

# Calculat3 M3 CTF Writeup

This document is a walkthrough on one way to solve the **Calculat3 M3 CTF** on **CTFLearn**. The objective is to explain how I was able to solve this CTF to my future self.

## General Information

- *Difficulty:* **Easy**
- *Category:* **Web**
- *Link:* [Calculat3 M3 - CTFLearn](#)

## Solution

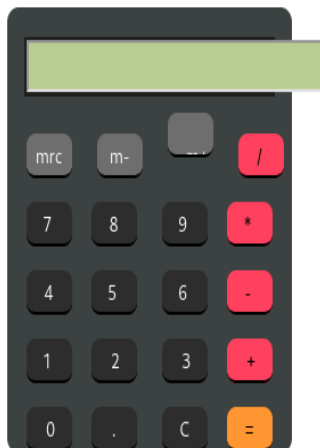
Calculat3 M3

80 points

Hard

Here! <http://web.ctflearn.com/web7/> I forget how we were doing those calculations, but something tells me it was pretty insecure.

We're given a website, with just a calculator which can do basic operations.



Now, looking at the javascript and HTML source code, we can guess how it works:

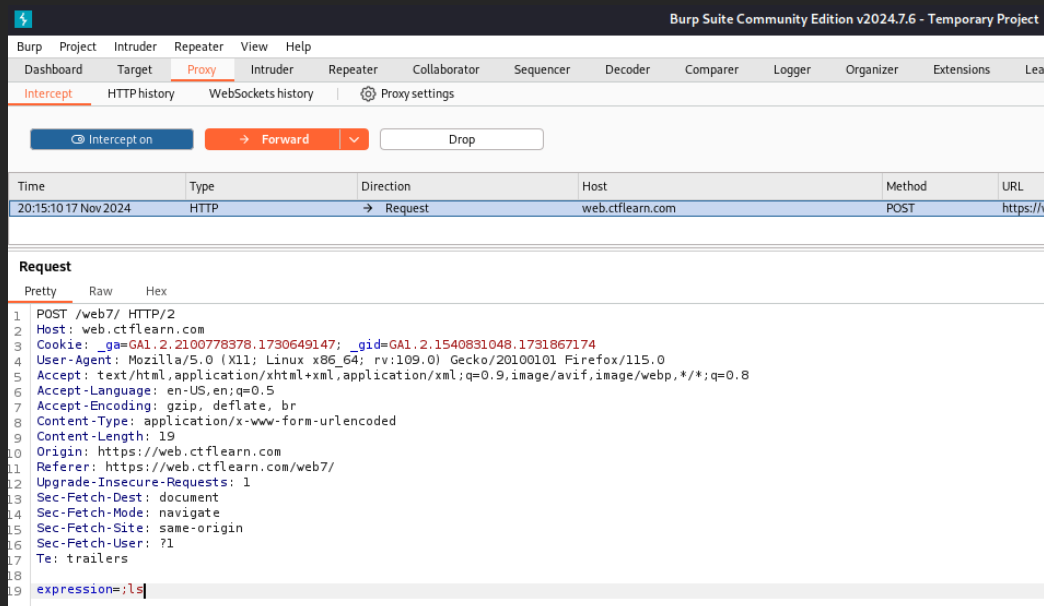
```
2 <html>
3 <head>
4   <link rel="stylesheet" href="main.css">
5   <script type="text/javascript" src="calc.js"></script>
6
7   </head>
8   </body>
9   <div class="box">
10  <div class="display">
11    <form action='.' method="post">
12    <input type="text" name="expression" readonly size="18" id="d"></div>
13  <div class="keys">
14    <p><input type="button" class="button gray"
15    value="mrc" onclick='c("Created.....")'>
16    <input type="button" class="button gray"
17    value="m-" onclick='c(".....by.....")'>
18    <input type="button" class="button gray" value="
19    m+" onclick='c(".....Anoop")'>
20    <input type="button" class="button pink"
21    value="/" onclick='v("/ ")'></p>
22    <p><input type="button" class="button black"
23    value="7 " onclick='v("7 ")'><input type="button"
24    class="button black" value="8 " onclick='v("8 ")'>
25    <input type="button" class="button black" value="9 "
26    onclick='v("9 ")'><input type="button"
27    class="button pink" value="*" onclick='v("* ")'></p>
28    <p><input type="button" class="button black"
29    value="4 " onclick='v("4 ")'><input type="button"
30    class="button black" value="5 " onclick='v("5 ")'>
31    <input type="button" class="button black" value="6 "
32    onclick='v("6 ")'><input type="button"
33    class="button pink" value="- " onclick='v("- ")'></p>
34    <p><input type="button" class="button black"
35    value="1 " onclick='v("1 ")'><input type="button"
36    class="button black" value="2" onclick='v("2 ")'>
37    <input type="button" class="button black" value="3"
38    onclick='v("3 ")'><input type="button"
39    class="button pink" value="+" onclick='v("+ ")'></p>
40    <p><input type="button" class="button black"
41    value="0" onclick='v("0 ")'><input type="button"
42    class="button black" value="." onclick='v(".")'>
43    <input type="button" class="button black" value="C"
44    onclick='c("")'><input type="submit"
45    class="button orange" value="="></p>
46  </div>
47 </div>
48 </body>
49 </html>
```

```
function c(val)
{
document.getElementById("d").value=val;
}
function v(val)
{
document.getElementById("d").value+=val;
}
function e()
{
try
{
c(eval(document.getElementById("d").value))
}
catch(e)
{
c('Error')
}
}
}
```

- The calculator takes input from the user, appends it to a variable called “**expression**”
- Upon submit, it sends the “**expression**” variable to the server, which executes the e() function, which will evaluate the **expression**, using the eval function. For example, expression = “3 + 3”.

But what if we were able to modify the value of expression, and **injected some code** into it, since it’s sent to the server via **POST** method, so the client has full control over it.

Using Burp Suite, we’ve intercepted the POST request, and modified the expression value to “**!s**”. Now, the eval function which interprets the string as code, and executes it. In this case the “!s” command lists all directories in a UNIX-like system, and javascript can execute it here.



And there we have it, it seems like one of the directories’ name is literally the flag

```
calc.js ctf{watch_out_for_th3_m0ng00s3} index.php main.css main.css
```

```
ctf{watch_out_for_th3_m0ng00s3}
```