一、靶场简介

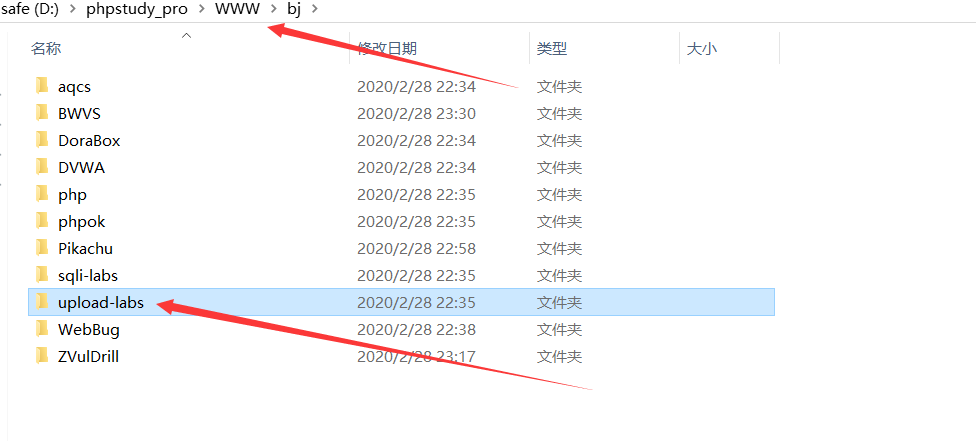
upload-labs是一个使用php语言编写的，专门收集渗透测试和CTF中遇到的各种上传漏洞的靶场。旨在帮助大家对上传漏洞有一个全面的了解。目前一共20关，每一关都包含着不同上传方式。

1. 靶场安装

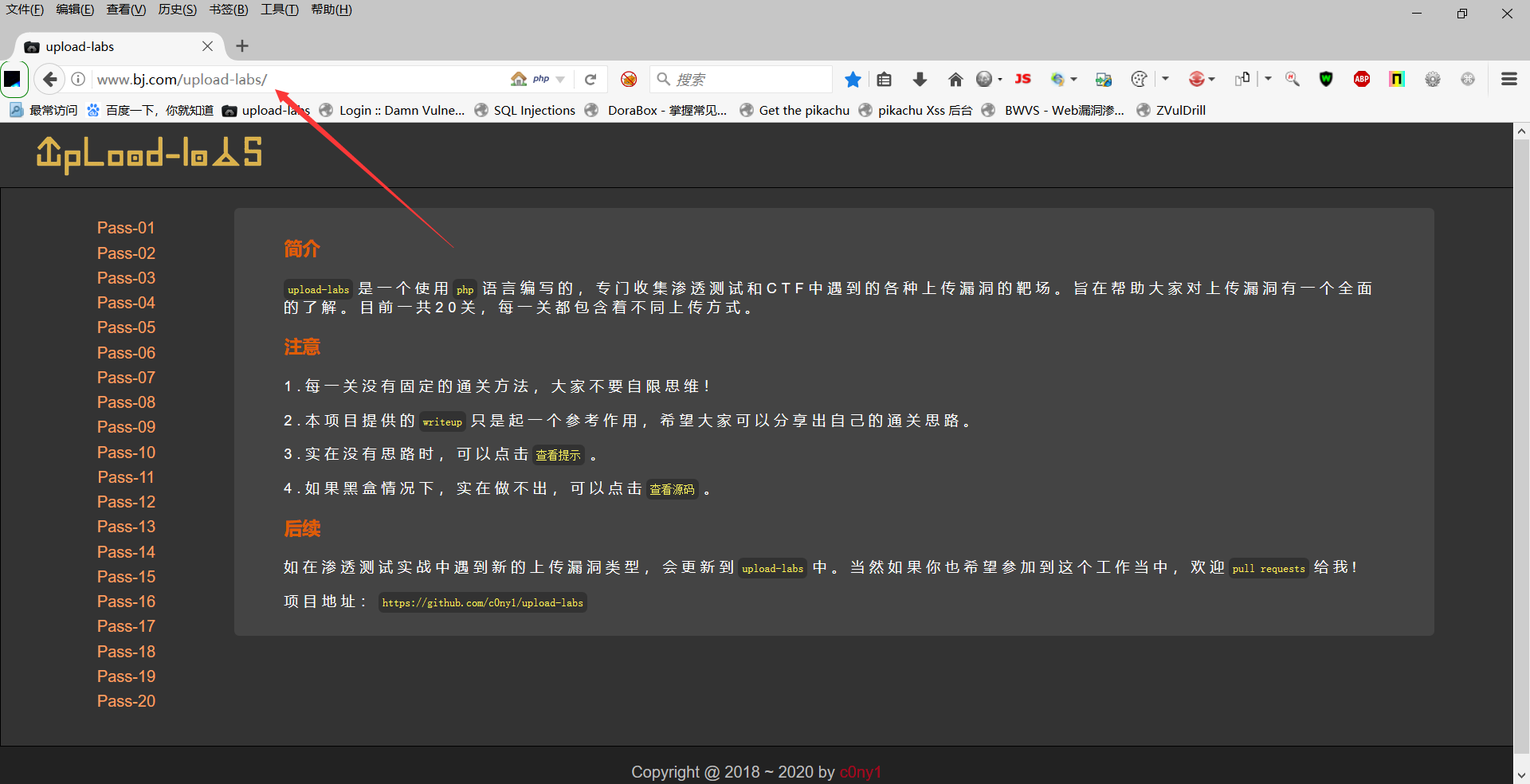
由于upload-labs 是一个由php 语言编写的靶场，因此需要在php环境下进行使用。我这里推荐用phpstudy集成环境。

靶场项目地址：<https://github.com/c0ny1/upload-labs>

将源码下载下来之后，在www目录里面新建一个文件夹来存放。



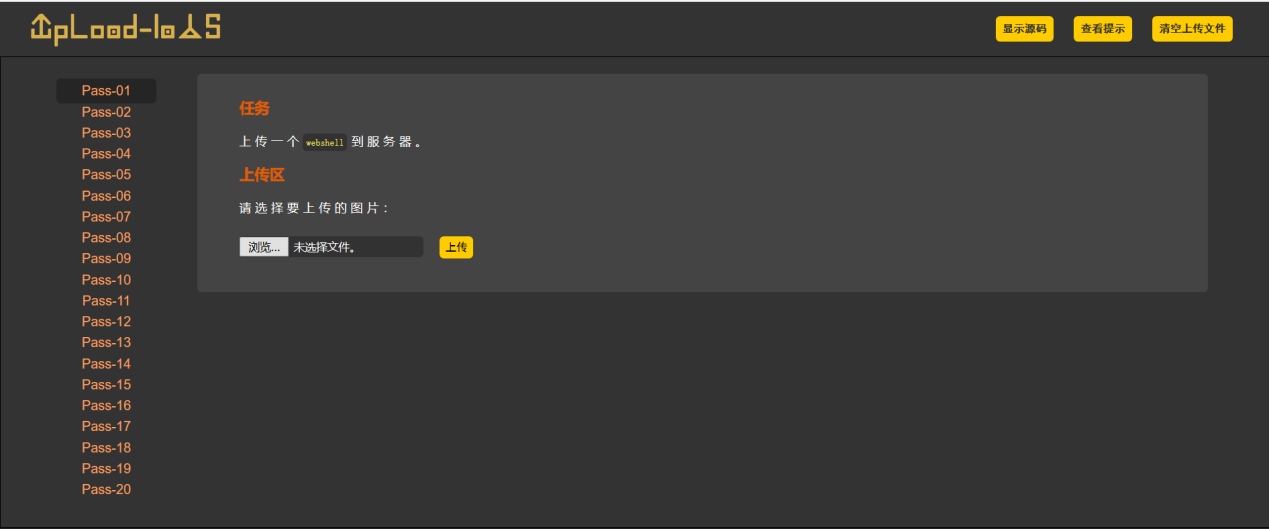
然后直接在浏览器打开即可



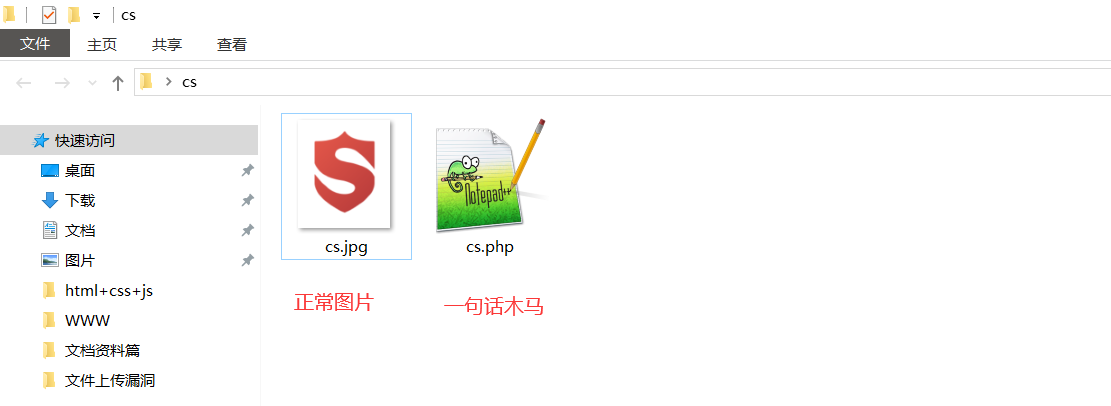
本文先详细讲解一下第一关，后续文章中有些相关操作就会一笔带过了。

Pass-01

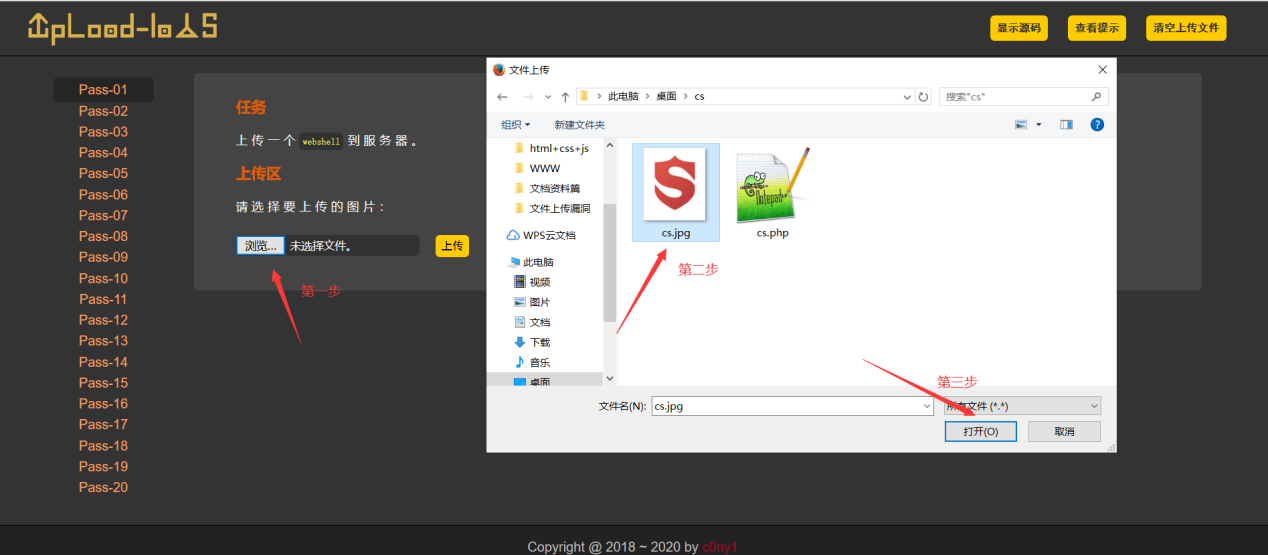
我们打开第一关可以看到一个上传点



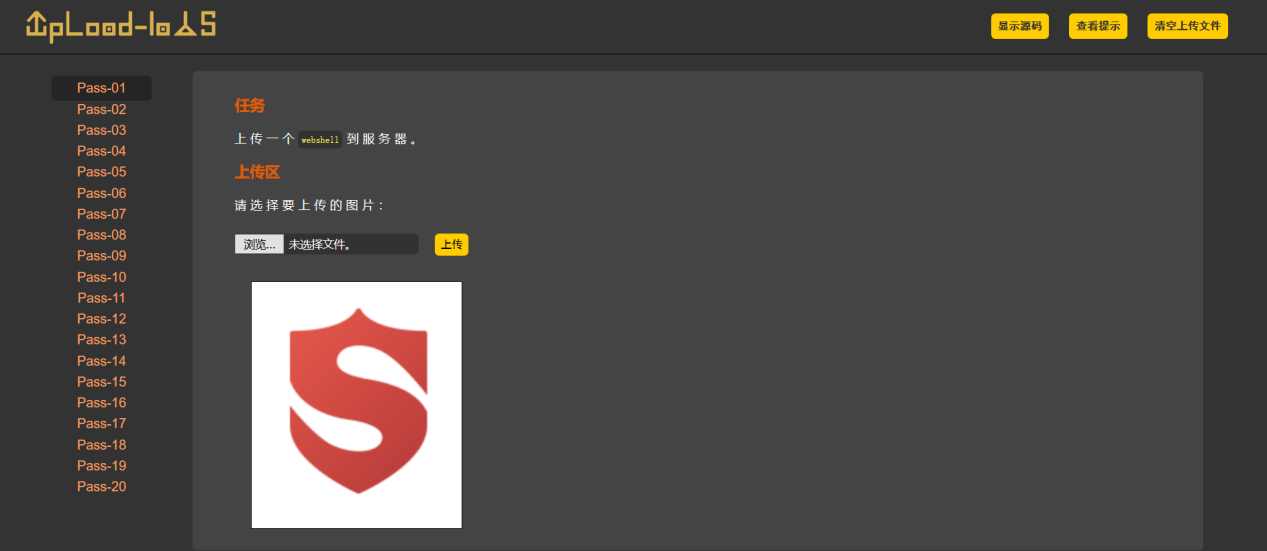
既然说了选择上传的图片，那么我们先上传一张正常的图片试试。这里我准备了一张测试用图



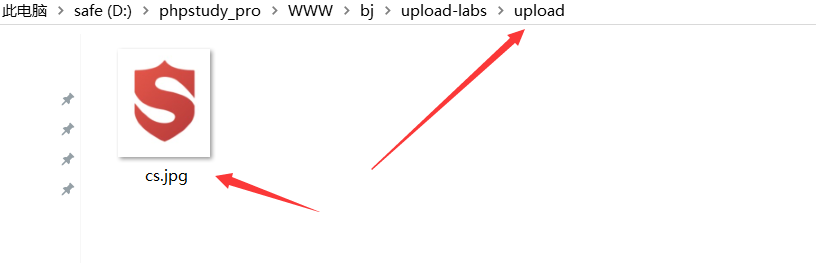
直接选中



然后点击上传



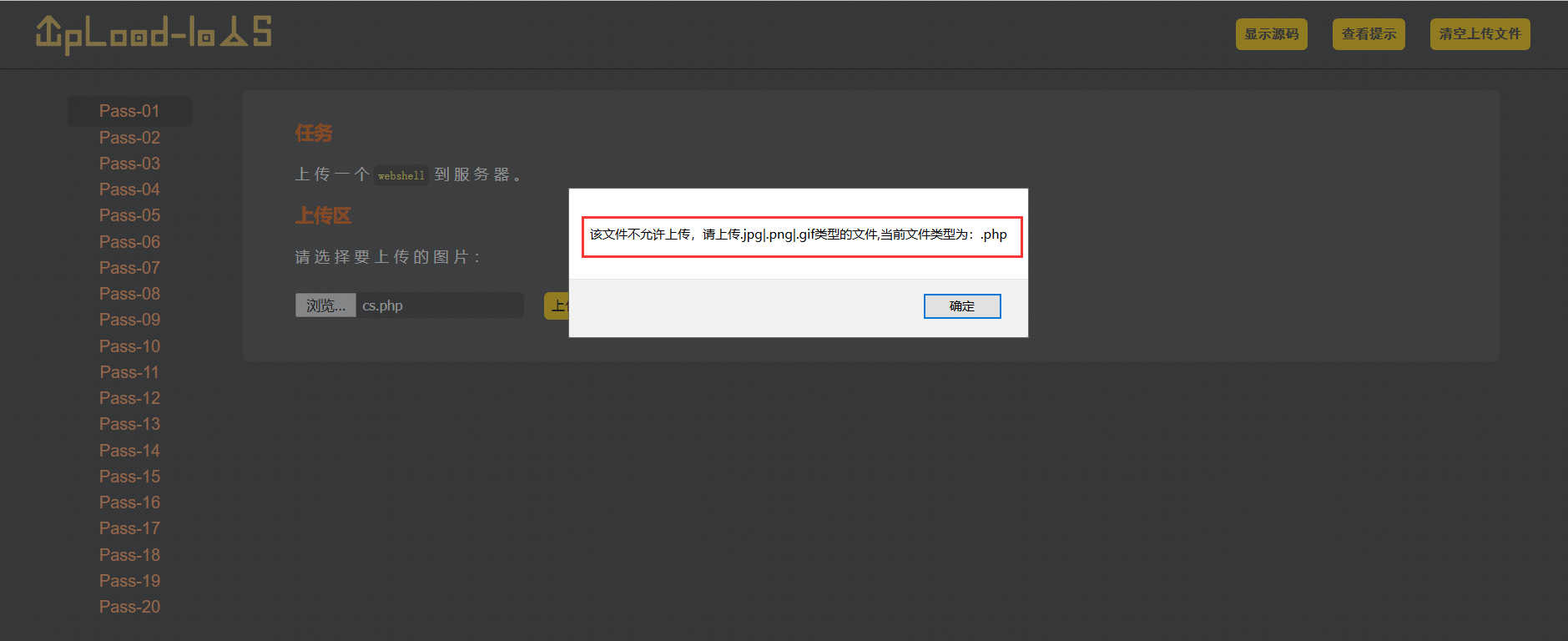
上传成功后不仅页面会有显示，在网站目录的upload文件夹下也会多出来刚刚上传的图片



正常图片可以上传，那么我们直接上传一句话木马试试（一句话木马的相关知识请自行了解）

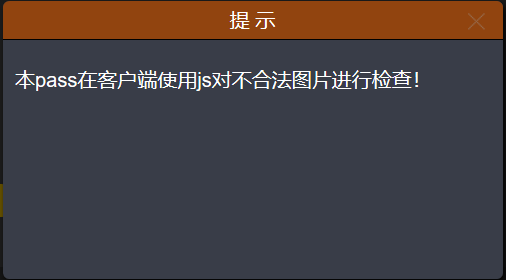


图中文件就是php一句话，点击上传却发现无法上传

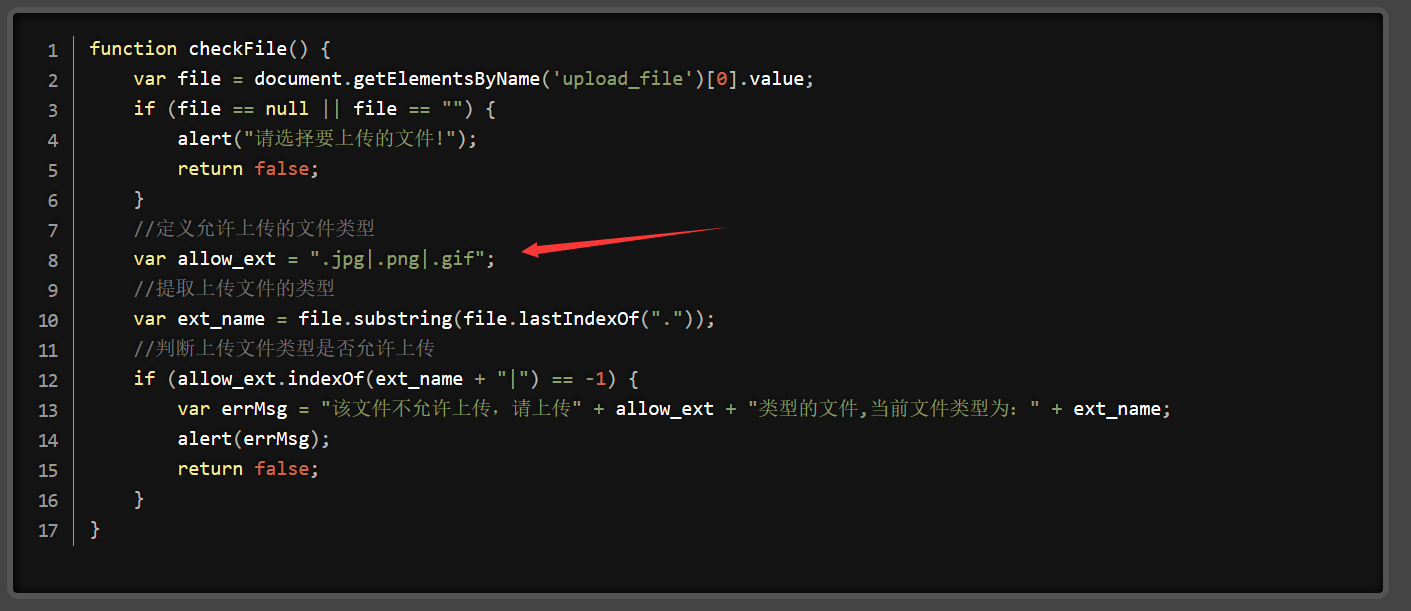


弹出错误：该文件不允许上传，请上传.jpg|.png|.gif类型的文件,当前文件类型为：.php

无法直接上传shell，那么肯定是有检测手段咯，查看提示我们可以知道，第一关是用的js前端校验



既然是js前端校验，那么我们就来看看源代码，瞅瞅js都做了什么



可以看到js定义了一个白名单，只允许上传.jpg|.png|.gif格式的文件，并且对上传文件的后缀名同白名单做了对比校验。

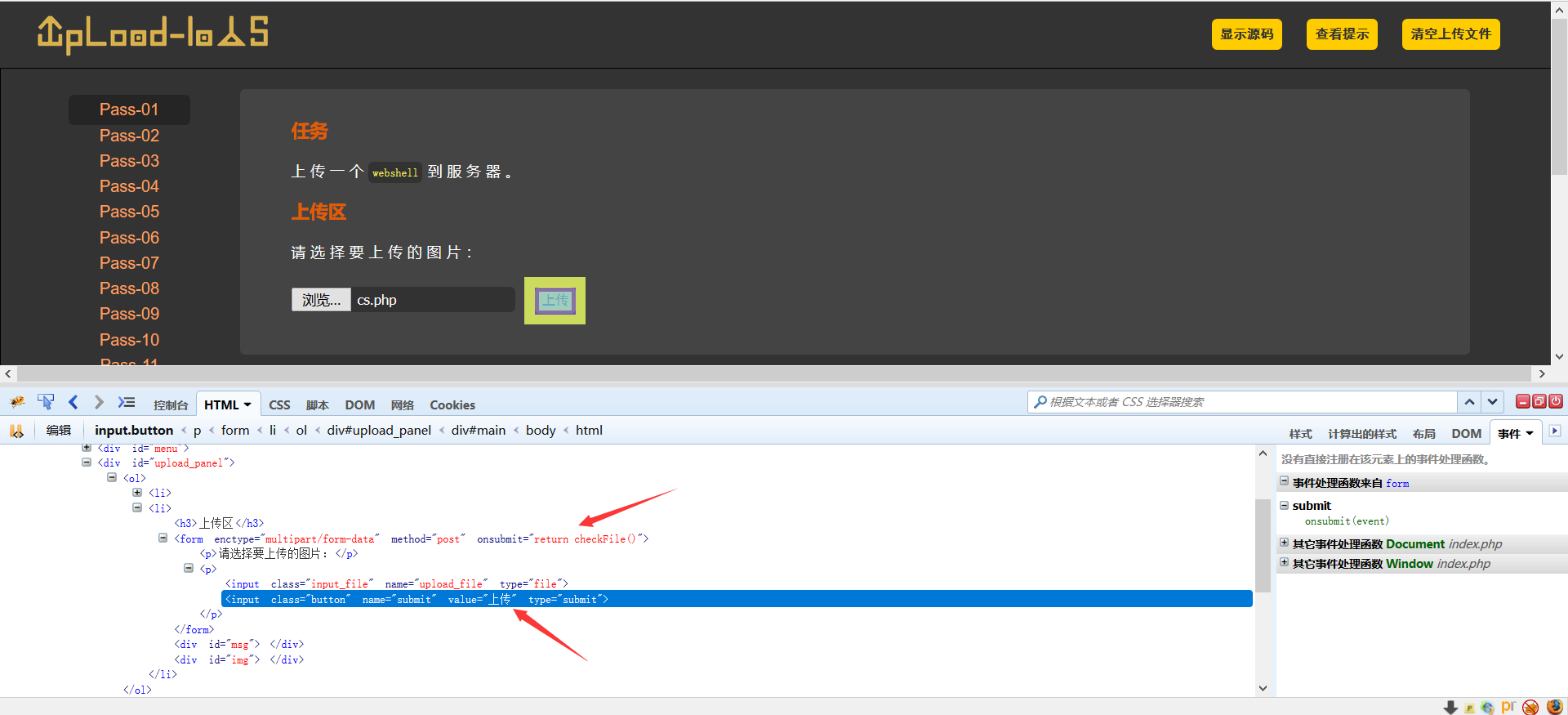
对于js校验，我了解的有三种方式可以进行绕过

1. 删除js校验代码
2. 更改js校验中的白名单
3. 先改文件后缀名通过js校验再抓包更改后缀上传至服务器

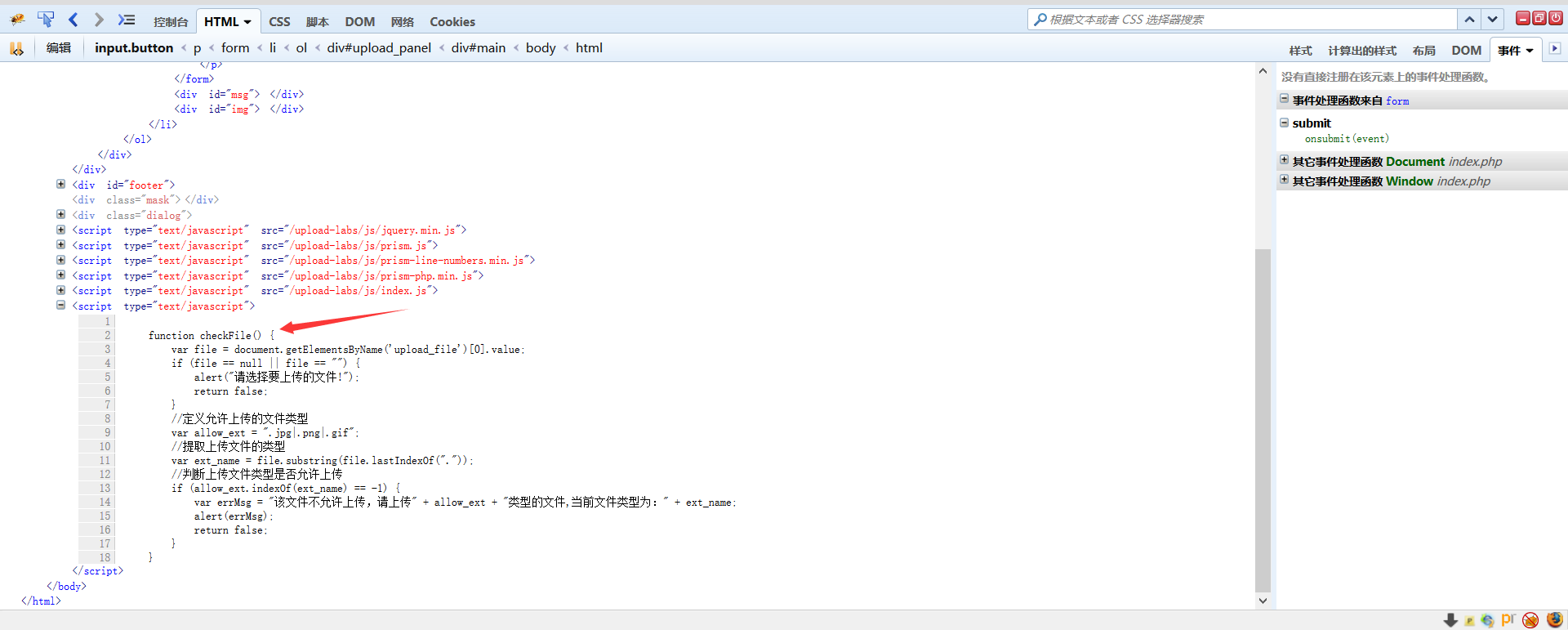
下面对三种方式进行详细的讲解

1. 删除js校验代码

我们既然知道了是什么在阻止我们上传shell，那么干掉它不就可以了吗？



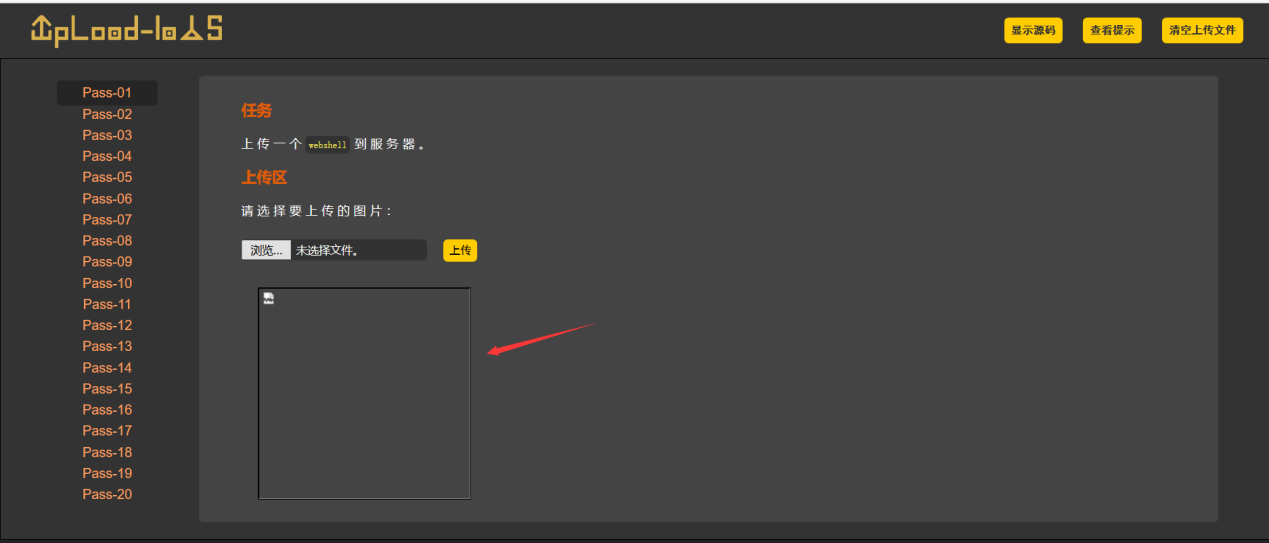
通过审查元素我们可以发现上传这里是调用了checkFile函数的，那么我们先找到这个函数（肯定是在script标签里，不要问为什么）



往下翻我们在最后一个标签中看到了这个函数，接下来就是对它的校验功能进行删除了



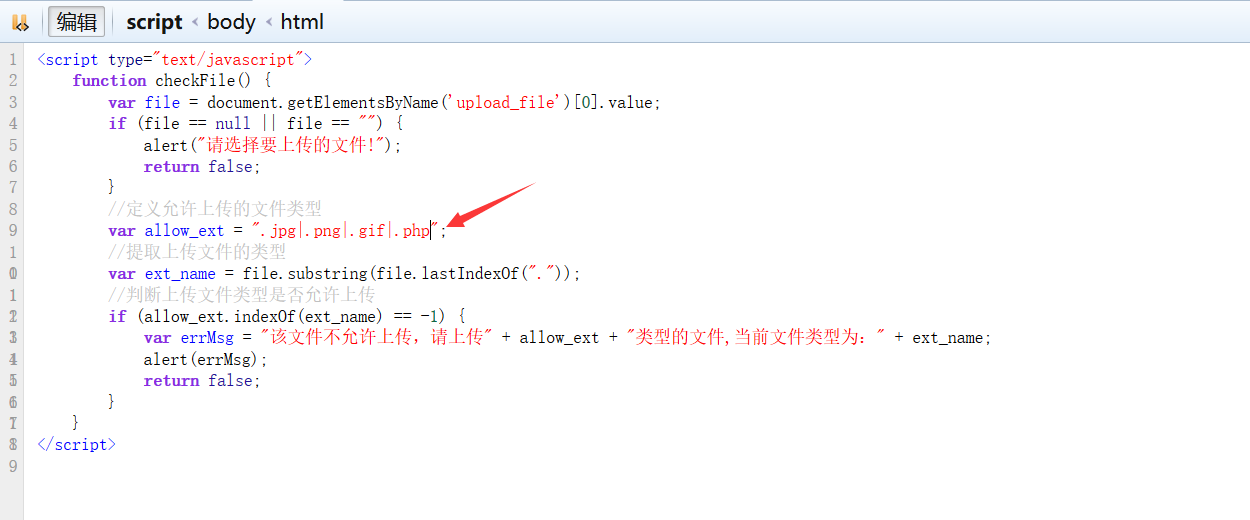
删除之后在进行上传发现成功了。





二、更改js校验中的白名单

同样是对js代码进行操作，只不过不同于上面的直接删除，这里是在js校验的白名单中把php文件格式添加上



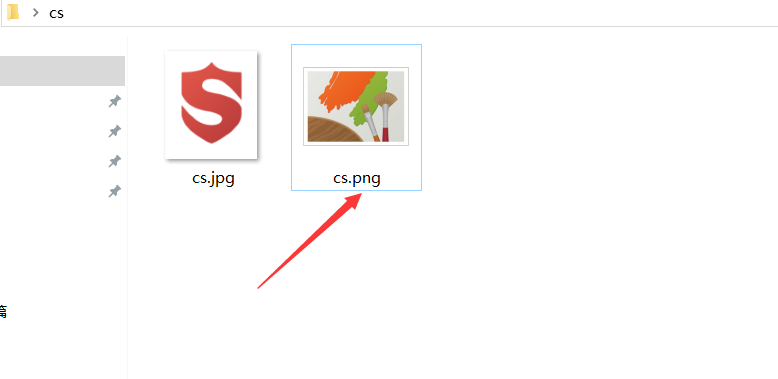
修改之后同样可以绕过js检测。

1. 先改文件后缀名通过js校验再抓包更改后缀上传至服务器

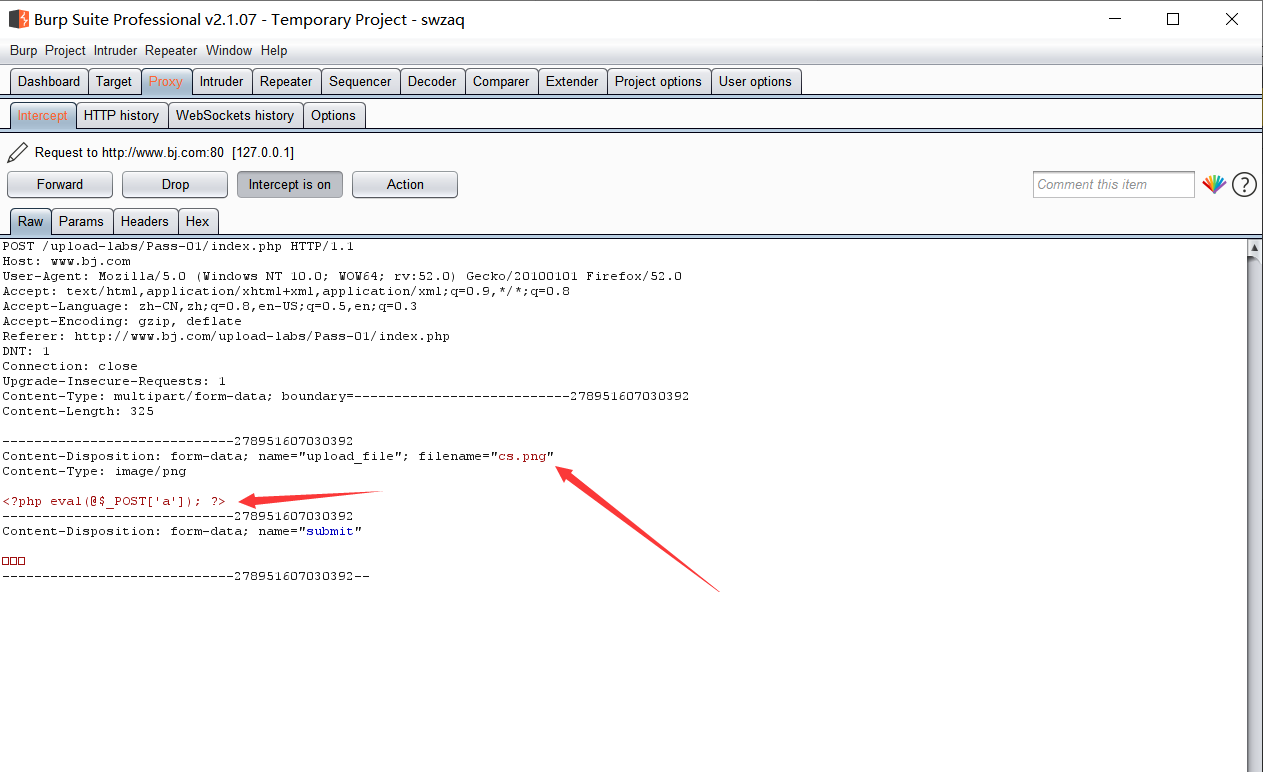
这种方式稍微复杂一点，对小白选手来说，我尽量写得详细些。

这里值得说一下的是，我们的浏览器在向服务器请求带有上传点的这个页面，并且服务器返回了页面内容之后，它们之间的连接就断开了。js检测完全就是浏览器本地的行为。当选择的文件在浏览器检测允许上传之后才会上传至服务器，如果检测不通过不会和服务器有任何的数据交互。

第三种方法就是先把shell的后缀名修改为浏览器js校验允许的类型



这里我修改成了png格式。然后对浏览器进行抓包分析。点击上传按钮之后浏览器会先进行js校验，通过后才会将数据发向服务器



因为shell的后缀名是修改为了png格式的，所以可以通过检测，这里抓到的数据包可以看到浏览器向服务器上传的的文件后缀还是png格式，我们只需将它改为php格式的，就可以将shell成功上传到服务器

