bugkuctf 江湖魔头Writeup

原创

丶没胡子的猫 于 2020-11-07 12:50:50 发布 115 收藏

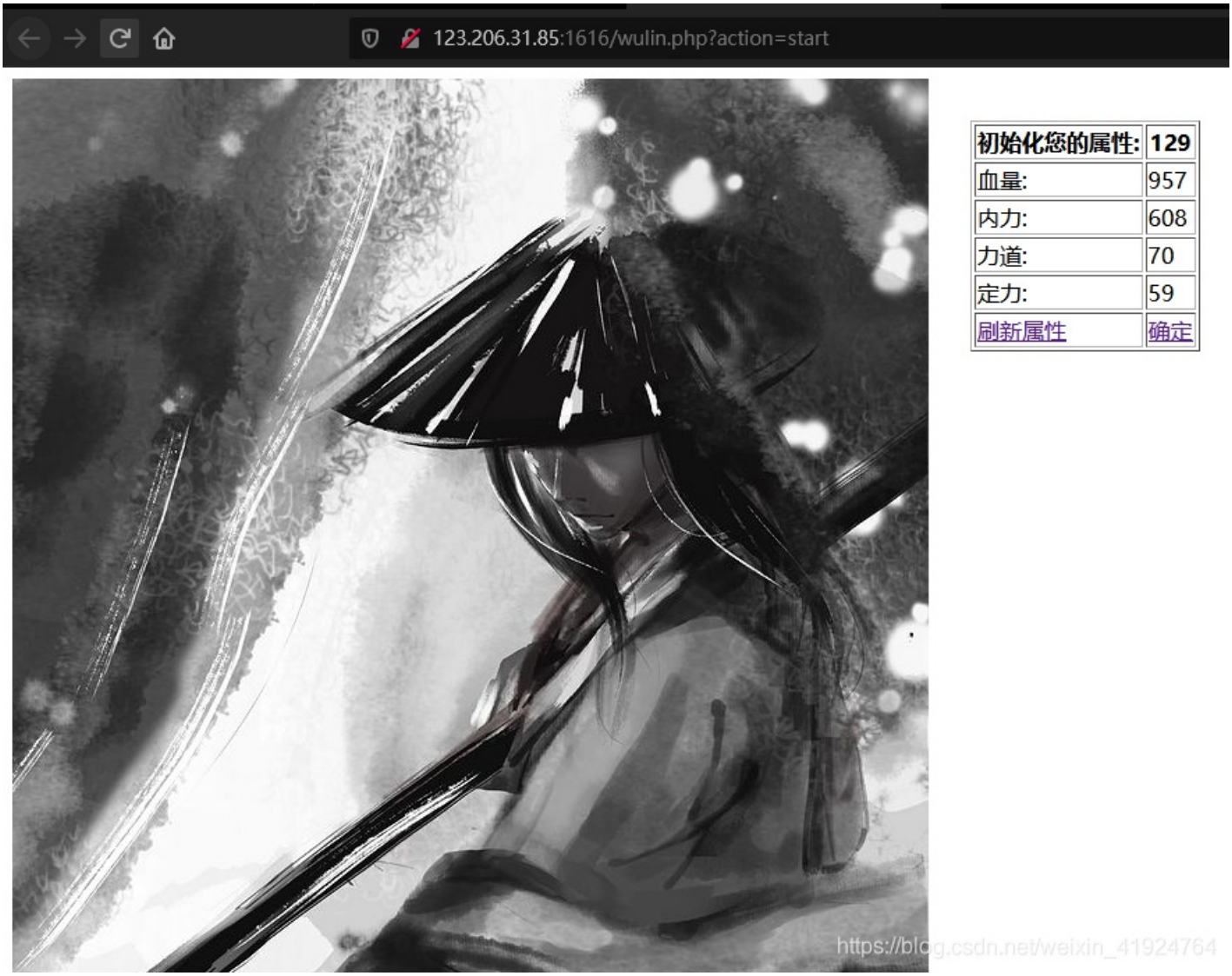
分类专栏： CTF 文章标签： js 安全

本文链接：https://blog.csdn.net/weixin_41924764/article/details/109488408

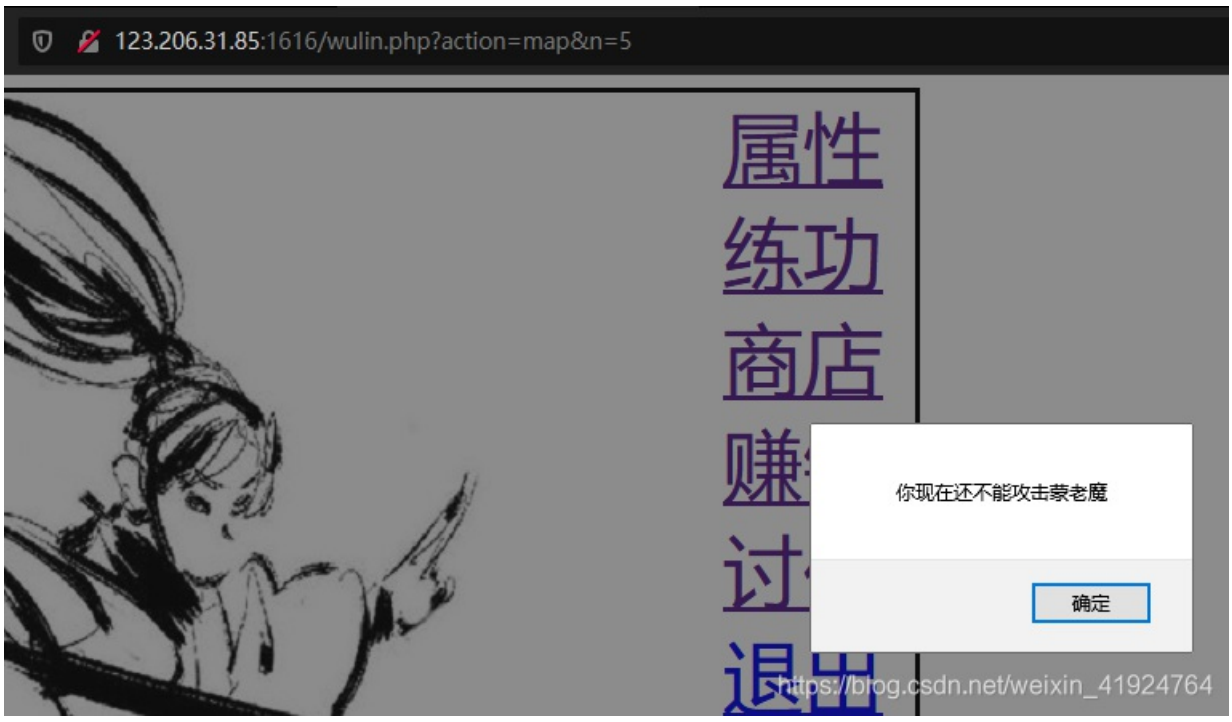
bugkuctf：https://ctf.bugku.com

靶机地址：http://123.206.31.85:1616/

进入靶机后，选择你要的人物属性，开始游戏

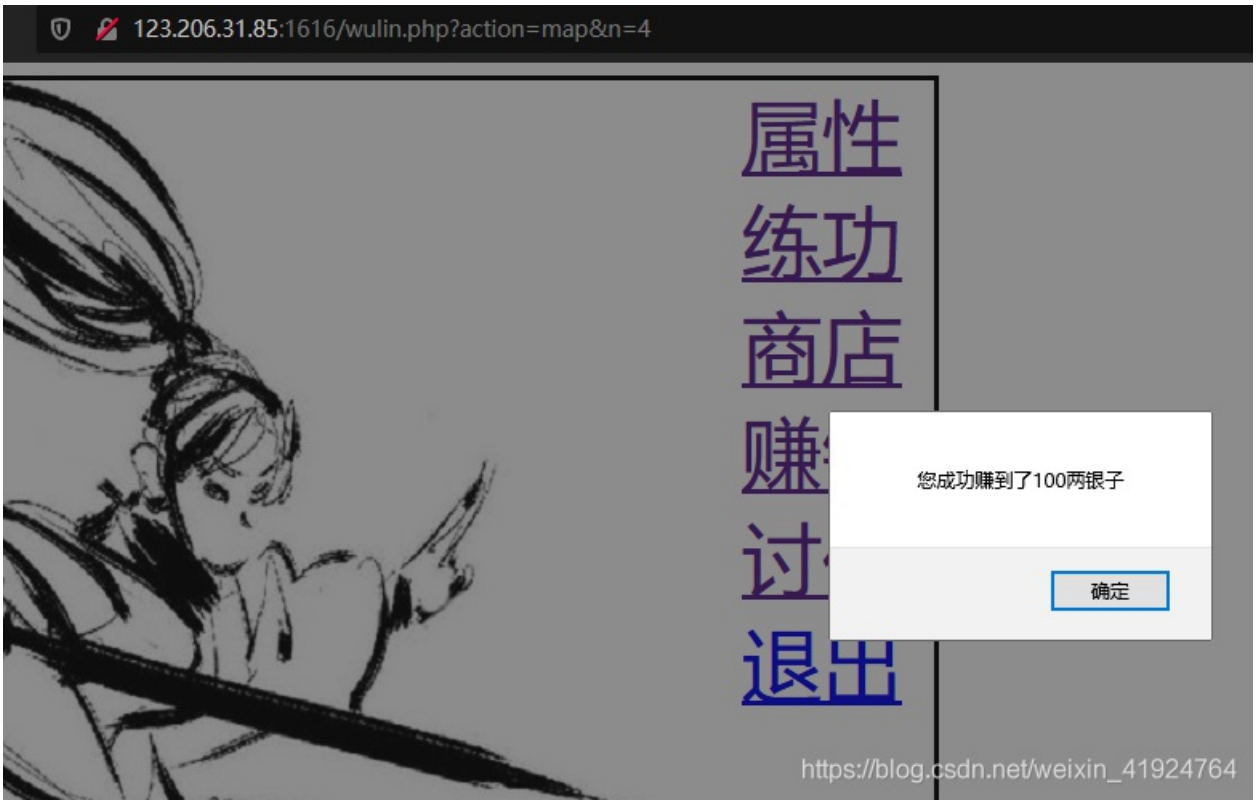


看了其他的游戏功能，发现我们需要通过讨伐，打败老魔才能通关

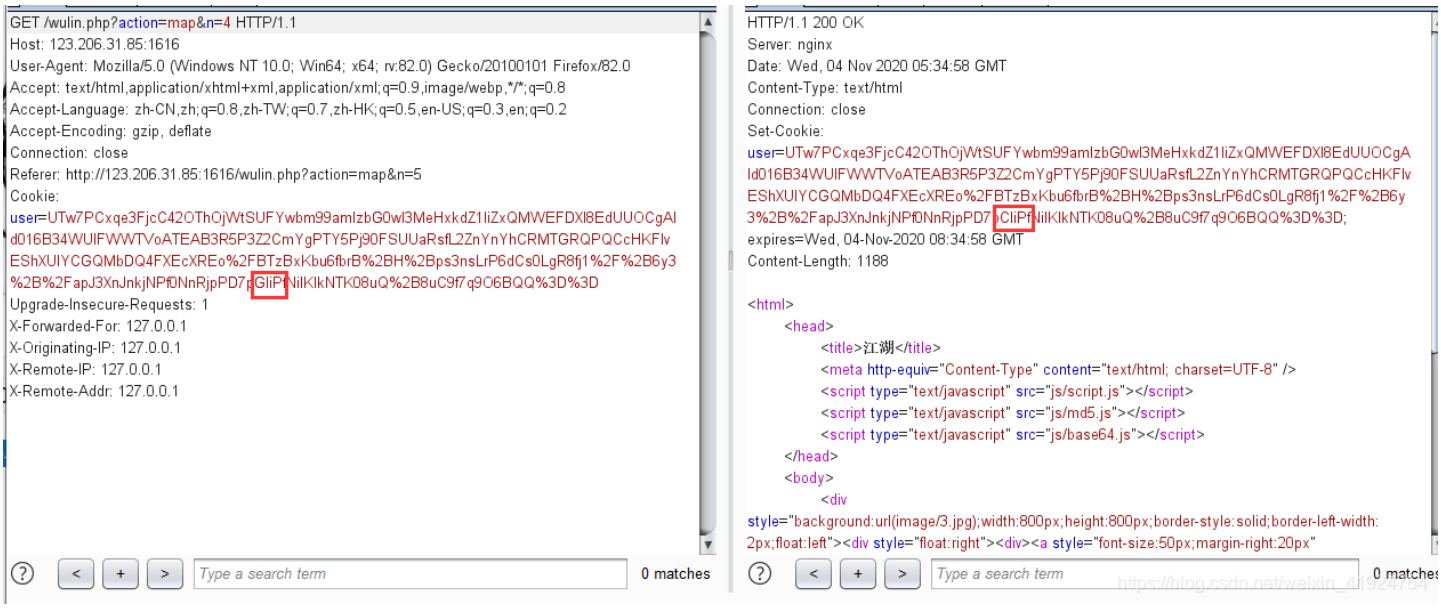


从商城可以购买如来神掌，但是需要利用银两购买，我们还有没有那么多钱购买秘籍

可以通过赚钱功能，每5秒可以赚100两

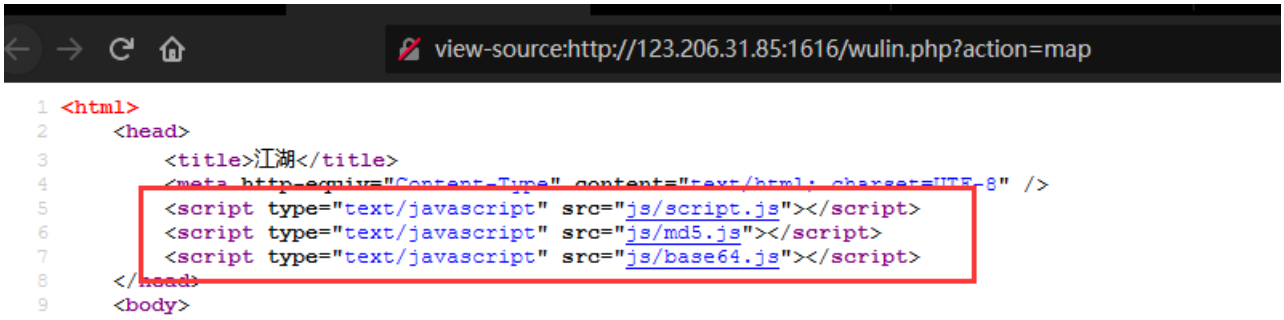


通过赚钱我们可以看到，服务器set-cookie返回给我们的cookie中，只有一个字母发生了变化而已

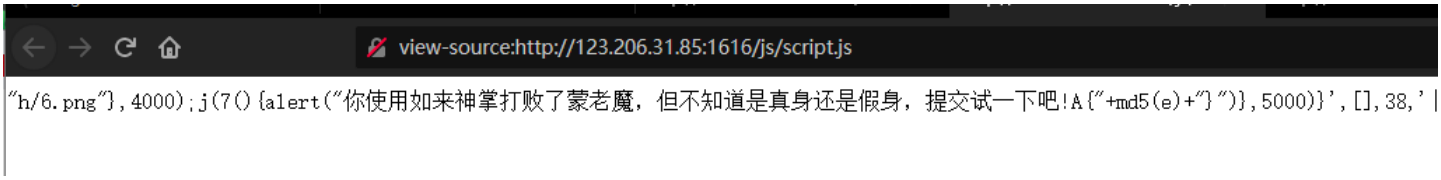


当然，我们可以写一个python脚本，自动匹配到返回包的setcookie替换掉自己的cookie，然后刷钱，但我算了一下，如来神掌是100000银两，我们需要刷1000次，5秒一次，就是5000秒，那就需要1个小时左右才刷出。

这不切实际，然后查看了html源代码，也没发现什么可以的，但有3个js文件。



当我点击了script.js后发现有些异常字符，然后我将浏览器编码设置成unicode，显示以下字符。



我将代码进行了js美化，并进行代码分析：

```
function getCookie(cname) {#下面函数中有一个变量user被传输进来
    var name = cname + "=";#name = "user="
    var ca = document.cookie.split(';');#以分号切割cookie，赋值到ca变量中
    for (var i = 0; i < ca.length; i++) {#有一个cookie就执行一次for循环
        var c = ca[i].trim();#trim函数为去空，将第一个cookie去空赋值c变量
        if (c.indexOf(name) == 0) # 从变量c（c就是我们的cookie）中查找字符串name（uname就是字符喜欢"user="），如果找到字符串"uname="排最前，那么就会返回0
         return c.substring(name.length, c.length)
         # c.length(为cookie的长度) name的长度为字符串"user="的长度
         # 返回cookie里等号及以后的内容
    }
    return "" #否则返回空
}# 这个函数就是截取cookie中user变量中的内容

function decode_create(temp) {#temp为url解码后的cookie
    var base = new Base64();#实例化base64到base变量中
    var result = base.decode(temp);# base64解码temp变量（temp为cookie）
    var result3 = "";
    for (i = 0; i < result.length; i++) {
        var num = result[i].charCodeAt();#返回字符串 Unicode 编码
        num = num ^ i;#位异或
        num = num - ((i % 10) + 2);
        result3 += String.fromCharCode(num)#将 Unicode 编码转为一个字符
    }
    return result3
}

function ertqwe() {
    var temp_name = "user";
    var temp = getCookie(temp_name);#截取cookie中user变量的内容
    temp = decodeURIComponent(temp);#将cookie url解码，重新赋值到temp变量中
    var mingwen = decode_create(temp);#传入decode_create，经过一次加密后重新赋值给mingwen变量，这里的执行过程我在下面解析时有讲解
    var ca = mingwen.split(';');#切割，以分号切割，赋值ca变量中
    var key = "";
    for (i = 0; i < ca.length; i++) {
        if (-1 < ca[i].indexOf("flag")) {
            key = ca[i + 1].split(":")[2]
        }
    }
    key = key.replace('"', "").replace('"', "");
    document.write('<img id="attack-1" src="image/1-1.jpg">');
    setTimeout(function () {
        document.getElementById("attack-1").src = "image/1-2.jpg"
    }, 1000);
    setTimeout(function () {
        document.getElementById("attack-1").src = "image/1-3.jpg"
    }, 2000);
    setTimeout(function () {
        document.getElementById("attack-1").src = "image/1-4.jpg"
    }, 3000);
    setTimeout(function () {
        document.getElementById("attack-1").src = "image/6.png"
    }, 4000);
    setTimeout(function () {
        alert("你使用如来神掌打败了蒙老魔，但不知道是真身还是假身，提交试一下吧!flag{" + md5(key) + "}")
    }, 5000)
}
```

`ertqwe()` 函数中，mingwen变量获取到的内容

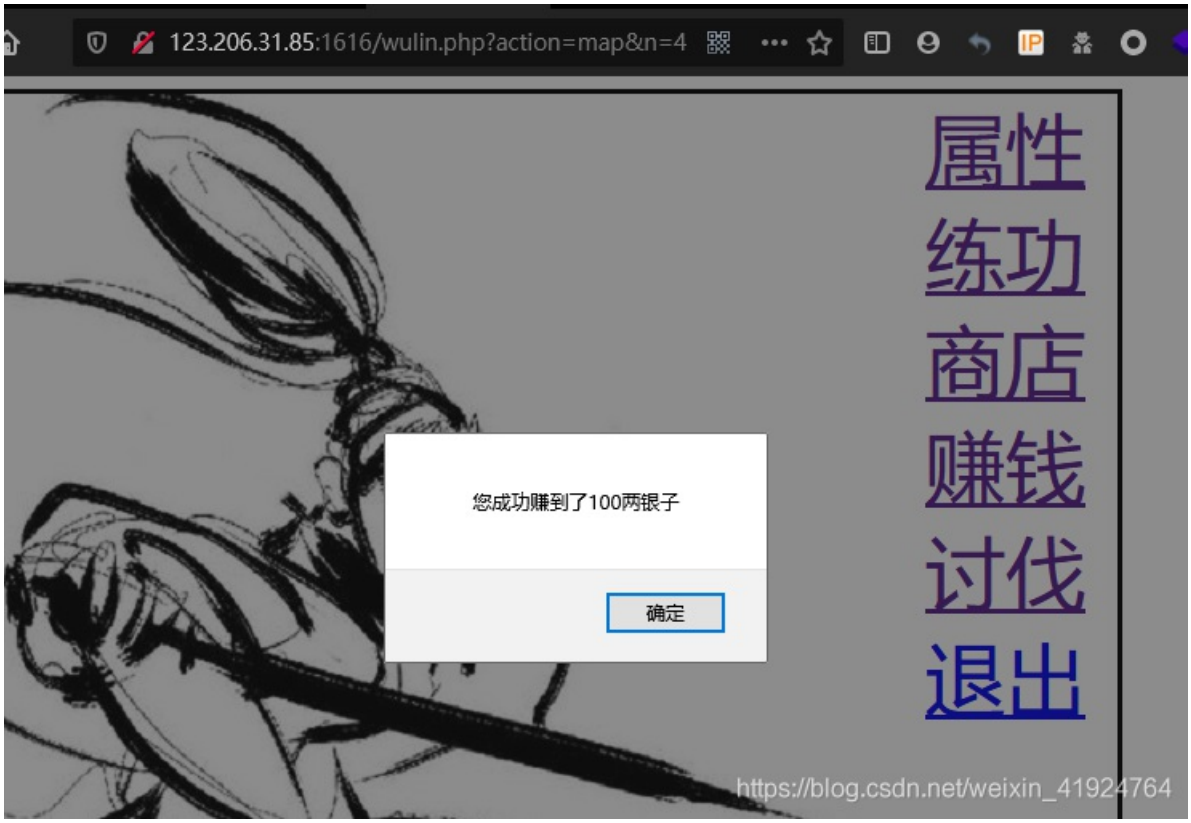
```
#原代码
var mingwen = decode_create(temp);
```

上面分析temp为我们自己的cookie，我将自己的cookie带入到decode_create()函数中执行，获得以下内容

```
O:5:\"human\":10:{s:8:\"xueliang\";i:758;s:5:\"neili\";i:758;s:5:\"lidao\";i:61;s:6:\"dingli\";i:60;s:7:\"waigong\";i:0;s:7:\"neigong\";i:0;s:7:\"jingyan\";i:0;s:6:\"yelian\";i:0;s:5:\"money\";i:0;s:4:\"flag\";s:1:\"0\";}
```

```
>> temp = decodeURIComponent(temp);
<- "UTw7PCxqe3FjcC42OThOjWtSUFYwbm99amlzbG0wI3MeHRkcZ1liZxQMWEFDXl8EdUUOCQ0Id016B34WUlFWWTVoATEABHV5P3Z2CmYgPTY5Pj90FSUUaGIfL2ZnYnYhCRMTGRQPQCcHKFIvEShXUlYCGQMbDQ4FXEcXREo/BTzBxKbu6fbrB+H+ps3nsLrP6dCs0LgR8fj1/+6y3+/apJ3XnJnkjNPf0NnRjpPD7pjzzfaMiJDcxt/XkP/B+I2C5vTqgUE="
>> var mingwen = decode_create(temp);
<- undefined
>> mingwen
<- "O:5:\"human\":10:{s:8:\"xueliang\";i:758;s:5:\"neili\";i:758;s:5:\"lidao\";i:61;s:6:\"dingli\";i:60;s:7:\"waigong\";i:0;s:7:\"neigong\";i:0;s:7:\"jingyan\";i:0;s:6:\"yelian\";i:0;s:5:\"money\";i:0;s:4:\"flag\";s:1:\"0\";}"
```

然后我再次去点击赚钱功能



将cookie再次带入脚本去跑，发现变化。

```
>> mingwen
<- "O:5:\"human\":10:{s:8:\"xueliang\";i:948;s:5:\"neili\";i:516;s:5:\"lidao\";i:57;s:6:\"dingli\";i:73;s:7:\"waigong\";i:0;s:7:\"neigong\";i:0;s:7:\"jingyan\";i:0;s:6:\"yelian\";i:0;s:5:\"money\";i:200;s:4:\"flag\";s:1:\"0\";}"
"O:5:\"human\":10:{s:8:\"xueliang\";i:758;s:5:\"neili\";i:758;s:5:\"lidao\";i:61;s:6:\"dingli\";i:60;s:7:\"waigong\";i:0;s:7:\"neigong\";i:0;s:7:\"jingyan\";i:0;s:6:\"yelian\";i:0;s:5:\"money\";i:0;s:4:\"flag\";s:1:\"0\";}"
```

这个是我刚刚点击赚钱功能，赚到的200块

这个是最初的cookie

现在的想法就是，改cookie，让我们边边钱，去学如来神掌

改cookie的方法就是将明文逆向编码：

刚开始写了挺久的逆向解密，发现调用base64中的encode时，怎么也解不出原来的加密代码，encode怎么也加密不出原来的cookie。

```
» var temp_name = "user";
← undefined
» var temp = getCookie(temp_name);
← undefined
» temp = decodeURIComponent(temp);
← "UTw7PCxqe3FjcC42OThOjWtSUFYwbm99amlzbG0wI3MeHBsUZ11lZxQMWEFDXl8EdUUOCgACd016B34WUlFWWTVoATEAAXF5P3Z2CmYgPTY5Pj90FSUUaGUfL2ZnYnYhCRMTGRQPQCcHKFIvEShXUlYCGQMbDQ4FXEcXREo/BTzBxKbu6fbrB++H+ps3nsLrP6dCs0LgR8fj1/+6y3+
/apJ3XnJnkjNPf0NnRjpPD7pjzzfaMiJDcxt/XkP/B+I2C5vTqgUE="
» var base = new Base64();
← undefined
» var result = base.decode(temp);
← undefined
» result
← "Q<;<,j{qcp.698N\u008dkRPV0no}jislm0#s\u001e\u001c\u001b\u0014gYbg\u0014\u000cXAC^_\u0004uE\u000e
\u0000\u0002wMz\u0007~\u0016RQvY5h\u00011\u0000\u0001qy?vv
f =69>?t\u0015%\u0014he\u001f/fgbv!     \u0013\u0013\u0019\u0014\u000f@'\u0007(R/\u0011(WRV\u0002\u0019\u0003\u001b\u000e\u0005\\G\u0017DJ?\u0005<ÁÄ¦Íéõë\u0007áþ¦Íç°ªÍéÐ¬Ð,\u0011Ñøðÿî²ßîÚM\u009d»\u009c\u0099ä\u008cÓßÐÛÑ\u008e\u0093Ãî\u0098óÍõ
\u008c\u0088\u0090Úæß×\u0090ÿÁø\u008d\u0082æõê\u0081A"
» var temp = base.encode(result)
← undefined
» temp
← "UTw7PCxqe3FjcC42OThOwo1rUlBWMG5vfWppc2xtMCNzHhwbFGdZYmcUDFhBQ15fBHVFDgoAAndNegd+FlJRVlk1aAExAAFxeT92dgpmID02OT4/dBUlFGhlHy9mZ2J2IQkTExkUD0AnByhSLxEoV1JWAhkDGw0OBVxHF0RKPwU8w4HDhMKmw67DqcO2w6sHw6HDvsKmw43Dp8KwwrrDj8Opw5DCrMOQwrgRw7HDuMO1w7
/DrsKyw5/Dr8OawqTCncOXwpzCmcOkwozDk8Ofw5DDmcORwo7Ck8ODw67CmMOzw43DtsKMwojCkMOcw4bDn8OXwpDDv8OBw7jCjCkCw6bDtMOqwoFB"
» |
```


猜想一定是base64.js有问题：

发现加密过程并不对应，encode调用了_utf8_encode
但decode并没有调用_utf8_encode

```
this.encode = function (input) {
    var output = "";
    var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
    var i = 0;
    input = _utf8_encode(input);
    while (i < input.length) {
        chr1 = input.charCodeAt(i++);
        chr2 = input.charCodeAt(i++);
        chr3 = input.charCodeAt(i++);
        enc1 = chr1 >> 2;
        enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
        enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
        enc4 = chr3 & 63;
        if (isNaN(chr2)) {
            enc3 = enc4 = 64;
        } else if (isNaN(chr3)) {
            enc4 = 64;
        }
        output = output +
        _keyStr.charAt(enc1) + _keyStr.charAt(enc2) +
        _keyStr.charAt(enc3) + _keyStr.charAt(enc4);
    }
    return output;
}

// public method for decoding
this.decode = function (input) {
    var output = "";
    var chr1, chr2, chr3;
    var enc1, enc2, enc3, enc4;
    var i = 0;
    input = input.replace(/[^A-Za-z0-9\+\/\=]/g, "");
    while (i < input.length) {
        enc1 = _keyStr.indexOf(input.charAt(i++));
        enc2 = _keyStr.indexOf(input.charAt(i++));
        enc3 = _keyStr.indexOf(input.charAt(i++));
        enc4 = _keyStr.indexOf(input.charAt(i++));
        chr1 = (enc1 << 2) | (enc2 >> 4);
        chr2 = ((enc2 & 15) << 4) | (enc3 >> 2);
        chr3 = ((enc3 & 3) << 6) | enc4;
        output = output + String.fromCharCode(chr1);
        if (enc3 != 64) {
            output = output + String.fromCharCode(chr2);
        }
        if (enc4 != 64) {
            output = output + String.fromCharCode(chr3);
        }
    }
    //output = _utf8_decode(output);
    return output;
}
```


只要我们注释掉encode中的`input = _utf8_encode(input);`即可将明文反加密，我们将base64复制到本地进行测试：

```
//1.html
<script>
```

```
function Base64() {

 // private property
 _keyStr = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";

 // public method for encoding
 this.encode = function (input) {
  var output = "";
  var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
  var i = 0;
  //input = _utf8_encode(input); (注释掉这个函数调用)
  while (i < input.length) {
   chr1 = input.charCodeAt(i++);
   chr2 = input.charCodeAt(i++);
   chr3 = input.charCodeAt(i++);
   enc1 = chr1 >> 2;
   enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
   enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
   enc4 = chr3 & 63;
   if (isNaN(chr2)) {
    enc3 = enc4 = 64;
   } else if (isNaN(chr3)) {
    enc4 = 64;
   }
   output = output +
   _keyStr.charAt(enc1) + _keyStr.charAt(enc2) +
   _keyStr.charAt(enc3) + _keyStr.charAt(enc4);
  }
  return output;
 }

 // public method for decoding
 this.decode = function (input) {
  var output = "";
  var chr1, chr2, chr3;
  var enc1, enc2, enc3, enc4;
  var i = 0;
  input = input.replace(/[^A-Za-z0-9\+\/\=]/g, "");
  while (i < input.length) {
   enc1 = _keyStr.indexOf(input.charAt(i++));
   enc2 = _keyStr.indexOf(input.charAt(i++));
   enc3 = _keyStr.indexOf(input.charAt(i++));
   enc4 = _keyStr.indexOf(input.charAt(i++));
   chr1 = (enc1 << 2) | (enc2 >> 4);
   chr2 = ((enc2 & 15) << 4) | (enc3 >> 2);
   chr3 = ((enc3 & 3) << 6) | enc4;
   output = output + String.fromCharCode(chr1);
   if (enc3 != 64) {
    output = output + String.fromCharCode(chr2);
   }
   if (enc4 != 64) {
    output = output + String.fromCharCode(chr3);
   }
  }
  //output = _utf8_decode(output);
  return output;
 }

 // private method for UTF-8 encoding
```

```
 // private method for UTF-8 encoding
 _utf8_encode = function (string) {
  string = string.replace(/\r\n/g,"\n");
  var utftext = "";
  for (var n = 0; n < string.length; n++) {
   var c = string.charCodeAt(n);
   if (c < 128) {
    utftext += String.fromCharCode(c);
   } else if((c > 127) && (c < 2048)) {
    utftext += String.fromCharCode((c >> 6) | 192);
    utftext += String.fromCharCode((c & 63) | 128);
   } else {
    utftext += String.fromCharCode((c >> 12) | 224);
    utftext += String.fromCharCode(((c >> 6) & 63) | 128);
    utftext += String.fromCharCode((c & 63) | 128);
   }

  }
  return utftext;
 }

 // private method for UTF-8 decoding
 _utf8_decode = function (utftext) {
  var string = "";
  var i = 0;
  var c = c1 = c2 = 0;
  while ( i < utftext.length ) {
   c = utftext.charCodeAt(i);
   if (c < 128) {
    string += String.fromCharCode(c);
    i++;
   } else if((c > 191) && (c < 224)) {
    c2 = utftext.charCodeAt(i+1);
    string += String.fromCharCode(((c & 31) << 6) | (c2 & 63));
    i += 2;
   } else {
    c2 = utftext.charCodeAt(i+1);
    c3 = utftext.charCodeAt(i+2);
    string += String.fromCharCode(((c & 15) << 12) | ((c2 & 63) << 6) | (c3 & 63));
    i += 3;
   }
  }
  return string;
 }
}


//原来我的cookie
var temp = "UTw7PCxqe3FjcC42OThOjWtSUFYwbm99amlzbG0wI3MeHBsUZ1liZxQMWEFDXl8EdUUOCgACd016B34WUlFWWTVoATEAAXF5P3Z2
CmYgPTY5Pj90FSUUaGUfL2ZnYnYhCRMTGRQPQCcHKFIvEShXUlYCGQMbDQ4FXEcXREo/BTzBxKbu6fbrB+H+ps3nsLrP6dCs0LgR8fj1/+6y3+/a
pJ3XnJnkjNPf0NnRjpPD7pjzzfaMiJDcxt/XkP/B+I2C5vTqgUE=";
//进行加密，获取明文

var base = new Base64();
var result = base.decode(temp);
var result3 = "";
for (i = 0; i < result.length; i++) {
 var num = result[i].charCodeAt();
 num = num ^ i;
 num = num - ((i % 10) + 2);
```

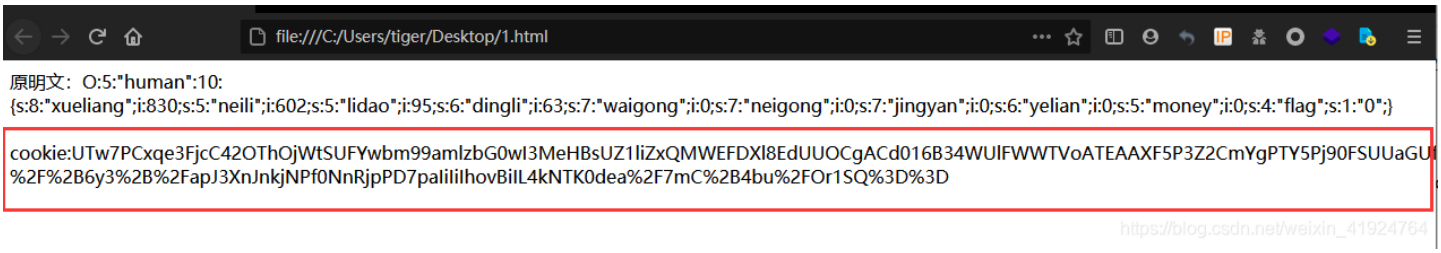
```
 result3 += String.fromCharCode(num);
}
document.write("原明文："+result3+'<br/>');
document.write('<br/>');
//修改明文
var result3 = 'O:5:"human":10:{s:8:"xueliang";i:830;s:5:"neili";i:602;s:5:"lidao";i:95;s:6:"dingli";i:63;s:7:"wa
igong";i:0;s:7:"neigong";i:0;s:7:"jingyan";i:0;s:6:"yelian";i:0;s:5:"money";i:200000;s:4:"flag";s:1:"0";}';

//反编码获取cookie
var result = "";
for (i = 0;i<result3.length;i++){
 num = result3[i].charCodeAt();
 num = num + ((i % 10) + 2);
 num = num ^ i;
 result += String.fromCharCode(num);
}
var temp= base.encode(result);

//将cookie进行url编码
temp = encodeURIComponent(temp);
document.write("cookie:"+temp+"</br>");
</script>
```

反编码后的cookie会输出再浏览器上，

```
UTw7PCxqe3FjcC42OThOjWtSUFYwbm99amlzbG0wI3MeHBsUZ1liZxQMWEFDXl8EdUUOCgACd016B34WUlFWWTVoATEAAXF5P3Z2CmYgPTY5Pj90
FSUUaGUfL2ZnYnYhCRMTGRQPQCcHKFIvEShXUlYCGQMbDQ4FXEcXREo%2FBTzBxKbu6fbrB%2BH%2Bps3nsLrP6dCs0LgR8fj1%2F%2B6y3%2B%2
FapJ3XnJnkjNPf0NnRjpPD7paIiIiIhovBiIL4kNTK0dea%2F7mC%2B4bu%2FOr1SQ%3D%3D
```



直接复制到我们的cookie中，就可以看到我们现在有20W银两了，美滋滋。

然后就去商店买如来神掌，讨伐老魔。