Warm Up

发表于 2021-01-09 分类于 Challenge， 2020， CSICTF， Web
Challenge | 2020 | CSICTF | Web | Warm Up

点击此处获得更好的阅读体验

WriteUp来源

https://dunsp4rce.github.io/csictf-2020/web/2020/07/19/Warm-Up.html

by `INXS_JOY`

题目描述

> *If you know, you know; otherwise you might waste a lot of time.*

题目考点

解题思路

```
`<?php

if (isset($_GET['hash'])) {
if ($_GET['hash'] === "10932435112") {
die('Not so easy mate.');
}

$hash = sha1($_GET['hash']);
$target = sha1(10932435112);
if($hash == $target) {
include('flag.php');
print $flag;
} else {
print "csictf{loser}";
}
} else {
show_source(__FILE__);
}

?>`
```

This PHP code was provided when the above link is visited. PHP's == is notoriously know for type juggling. You can learn more about the vulnerability here.

> *The baseline is that, == operator in PHP converts strings which look like a number to a number before comparing. So, `sha(10932435112) gives 0e07766915004133176347055865026311692244`, which in integer terms is 0*10^07766915004133176347055865026311692244. We know that == converts anything which looks like integer, so 0^anthing is zero. Now this value is getting compared to the \(hash variable which is the sha1()hash which we send). So we need to find a string whose sha1() produces a hash starting with 0e`I just googled "sha1 hash starting with 0e". I used this [link](https://github.com/spaze/hashes/blob/master/sha1.md), and took the first stringaaroZmOk. Sending this data, we get the flag.http://chall.csivit.com:30272/?hash=aaroZmOk*

Flag

```
csictf{typ3_juggl1ng_1n_php}
```

- *本文作者：* CTFHub
- *本文链接：* https://writeup.ctfhub.com/Challenge/2020/CSICTF/Web/4HQQAq4eYSGj2aJEp17WXa.html
- *版权声明：* 本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！

 # Challenge # 2020 # Web # CSICTF

Oreo
Body Count