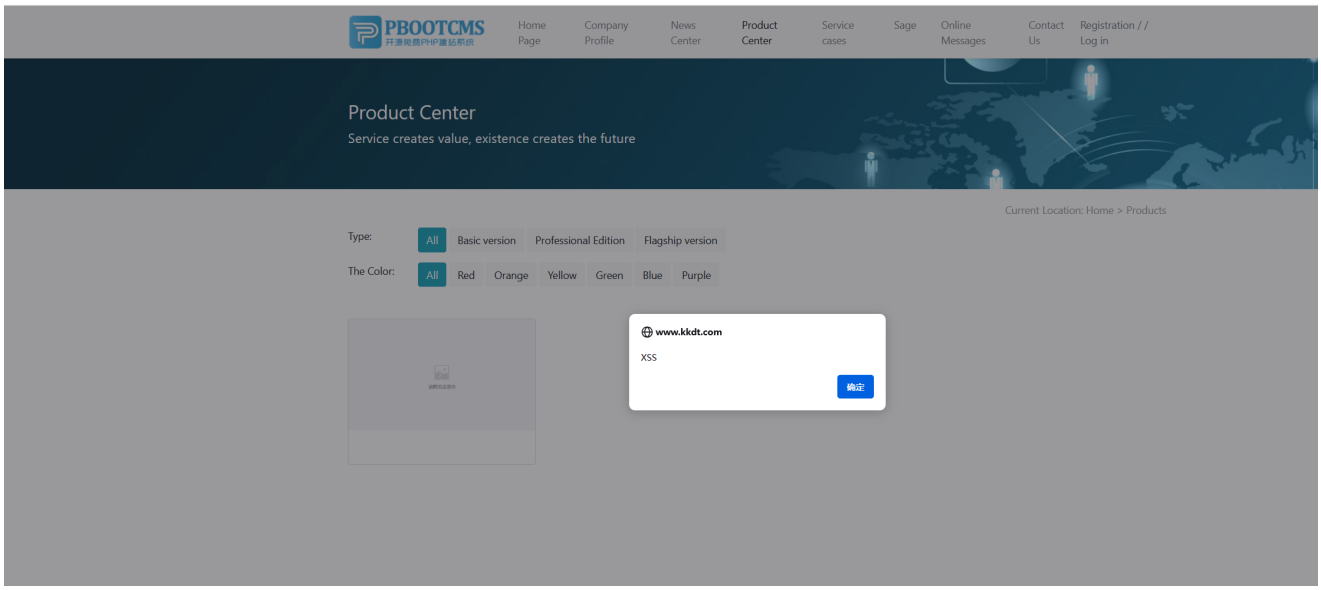
1. First, enter the rich text editor in the news content of the background interface, and write xsspayload, such asAs shown in Fig



2.A storage-optimized XSS vulnerability is found to be uploaded and displayed on the following page:



It can also be set to be displayed on the product page



The following is the code audit section:

Vulnerability details

1. `make_area_Select`

Location of the vulnerability

```
$list_html .= "<option value='{$values->acode}' $select $disabled>
{$blank}{$values->acode} {$values->name}";
```

Vulnerability analysis

- The code inserts '$values->name' directly into the HTML without escaping it in any way. If '$values->name' contains malicious script code (e.g. '', the script will be executed in the user's browser when the page is rendered.

Attack scenario

An attacker can pass content containing malicious scripts to the system as a region name through a form submission, and trigger an XSS attack when another user visits a page containing that region selection.

2.Multiple buttons generate functions（`get_btn_back`、`get_btn_add`、`get_btn_more`）

Location of the vulnerability

Take the 'get_btn_back' function as an example:

```
$btn_html = "<a href='" . $url . "' class='layui-btn layui-btn-primary'>$btnName</a>";
```

Vulnerability analysis

- These functions insert user-controllable data (e.g., '$url', '$btnName') directly into the HTML when generating HTML buttons, without escaping. If this data contains malicious scripts, it can lead to XSS vulnerabilities. For example, if '$url' is constructed as 'javascript:alert('XSS')', the script will be executed when the user clicks the button.

Attack Scenario

An attacker can trigger an XSS attack by tampering with URL parameters or submitting a button name containing malicious scripts to trick users into clicking on a button.

3. Steps to reproduce the vulnerability

'make_area_Select' function

1. Constructing Malicious Input: Constructing a zone name that contains malicious scripts, such as '). `。
2. Enter malicious data: Submit the malicious name to the system as a region name through a form or other data entry method.
3. **Trigger Vulnerability**: Visit the page with the region selection, if the page renders properly, an 'alert' box will pop up, indicating that the XSS vulnerability has been triggered.

button generation function reproduction

1. Construct Malicious URL: Construct a malicious URL, such as 'javascript:alert('XSS')'.

2. Tampering with URL parameters: Modify URL parameters to pass a malicious URL to the appropriate button generator function.
3. **Trigger vulnerability**: Click the generated button, if the 'alert' box pops up, it means that the XSS vulnerability has been triggered.

IV. Impact of the vulnerability

- User information disclosure: An attacker can exploit an XSS vulnerability to steal sensitive information such as a user's session cookies and login credentials, and then impersonate the user.
- Page content tampering: An attacker can inject scripts to modify page content to mislead users or spread malicious information.
- **Damaged User Experience**: Frequent pop-up 'alert' boxes or page exceptions can severely impact the user experience.

V. Repair suggestions

1. 'make_area_Select' function fix

HTML entity escape for '$values->name' can be done using the 'htmlspecialchars' function.

```php
$list_html .= "<option value='{$values->acode}' $select $disabled>{$blank}
{$values->acode} ". htmlspecialchars($values->name, ENT_QUOTES, 'UTF-8') .
"";
```

2. Button generation function fix

Escape user-controlled data (e.g., '$url', '$btnName').

2. Button generation function fix

Escape user-controlled data (e.g., '$url', '$btnName'). 6. Summary

The 'make_area_Select' function and multiple button generators in this PHP file are at risk of XSS vulnerabilities, and the data entered by the user needs to be strictly escaped to avoid the injection and execution of malicious scripts. After a fix, adequate testing should be carried out to ensure that the vulnerability has been completely fixed.