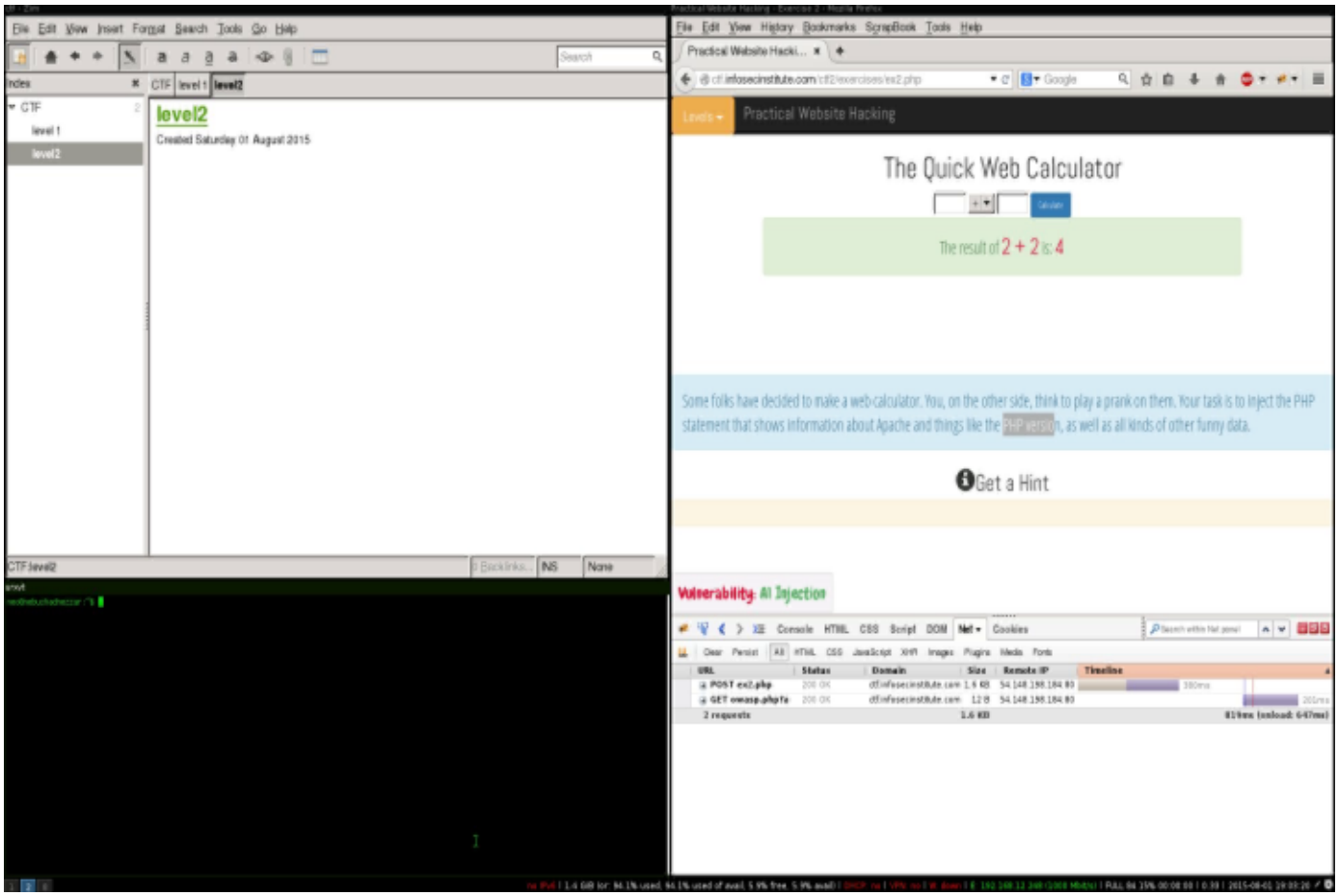
Infosecinstitute CTF 2 - LEVEL 2

This will be solution for Level 2 Practical Web Hacking CTF #2.

In this level we have a simple web php calulator, the vulnerability type is A1 Injection and our objective is successfully execute code in order to get information about the server and PHP version.



This is consistent with PHP eval statement, after first inspection we come to the conclusion that on the server site might be something like the flowing:

```
<?php
        $operand1 = $_GET['operand1'];
        $operator = $_GET['operator'];
        $operand2 = $_GET['operand2'];
        eval("echo ".$operand1.$operator.$operand2.";");
?>
```

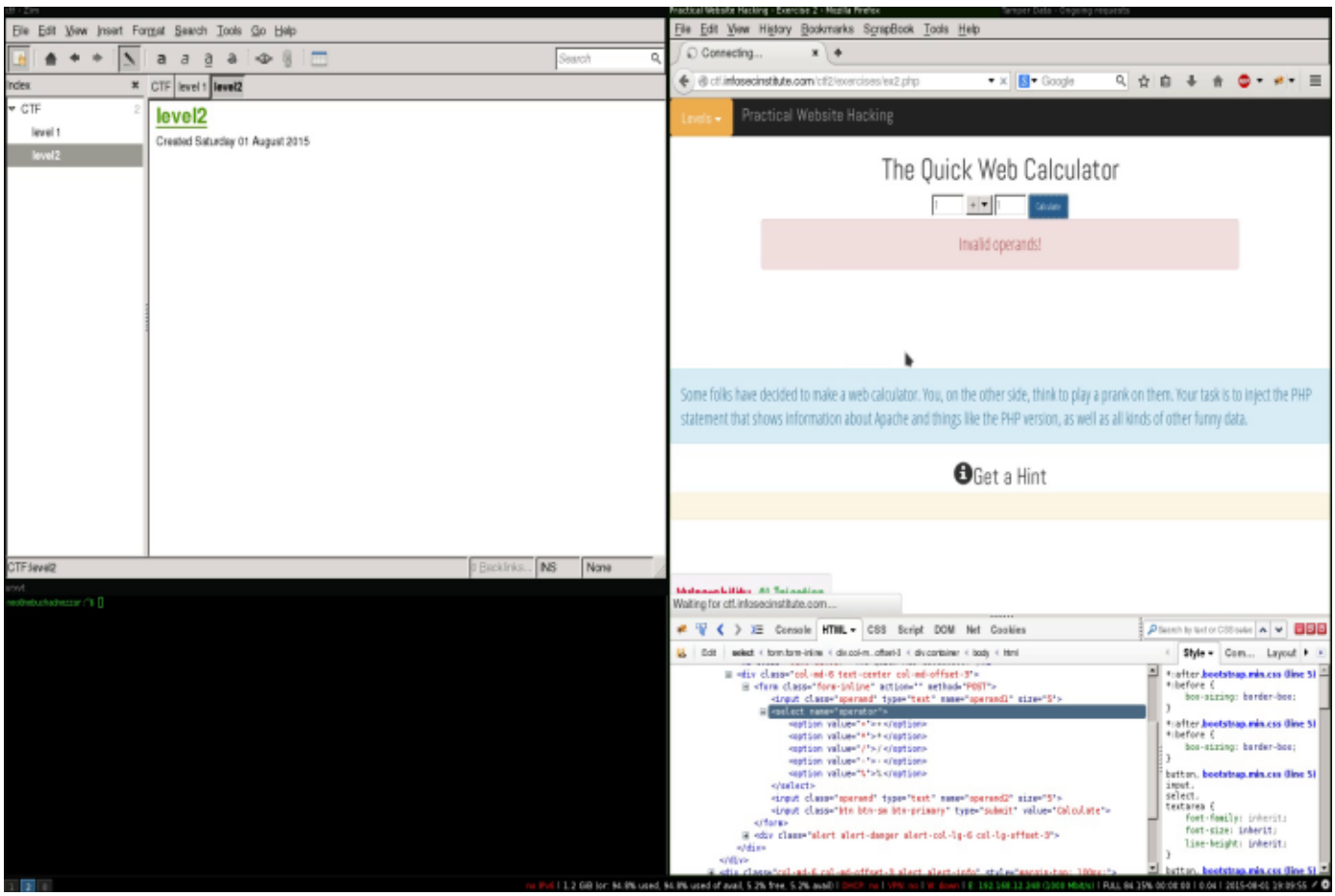
The code above is a representation of what might be running on the server, this level shows a very common misconception that select boxes can't be changed from the client side, but the truth is they can. Now that i have an idea of what might be on the other side my options are:

Possible Attack vectors

1. **Inject code in one of the operand (numeric) inputs**
2. **Inject code in one of the operator input**

After testing option **1** more detais about the implementation were revealed, the possiblity of a validation or cast, so our representation of the server side might look like:

```
<?php
        $operand1 = $_GET['operand1'];
        $operator = $_GET['operation'];
        $operand2 = $_GET['operand2'];
        eval("echo ".(int)$operand1.$operator.(int)$operand2.";");
?>
```

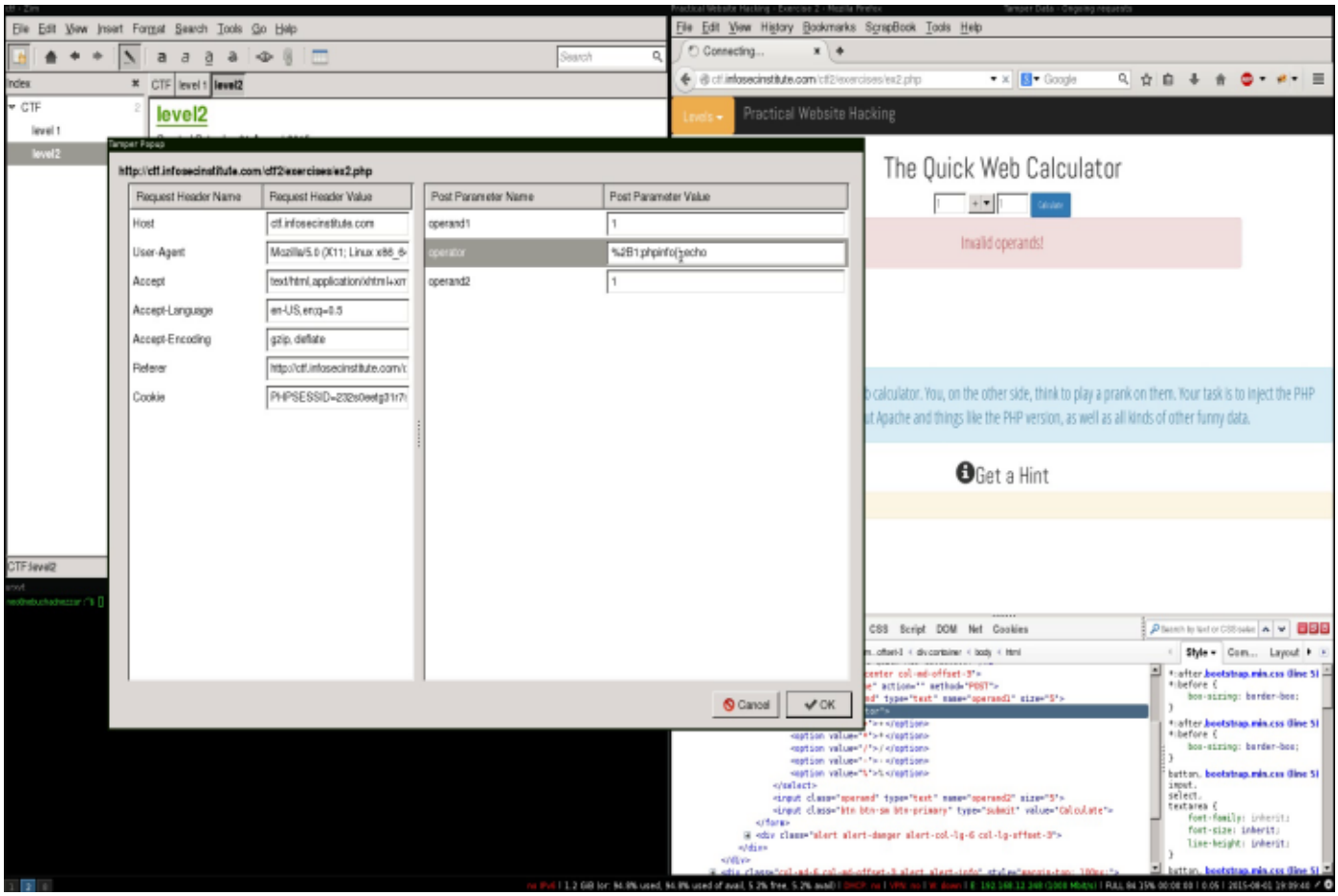


This leaves me with option **2** since we don't see any php native erros we can assume that some error handling might be in place which means that we need a valid operation the flowing is a representation how the eval statment needs to look like.

```php
<?php
        $operand1 = 1;
        $operator = "+ 2;phpinfo();echo ";
        $operand2 = 2;
        eval("echo ".(int)$operand1.$operator.(int)$operand2.";");
?>
```

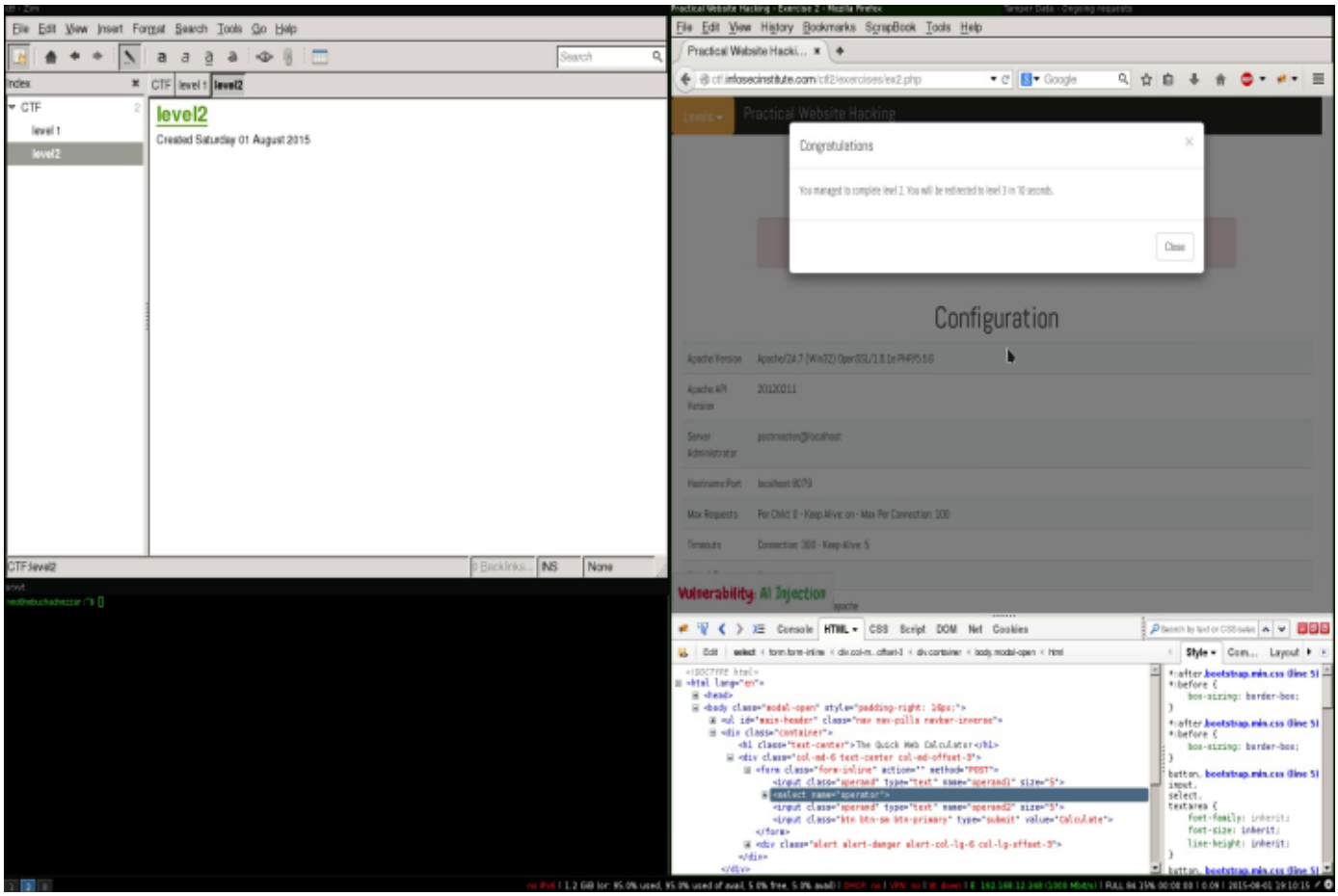
If we succeed the mathematical operation will be executed as expected next to our **phpinfo()** , since their might be an **";"** in the end, we can also inject an echo that will result in **3** operations executed by the eval statement.

The simplest way to modify the operation value (select box) sent to the server, is to modify the values before then are sent to the server in this case i used Tamper data



Exection flow

1. Start Temper data.
2. Set values in the form and click submit.
3. Change the operator field and add the flowing **"2;phpinfo();echo"**.
4. Submit.



Video

0x4E0x650x6F
0x4E0x650x6F