

# 2016HCTF giligili writeup

转载

[szaurora](#)



于 2017-03-04 13:05:06 发布



461



收藏

分类专栏: [writeup](#) 文章标签: [CTF](#) [writeup](#) [web](#) [信息安全](#) [HCTF](#)



[writeup](#) 专栏收录该内容

6 篇文章 1 订阅

订阅专栏

题目网址: <http://re4js.hctf.io/>

本题主要考察js及编程基本知识, 包括js函数、进制转换、异或等。

打开网页看到一张gif图片, 并听到那首非常有名的神曲, 下面有个输入框, 写着`hctf{xxxxxx}`, 点上去可以输入内容, 但一回车就显示“Sorry, you are wrong...”

按惯例先打开源码, 看到下面的一段js代码, 答案就全在里面了, 扒下来放在自己的电脑上慢慢研究。为此隆重请出这段代码:

```

//Come on and get flag:>

var_ = { 0x4c19cff: "random", 0x4728122: "charCodeAt",0x2138878: "substring", 0x3ca9c7b: "toString",
var$ = [ 0x4c19cff, 0x3cfbd6c, 0xb3f970, 0x4b9257a, 0x1409cc7, 0x46e990e,0x2138878, 0x1e1049, 0x164a
vara, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z;
function check() {
    var answer = document.getElementById("message").value;
    var correct = (function() {
        try{
            h= new MersenneTwister(parseInt(btoa(answer[_[$[6]]](0, 4)), 32));
            e= h[_[$[""+ +[]]]]()*("{}")[_[0x4728122]](0xc); for(var _1=0;_1<h.mti; _1++)
            l= new MersenneTwister(e), v = true;
            l.random();l.random(); l.random();
            o= answer.split("_");
            i= l.mt[~~(h.random()*$[0x1f])%0xff];
            s= ["0x" + i[_[$[42]]](0x10), "0x" + e[_[$[42]]](0x20).split("-
            e=- (this[_[$[42]]](0x31)(o[1]) ^ s[0]); if (-e != $[21]) return false;
            e^= (this[_[$[42]]](0x31)(o[2]) ^ s[1]); if (-e != $[22]) return false; e -
            t= new MersenneTwister(Math.sqrt(-e));
            h.random();
            a= l.random();
            t.random();
            y= [ 0xb3f970, 0x4b9257a, 0x46e990e ].map(function(i) { return $_[$[40]](i)+
            o[0]= o[0].substring(5); o[3] = o[3].substring(0, o[3].length - 1);
            u= ~~~~~~(a * i); if (o[0].length > 5) return false;
            a= parseInt(_[$[23]]("1", Math.max(o[0].length, o[3].length)), 3) ^eval(_[$[31]
            r= (h.random() * l.random() * t.random()) / (h.random() * l.random() *t.random(
            e^= ~r;
            r= (h.random() / l.random() / t.random()) / (h.random() * l.random() *t.random(
            e^= ~r;
            a+= _[$[31]](o[3].substring(o[3].length - 2)).split("x")[1]; if(parseInt(a.spli
            d= parseInt(a, 16) == (Math.pow(2, 16)+ -5+ "")) +o[3].charCodeAt(o[3].length -
            i= 0xffff;
            n= (p = (f = _[$[23]](o[3].charAt(o[3].length - 4), 3)) == o[3].substring(1,4))
            g= 3;
            t= _[$[23]](o[3].charAt(3), 3) == o[3].substring(5, 8) &&o[3].charCodeAt(1) * o
            h= ((31249*g) & i).toString(16);
            i= _[$[31]](o[3].split(f).join("").substring(0,2)).split("x")[1];
            s= i == h;
            return (p & t & s & d) === 1 || (p & t & s & d) === true;
        }catch (e) {
            console.log("gg");
            return false;
        }
    })());
    document.getElementById("message").placeholder= correct ? "correct" : "wrong";
    if(correct) {
        alert("Congratulations!you got it!");
    } else {
        alert("Sorry,you are wrong...");
    }
};

```

第一眼看得我头都晕了。这是啥?!好多十六进制数,还有各种\$啊\_啊一堆奇怪符号.....注意到如果把try里面的赋值变量竖着看,可以看到:

```
hello is ee that you are reading this
```

可以看到出题人满满的恶意啊.....不过这样也就知道，变量名为什么有26个字母，就是要组合这句话的。那么最后达到了correct，输出了Congratulations，应该就出结果了吧。然后看到check()函数返回的是p&t&s&d的值，显然必须要四个变量都为真，才能输出true。那么接下来就要找到满足这四个变量为真的字符串。

## p

关于p的代码如下：

```
n = (p = (f = _[$[23]](o[3].charAt(o[3].length - 4), 3)) == o[3].substring(1, 4));
```

\_\$[23]是啥？alert一下就看到是

```
function(_c, _d) { var _e = ""; for(var _f=0; _f<_d; _f++) { _e += _c; } return _e; };
```

这个函数，作用是将字符串\_c重复\_d次输出，这就意味着o[3]下标1,2,3这三个位的字符必须相等，且等于倒数第四位的字符。再看下o的代码：

```
o = answer.split("_");  
o[0] = o[0].substring(5); o[3] = o[3].substring(0, o[3].length - 1);
```

这两句说明o是将我们输入的内容以\_分开，且掐头去尾得到的数组，掐掉的头恰好是hctf{，掐掉的尾刚好是}，由此可以确定，我们输入的answer恰好是flag，且flag有四个单词。

## t

再来看关于t的代码：

```
t = _[$[23]](o[3].charAt(3), 3) == o[3].substring(5, 8) && o[3].charAt(1) * o[0].charAt(0) == 0x2ef3;
```

t的值由两个语句组合而成，前一个和p一样，说明o[3]下标5,6,7三个位的字符是一样的且等于下标为3的字符。而后一句话则非常关键，告诉我们o[3]下标为1的字符和o[0]下标为0字符的ascii码的乘积为0x2ef3（十进制12019），因式分解得到12019=7\*17\*101，那么这两个字符的ascii码只能分别是119和101，对应字符是w和e，但哪个对应哪个字符还不知道。再结合p的代码就知道o[3]的1,2,3,5,6,7位都是w或e。

## s

然后是关于s的代码：

```
h= ((31249*g) & i).toString(16);
i= _[$[31]](o[3].split(f).join(" ").substring(0, 2)).split("x")[1];
s= i == h;
```

这段比较简单，`toString(16)`是将整数转换为16进制的字符串，算一下得到`h="6e33"`，而`_[$[31]]`则是

```
function(_9) {var _8 = []; for (var _a = 0, _b = _9.length; _a < _b; _a++) {_8.push(Number(_9.charCodeAt(_a
```

，任务是将字符串`_9`的每一个字符化成对应的ascii码数值，再连成字符串，带上0x的头输出。f由刚才的判断是"eee"或"www"，截断并取前两个字符，恰好是`o[0][0]`和`o[0][3]`，分别对应的ascii十六进制码为6e和33，这样就得到`o[0][0]='n'`和`o[0][3]='3'`。

d

d要为真算起来比较复杂，还是先看代码：

```
a = parseInt(_[$[23]]("1", Math.max(o[0].length,o[3].length)), 3) ^ eval(_[$[31]](o[0]));
a+= _[$[31]](o[3].substring(o[3].length - 2)).split("x")[1]; if(parseInt(a.split("84")[1], $.length/
d= parseInt(a, 16) == (Math.pow(2, 16)+ -5+ "") +o[3].charCodeAt(o[3].length - 3).toString(16) + "53
```

d的前半截和a有关，先不去管，后半截是字符串拼接，`Math.pow(2,16)+ -5=65531`，`newDate().getFullYear() - 1=2015`，还有`o[3][length-4]`的ascii码16进制值是未知的（假设为XX），合起来是"65531XX538462015"，然后再看a，`$.length`是50，这句

```
if(parseInt(a.split("84")[1], $.length/2) != 0x4439feb) return false;
```

就是将a截掉84的后半部分作为25进制数，其值要等于`0x4439feb`，算得`a.split("84")[1]="783f3f"`，因为最后两个3f就是`o[3]`最后两个字符ascii码的16进制，于是得到`o[3]`最后两个字符是"??"。

接着就要敲定XX是多少。字符不多，试出来XX是64，对应的16进制是17481184783f3f，那么`o[3]`的倒数第三个字符就是d，第一行执行后的a就应该是1748118478，但是`o[0]`只知道第一个字符，`o[3]`和`o[0]`的长度均未知，因为代码中有一句

```
if (o[0].length > 5) return false;
```

也就是说`o[0]`的长度不会超过5，而`o[3]`的已知字符就有10个了，那么max的值至少是10。

接下来到第一句，异或的逆运算还是异或，因为`o[3]`比`o[0]`长，根据`o[3]`长度得到两个答案：如果`o[3]`长度是11，`o[0]`就是"h3r3"；如果`o[3]`长度是12，`o[0]`就是"h6&6"。但是`o[0]`第一个字符不是e或w吗！这困扰了我一阵，后来查资料发现，js的异或运算只计算32位或以下的数，大于32位部分的数字将会舍去。h3r3和h6r6的16进制码都恰好是8位，即2进制的32位，所以前面再加字符不影响a的值了。因此可以把e或w加在前面得到的`o[0]`上。

这样就得到了o[0]和o[3]的值。接下来要看几个return false的语句，这些也会导致程序无法走到correct，前面已经看了几个，就剩下e的两个：

```
s = ["0x" + i[_[$$.length/2]](0x10), "0x" + e[_[$$.length/2]](0o20).split("-")[1]];
e =- (this[_[$[42]]]this[_[$[42]]](o[31])(o[1]))^ s[0]; if (-e != $[21]) return false;
e ^= (this[_[$[42]]](o[31])(o[2])) ^s[1]; if (-e != $[22]) return false;
```

这里涉及到了s的值，s又和i有关，i和l有关，l是MersenneTwister函数产生的，一查这是个伪随机数生成算法，能够相当科学地产生随机数.....但是并没有任何卵用，因为一测发现s是常量，值是[0x-51a6949d,0x6e5ef2dc]，["0x" + i[\_[\$\$.length/2]](0x10)和"0x" + e[\_[\$\$.length/2]](0o20).split("-")[1]]是一个意思，都是toString(16)，也就是两个16进制数。

有了s，中间两个单词就比较简单了，\_[\$[42]]是eval，e=-\_[\$[31]](o[1])^s[0]=-s[21]，那么\_[\$[31]](o[1])=s[21]=0x697a，算得o[1]="iz"，下一行要注意的是e的值已变为s[21]，算得\_[\$[31]](o[2])=0x79307572，o[2]="y0ur"。最终得到四个答案：

hctf{wh3r3\_iz\_y0ur\_neee3eeed??}

hctf{eh3r3\_iz\_y0ur\_nwww3wwwd??}

hctf{wh6&6\_iz\_y0ur\_neee3eeeed??}

hctf{eh6&6\_iz\_y0ur\_nwww3wwwd??}

四个答案中显然第一个最有可能，输进去显示正确，题目就此结束。

做完题目后，最后我要严厉批判出题人，在题目中公然散播毒品，循环播放神曲，题目还要长时间面对页面，这是在借机传教！下次再见到这种题.....我还会继续做！

□

□

□