

第二节：html基础

前言

本次讲从实战的角度去讲解关于的html、css、以及javascript CTF实战题的做法。在我们浏览网页的过程中，html、css、javascript代码回直接发送到我们的客户端。故此类试题比较简单。

2.1html

在web的世界中，html、css一般只负责网站页面的美化，并不负责功能具体的实现。所以此类试题通过查看源码即可发现相关信息。

- 1.查看源码 此类型试题比较简单：一般在浏览器网页上点击右键，然后选择查看源码即可发现flag或者题目线索。
- 比如玄魂CTF平台上的"404 not found"右键查看源码后即可发现flag.



```
1 <html>
2 <body>
3 <center><h1>404 Not Found</h1></center>
4 <p hidden=flag{qwjdsklafjdfadfa}</p>
5 </body>
6 </html>
7
```

- 2.查看引用文件 有时可能需要点击网站引用css文件或者其他引用文件才会发现提示信息或者flag。

```
<ol id="billBox" class="bill_box">
</ol>
</div>
<div id="moveInfo" class="move_info"> </div>
</div>
<script src="js/common.js"></script>
<script src="js/play.js"></script>
<script src="js/AL.js"></script>
<script src="js/bill.js"></script>
<script src="js/[abcmlyx]{2}ctf{0-9}{3}.js"></script>
<script src="js/qambit.js"></script>
<div style="text-align:center;clear:both">
</div>
</div>
```

2.2js

js同html、css一样，代码传送到客户端，有客户端浏览器解释执行。不同的可以用js代码来做限制和校验，但是由于代码在客户端，我们可以直接查找到代码进行修改。

- 1、直接查看。js代码也是直接传输到客户端，如果flag藏在js代码。通过直接查看。即可获取到flag。
- ```
if ($(".input").val() == code && code != 9999) {
 alert("flag{CTF-bugku-0032}");
} else {
```
- 2、修改js 有些试题可能通过js来限制用户的某些输入，比如限制输入的长度。比如下面这道题。



根据提示进行计算，却发现只可以输入一位。摁F12。

56+74=?

2

验证

来源:BugKu-ctf

🔍 📄 Elements Console Sources Network Performance Memory Application Security Audits Adblock Plus

```
<!doctype html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>_</head>
 <body style>
 56+74=?
 ... <input type="text" class="input" maxlength="1"> == $0
 <button id="check">验证</button>
 >_</div>
 <script src="js/jquery-1.12.3.min.js"></script>
 <script type="text/javascript" src="js/code.js"></script>
 </body>
</html>
```

发现其限制最大输入长度为一位。点击进行修改。

56+74=?

2

验证

来源:BugKu-ctf

body | 1424 × 64

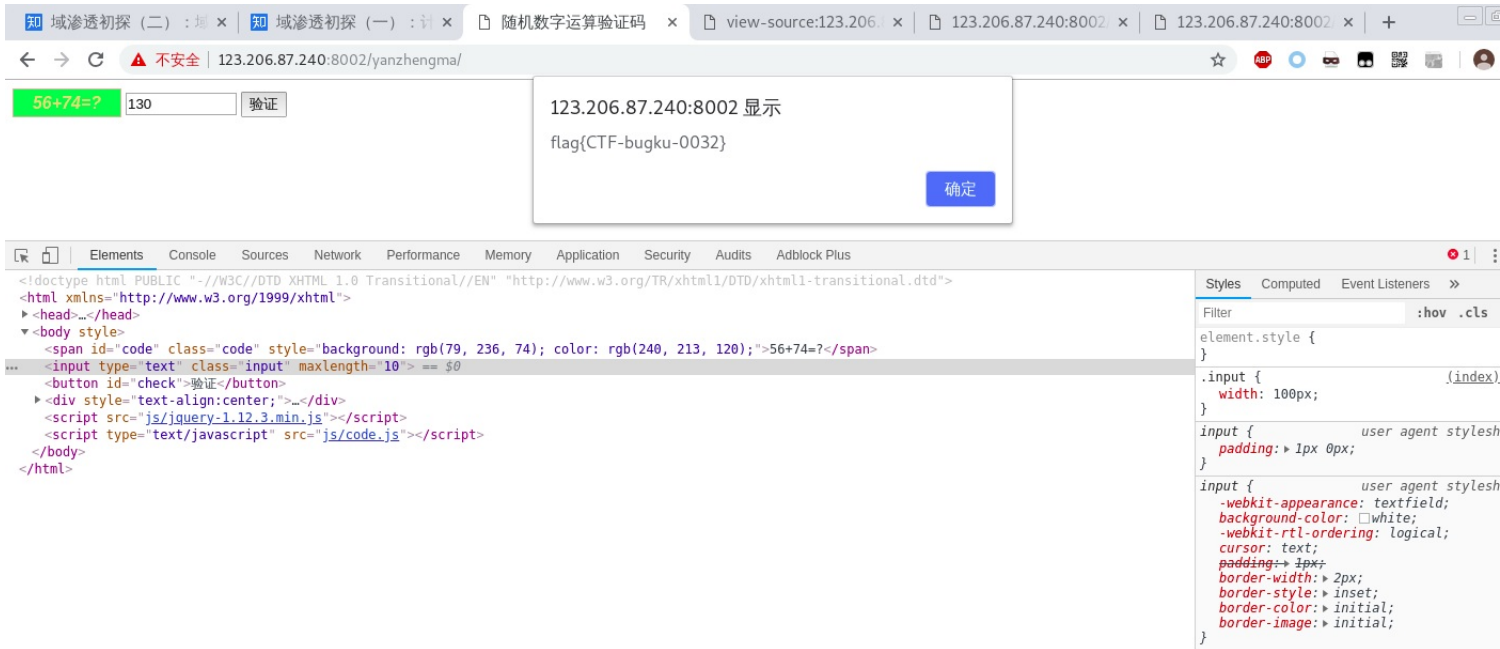
🔍 📄 Elements Console Sources Network Performance Memory Application Security Audits Adblock Plus

```
<!doctype html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>_</head>
 <body style>
 56+74=?
 ... <input type="text" class="input" maxlength="1"> == $0
 <button id="check">验证</button>
 >_</div>
 <script src="js/jquery-1.12.3.min.js"></script>
 <script type="text/javascript" src="js/code.js"></script>
 </body>
</html>
```

修改为10，然后重新输入计算结果。验证成功。

🔍 📄 Elements Console Sources Network Performance Memory Application Security Audits Adblock Plus

```
<!doctype html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>_</head>
 <body style>
 56+74=?
 <input type="text" class="input" maxlength="10"> == $0
 <button id="check">验证</button>
 <div style="text-align:center;">_</div>
 <script src="js/jquery-1.12.3.min.js"></script>
 <script type="text/javascript" src="js/code.js"></script>
 </body>
</html>
```



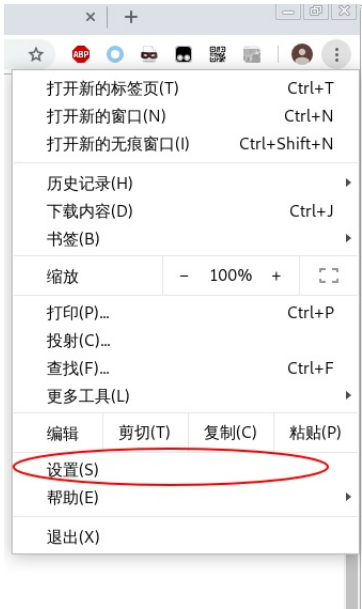
3、禁用js 有写网站仅仅使用js做校验。比如文件上传时限制文件类型。此时我们可以直接禁用js来绕过。  
比如靶场的第一关  
右键查看源码，发现它是通过js来做上传文件的校验。

```
<script type="text/javascript">
function checkFile() {
 var file = document.getElementsByName('upload_file')[0].value;
 if (file == null || file == "") {
 alert("请选择要上传的文件!");
 return false;
 }
 //定义允许上传的文件类型
 var allow_ext = ".jpg|.png|.gif";
 //提取上传文件的类型
 var ext_name = file.substring(file.lastIndexOf("."));
 //判断上传文件类型是否允许上传
 if (allow_ext.indexOf(ext_name) == -1) {
 var errMsg = "该文件不允许上传，请上传" + allow_ext + "类型的文件，当前文件类型为：" + ext_name;
 alert(errMsg);
 return false;
 }
}
</script>
```

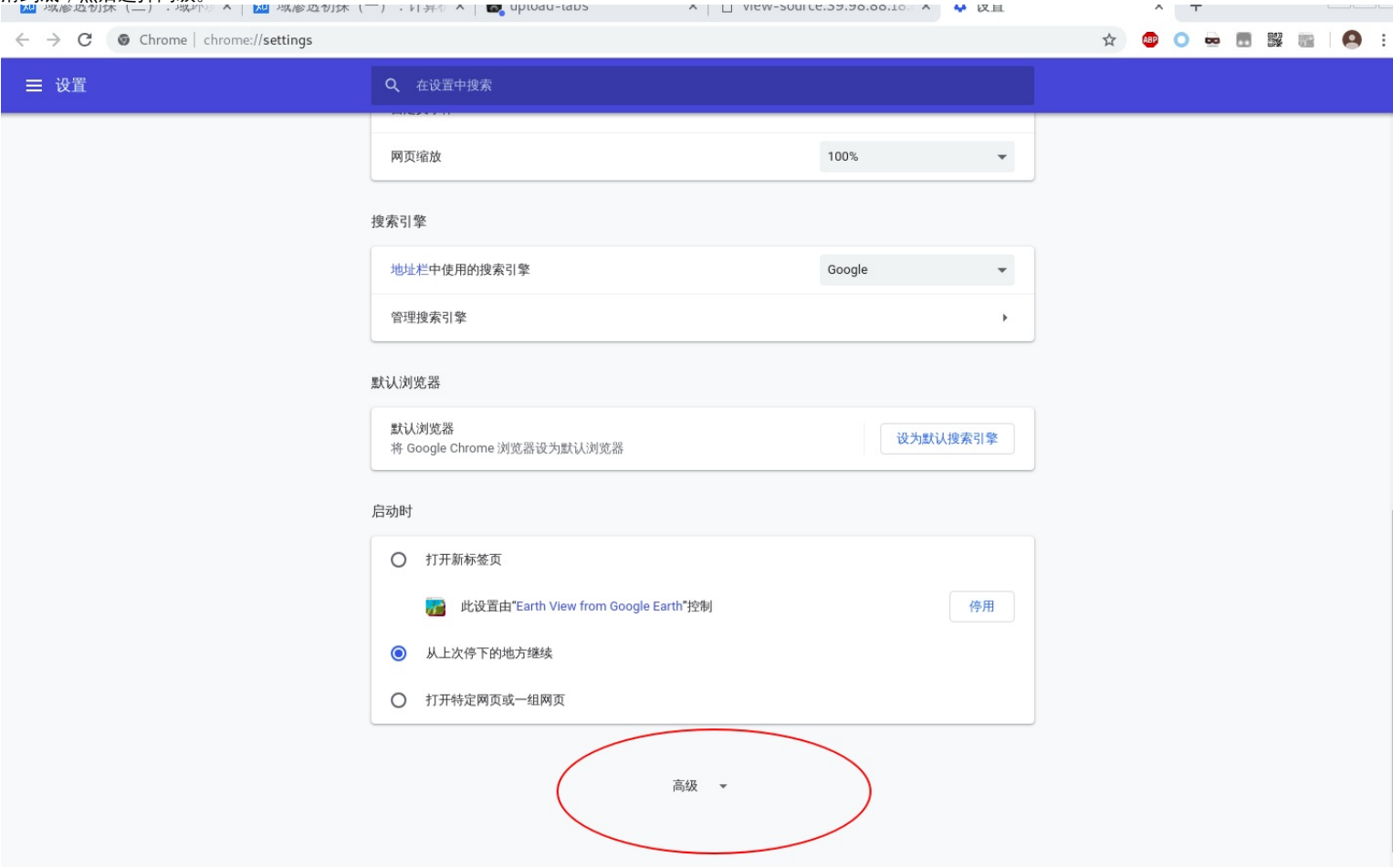
上传一个木马文件,发现禁止上传。仅允许上传图片文件。



浏览器禁止js运行。谷歌浏览器，点击右上角的三个点点。然后选择设置。





滑到底，然后选择高级。



下滑，点击内容控制。






















将一些系统信息和网页内容发送至 Google	
将使用情况统计信息和崩溃报告自动发送给 Google	
使用网络服务帮助解决拼写错误 将您在浏览器中输入的内容发送给 Google，以便提供更智能的拼写检查功能	
随浏览流量一起发送“不跟踪”请求	
允许网站检查您是否已保存付款方式	
管理证书 管理 HTTPS/SSL 证书和设置	
内容设置 控制网站可使用的信息以及可向您显示的内容	
清除浏览数据 清除浏览记录、Cookie、缓存及其他数据	

语言

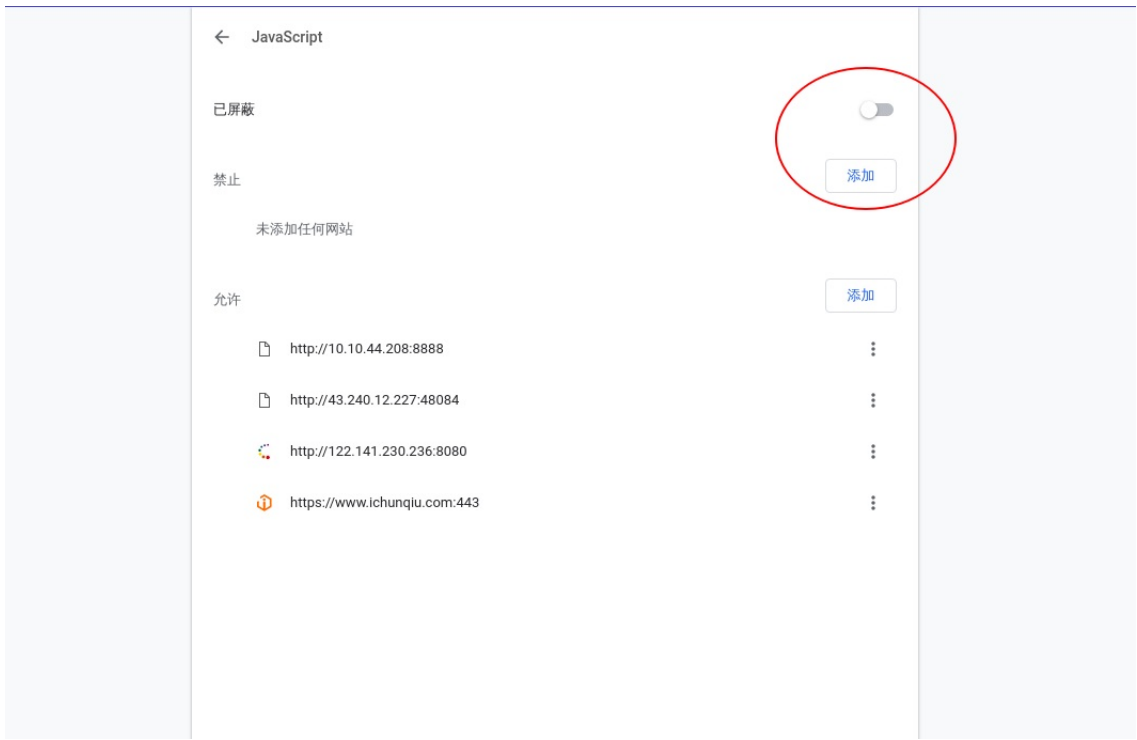
语言	
拼写检查 英语（美国）	

下载内容

点击javascript.

← 内容设置	
 Cookie 允许网站保存和读取 Cookie 数据	
 位置 使用前先询问	
 摄像头 使用前先询问	
 麦克风 使用前先询问	
 通知 已屏蔽	
 JavaScript 允许	
 Flash 先询问	
 图片 全部显示	
 弹出式窗口和重定向 已屏蔽	
 广告 已禁止会展示侵扰性或误导性广告的网站显示广告	

点击右上方的开关。将其屏蔽掉。



然后我们再一次的上传木马，上传成功。



右键查看源码，发现文件路径

```
40 </div>
41
42 <div id="upload_panel">
43
44
45 <h3>任务</h3>
46 <p>上传一个<code>webshell</code>到服务器。</p>
47
48
49 <h3>上传区</h3>
50 <form enctype="multipart/form-data" method="post" onsubmit="return checkfile()">
51 <p>请选择要上传的图片：<p>
52 <input class="input_file" type="file" name="upload_file"/>
53 <input class="button" type="submit" name="submit" value="上传"/>
54 </form>
55 <div id="msg">
56 </div>
57 <div id="img">
58
59 </div>
60
61
62 </div>
63
64 <div id="footer">
```

## 2.3 课后习题

1. [html查看源码](#)
2. [查看css文件](#)
3. [修改查看js](#)
4. [禁用js](#)

禁用js靶场地址：<http://39.98.88.18:8080/upload/Pass-01/index.php>