[CTF]No.0005 [强网杯] broken

原创

林毅洋 于 2017-09-13 03:05:46 发布 680 收藏

分类专栏： CTF CTF每日一题 文章标签： ctf ctf每日一题

本文链接：https://blog.csdn.net/kajweb/article/details/77960225

CTF 同时被 2 个专栏收录

13 篇文章 0 订阅

订阅专栏

CTF每日一题

9 篇文章 0 订阅

订阅专栏

[CTF]No.0005 [强网杯] broken

来源：强网杯-第二届-WEB

类别：代码混淆 js加密

来源：https://gdqwb2017.ichunqiu.com/competition/index

用到工具：无

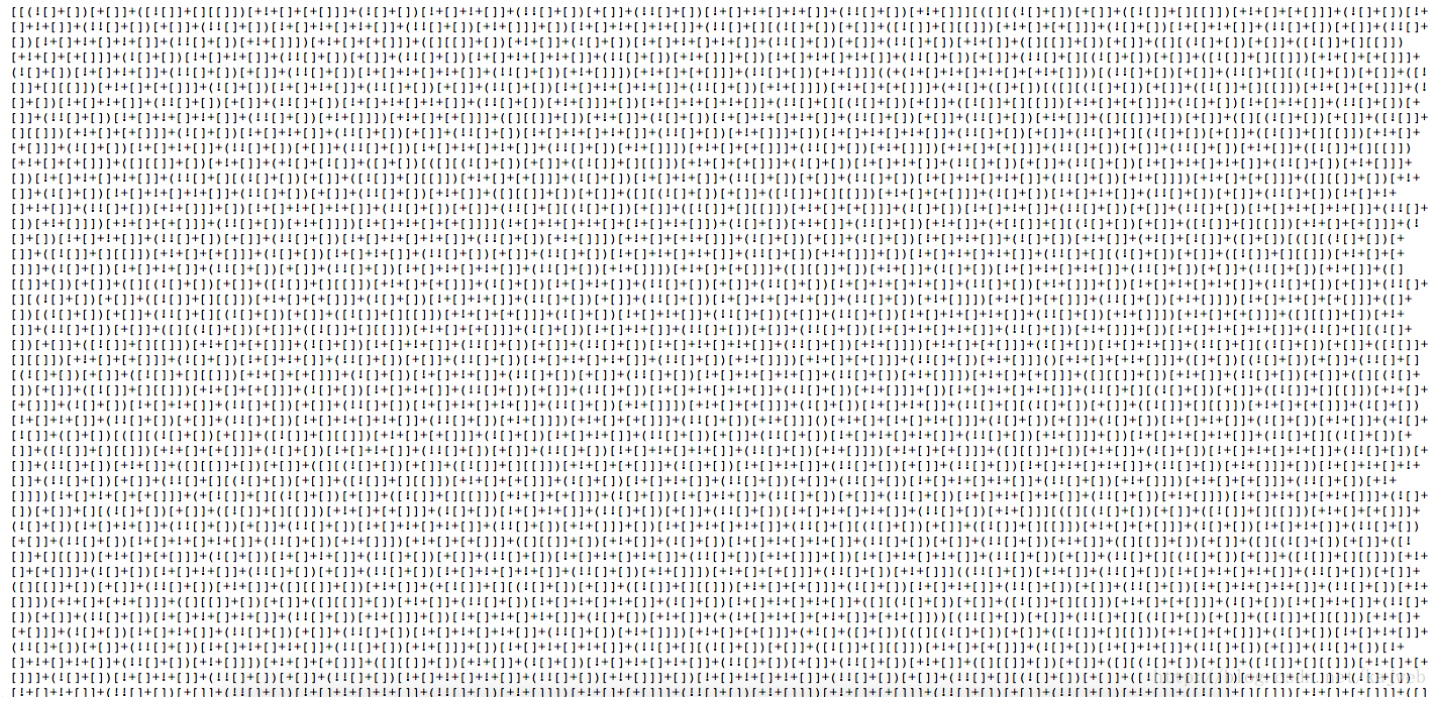
writeUp->Broken

尝试

进入首页，提醒：

> Hi, a CTFer. You got a file, but it looks like being broken.

查看源码，无异常。点进链接，得到一大串乱码：



经过百度搜索到可为能为JSFuc

分析

直接在console中运行js代码，发现运行不了。（听说在firefox会报错少了]），通过浏览JSFuck官网发现 .

Basics

```
false       =>  ![]
true        =>  !![]
undefined   =>  [][[]]
NaN         =>  +[![]]
0           =>  +[]
1           =>  +!+[]
2           =>  !+[]+!+[]
10          =>  [+!+[]]+[+[]]
Array       =>  []
Number      =>  +[]
String      =>  []+[]
Boolean     =>  ![]
Function    =>  []["filter"]
eval        =>  []["filter"]["constructor"]( CODE )()
window      =>  []["filter"]["constructor"]("return this")()
```

可以看到都是[]匹配的。再看代码和题目名字，联想到代码缺失的情况。

通过尝试在代码第二个位置添加]，可以看到浏览器alert(‘flag is not here’);

看回去看相关代码，并没有发现flag。所以可以考虑从混淆的代码入手。
通过去除后面的 () 得到了一个函数原型：

```
f anonymous() {
var flag="flag{f_f_l_u_a_c_g_k}";alert('flag is not here');
}
```

可以取得flag

方法2：

根据writeUp可以得到，firefox有反混淆器插件，代码修复后，通过之查看即可获得flag。

JSFuck代码：

```
[(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]
```