**bluecms渗透测试\_吴汶航**

**目录**

一、信息收集

二、利用burpsuit进行漏洞扫描

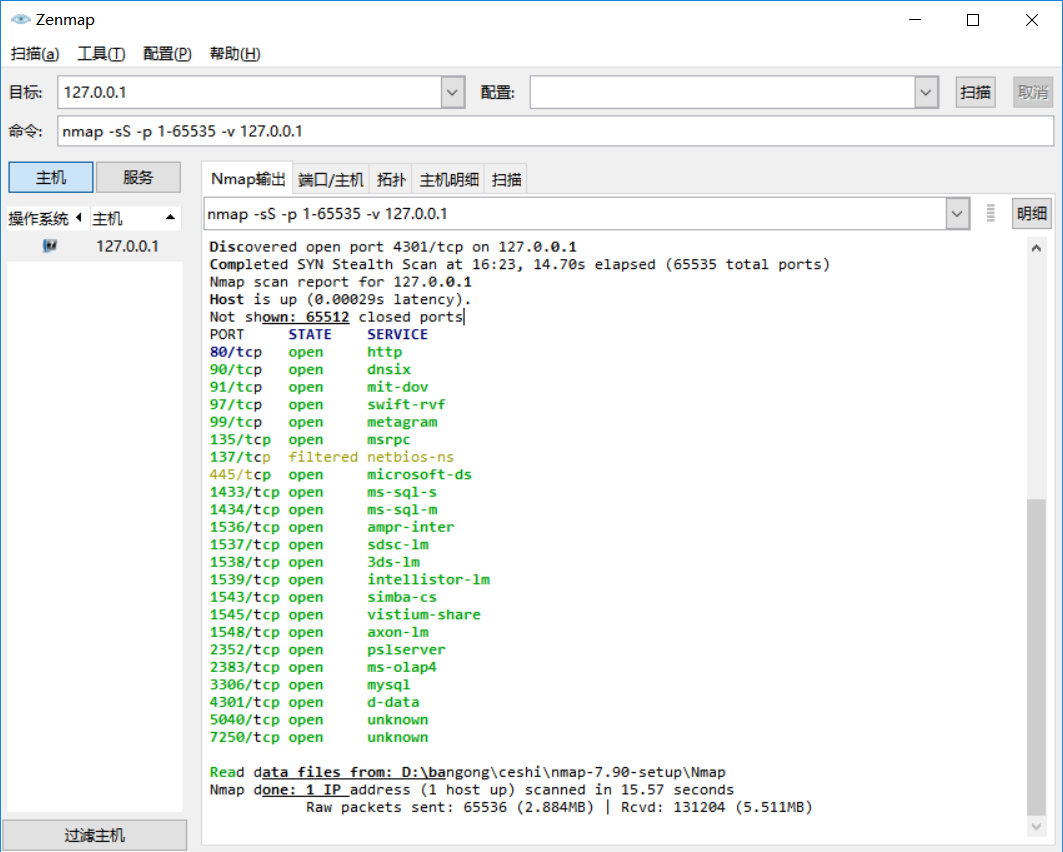
三、爬取网站

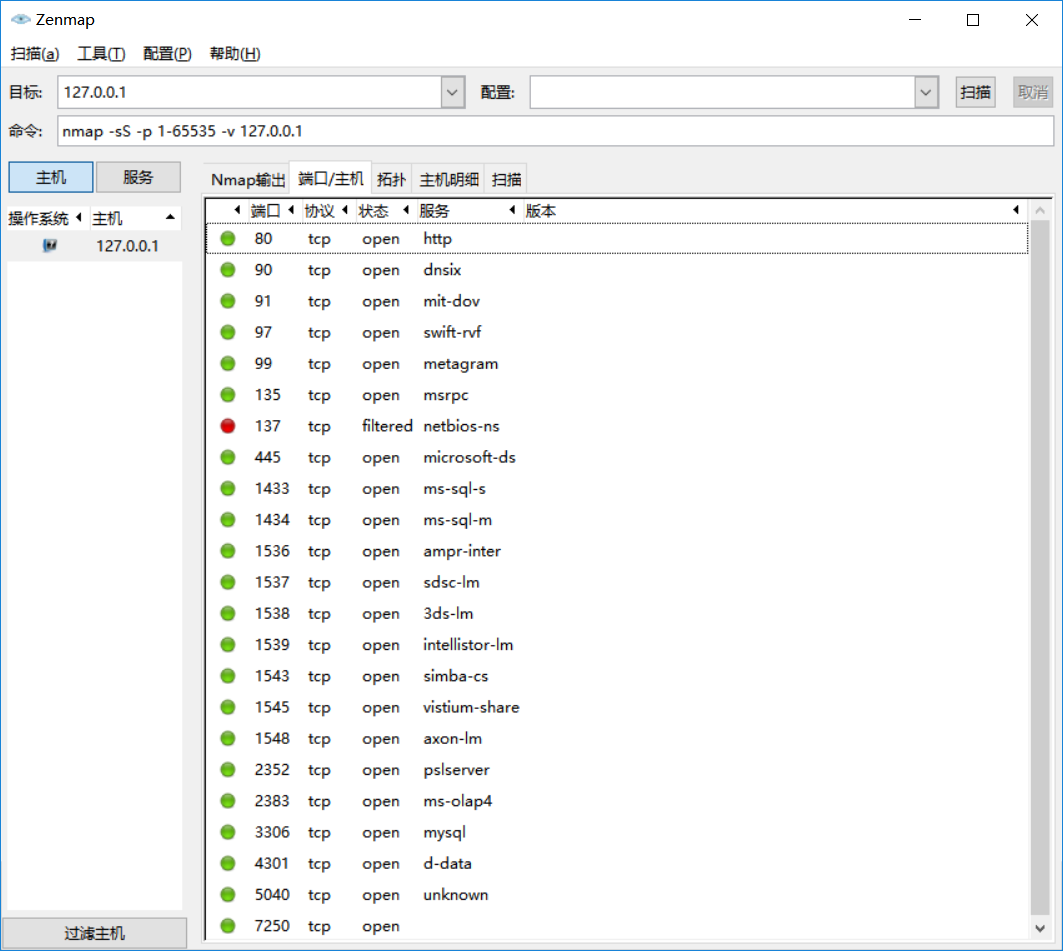
四、自动化攻击

五、对网站进行手工测试

一、信息收集：

1.1使用Zenmap扫描端口



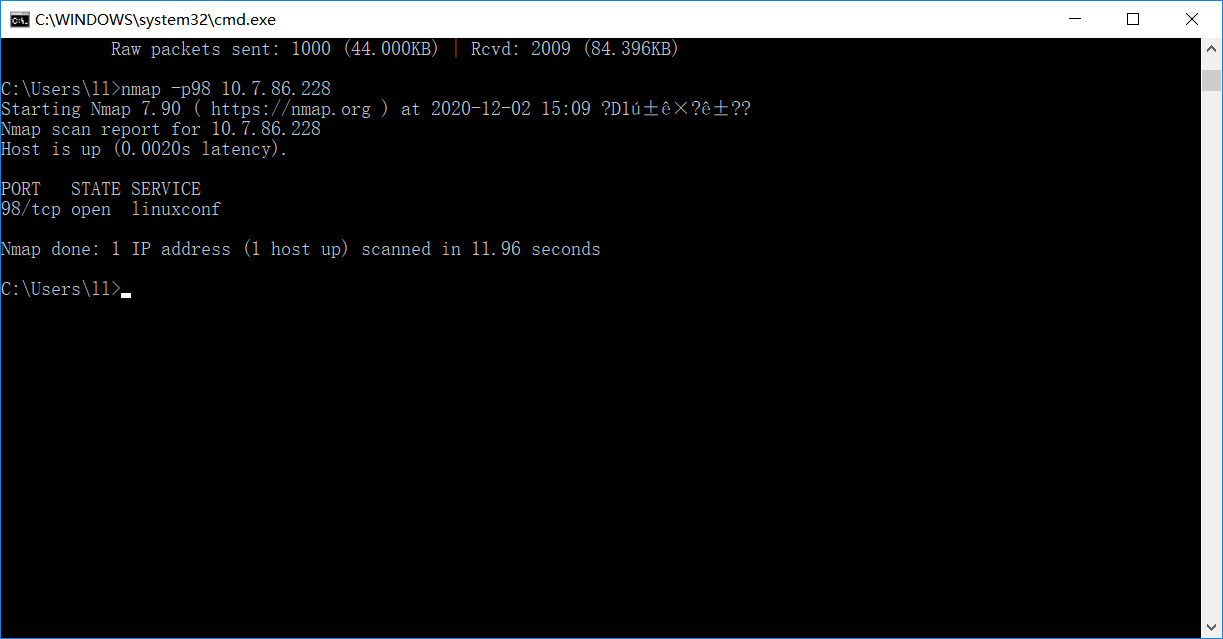


1.2对burpsuit创建网站，分配端口号：98

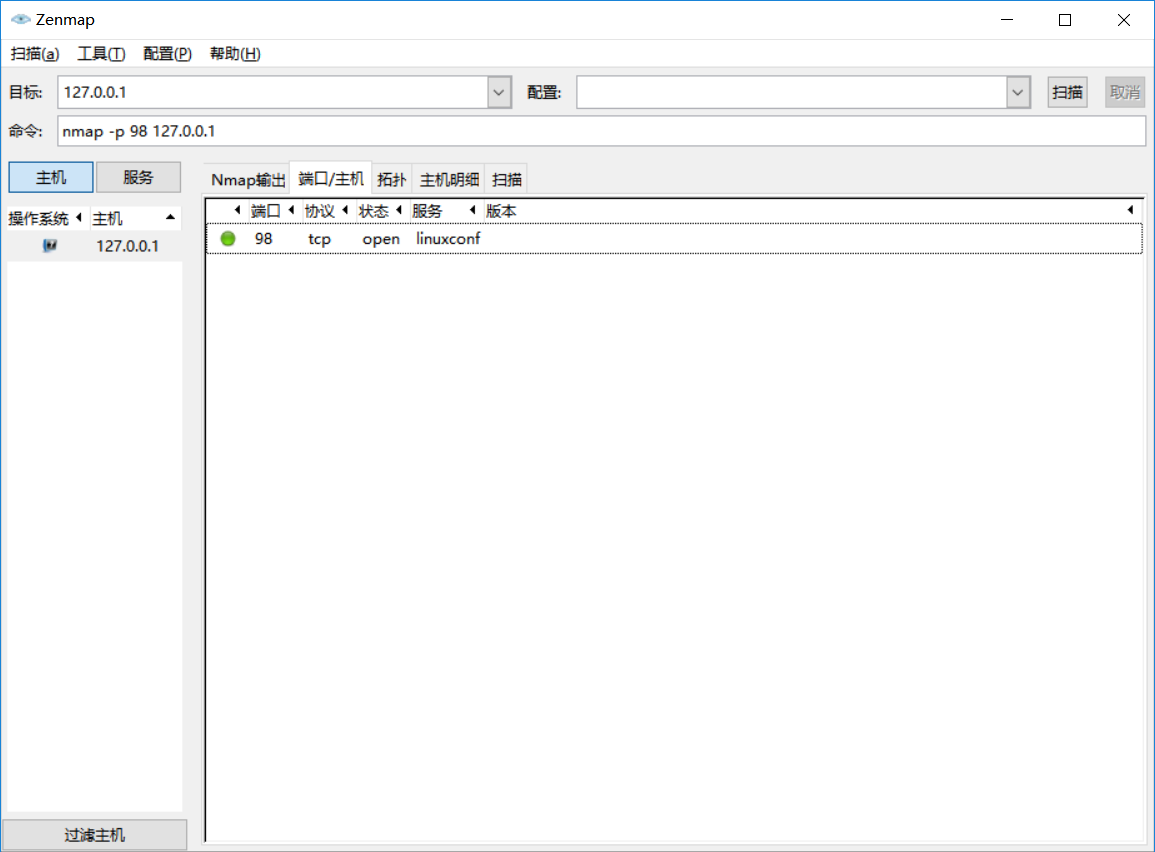


1.3扫描98端口：

1.3.1命令行：



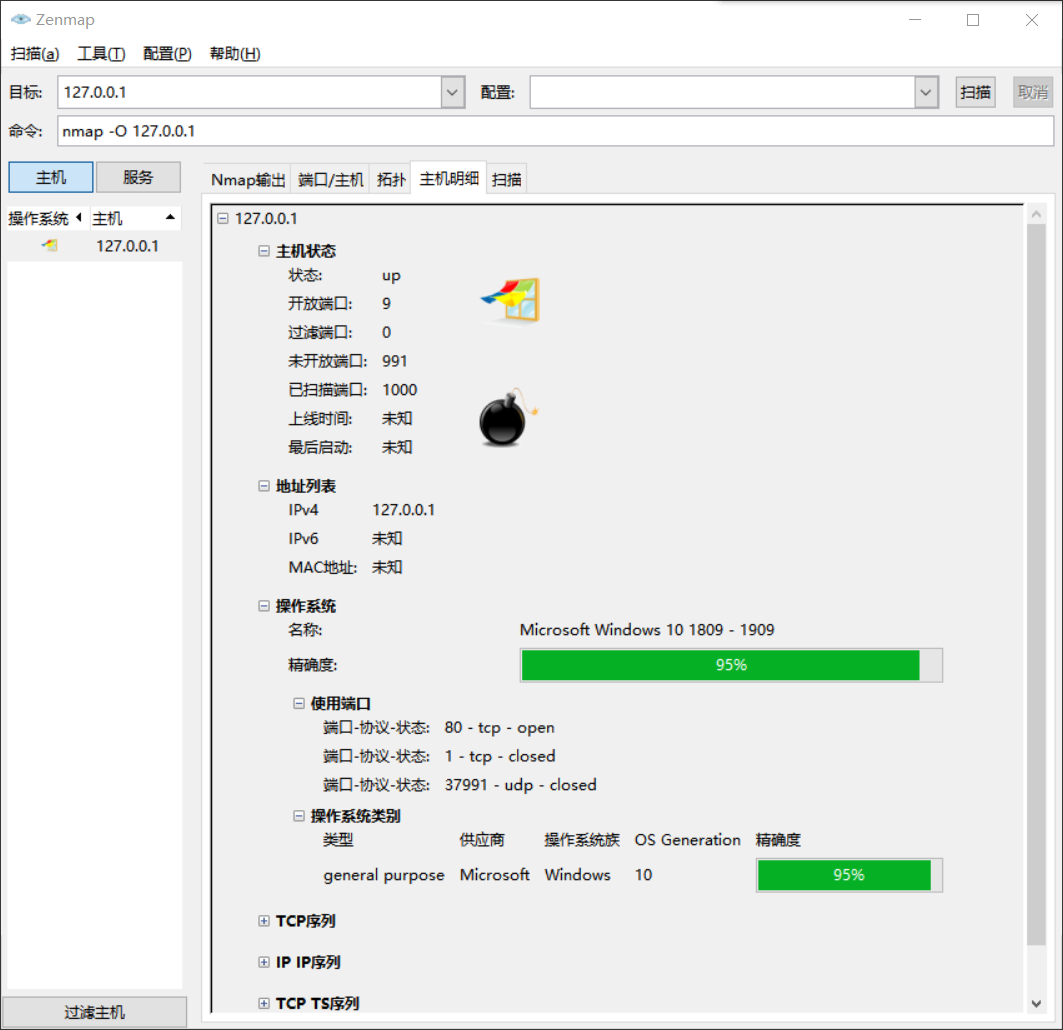
1.3.2借助Zenmap工具扫描：



1.3.3

根据扫描结果，可知，98端口号此时已经得到分配且已经打开，其服务为linuxconf

1.4本机操作系统检测



扫描可得，本季操作系统为微软公司的Windows操作系统，版本是10

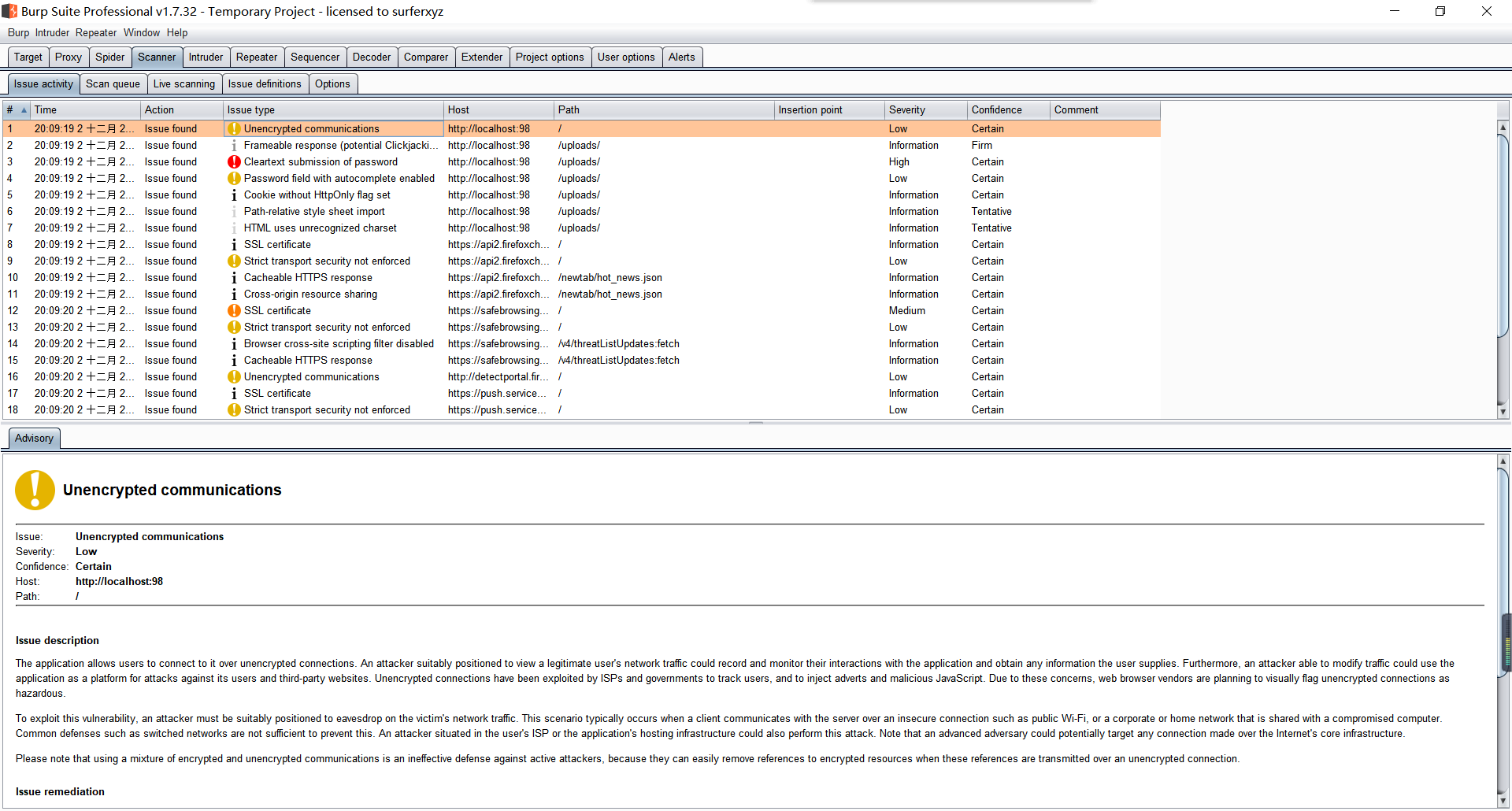
二、利用burpsuit进行漏洞扫描

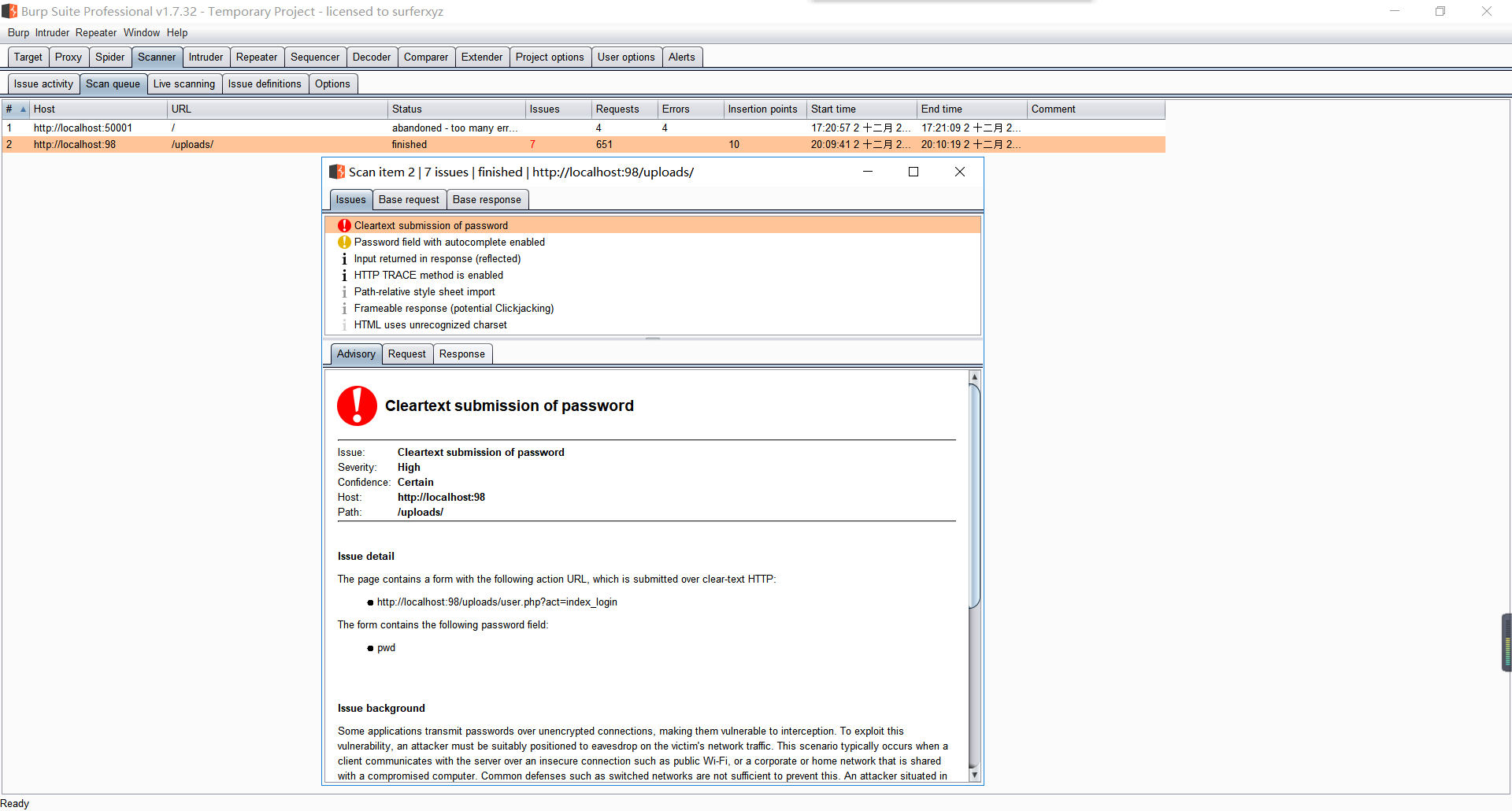
对浏览器设置代理，打开burpsuit

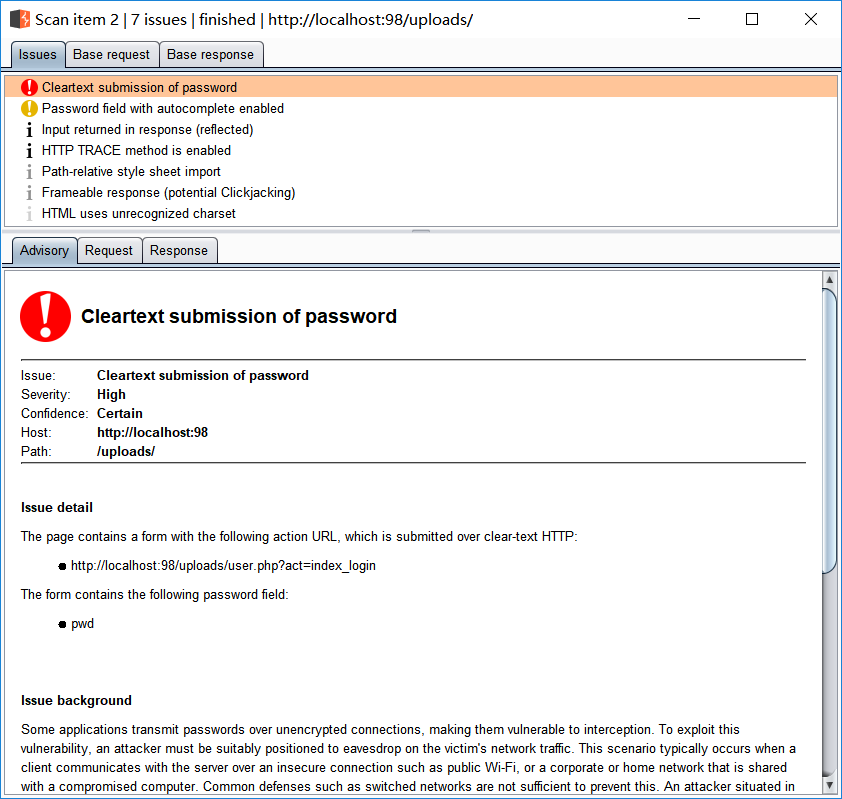
2.1对前端扫描漏洞：

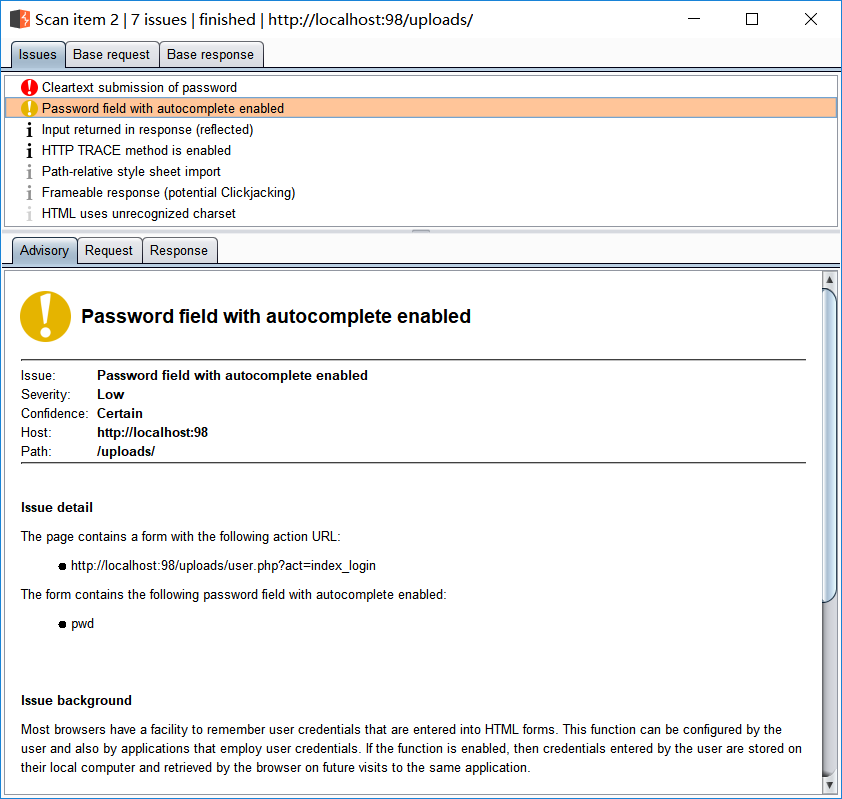
浏览器访问http://localhost:98/uploads/，拦截到之后扫描

扫描结果：









根据扫描结果可知：

High级别漏洞一个：

Cleartext submission of password【明文提交密码】

Low级别漏洞一个：

password field with autocomplete enabled【密码字段自动完成启用】

Information级别漏洞五个：

Input returned in response (reflected)【相应返回的输入】

HITP TRACE method is enabled【启用HTIP跟踪方法】

Path-relative style sheet import【路径相关样式表导入】

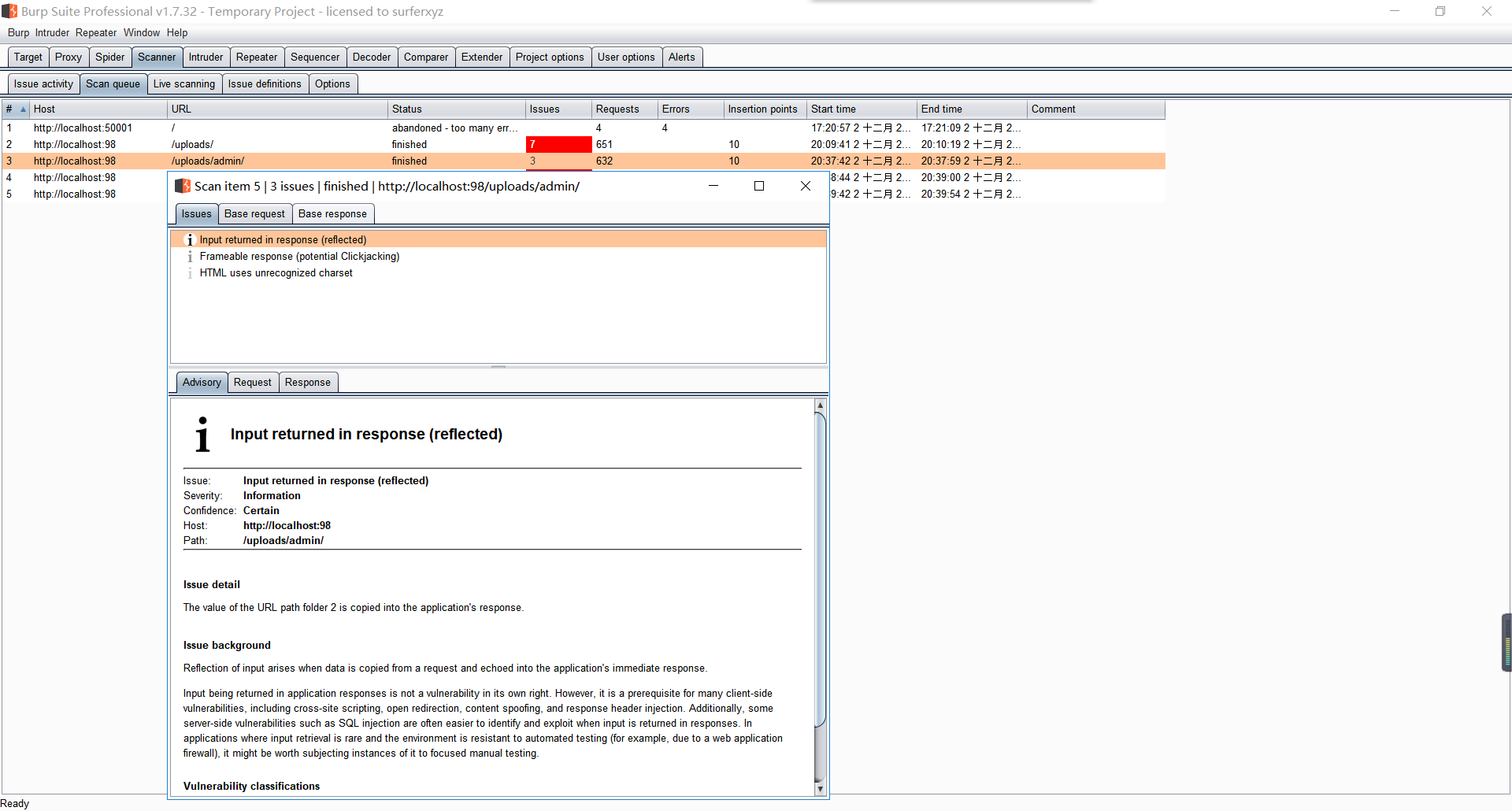
Frameable response (potential Clickjacking)【可Frameable响应（潜在的点击劫持）】

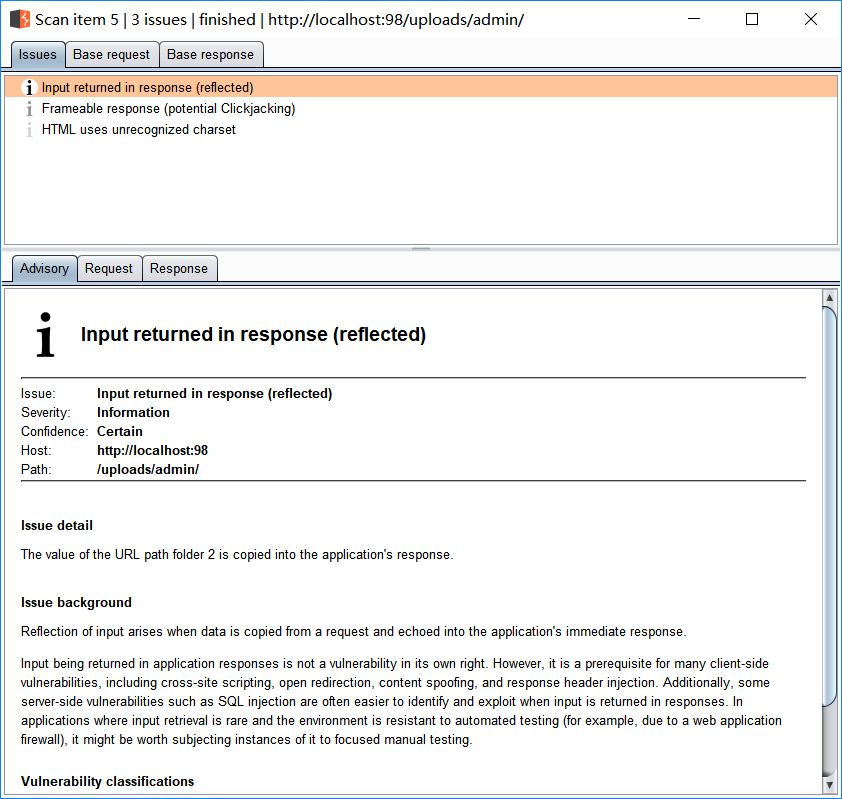
HTML uses unrecognized charset【HTML使用无法识别的字符集】

2.2对后台扫描漏洞：

浏览器访问http://localhost:98/uploads/admin，拦截到之后扫描

扫描结果：





根据扫描结果可知：

Information级别漏洞三个：

Input returned in response (reflected)【相应返回的输入】

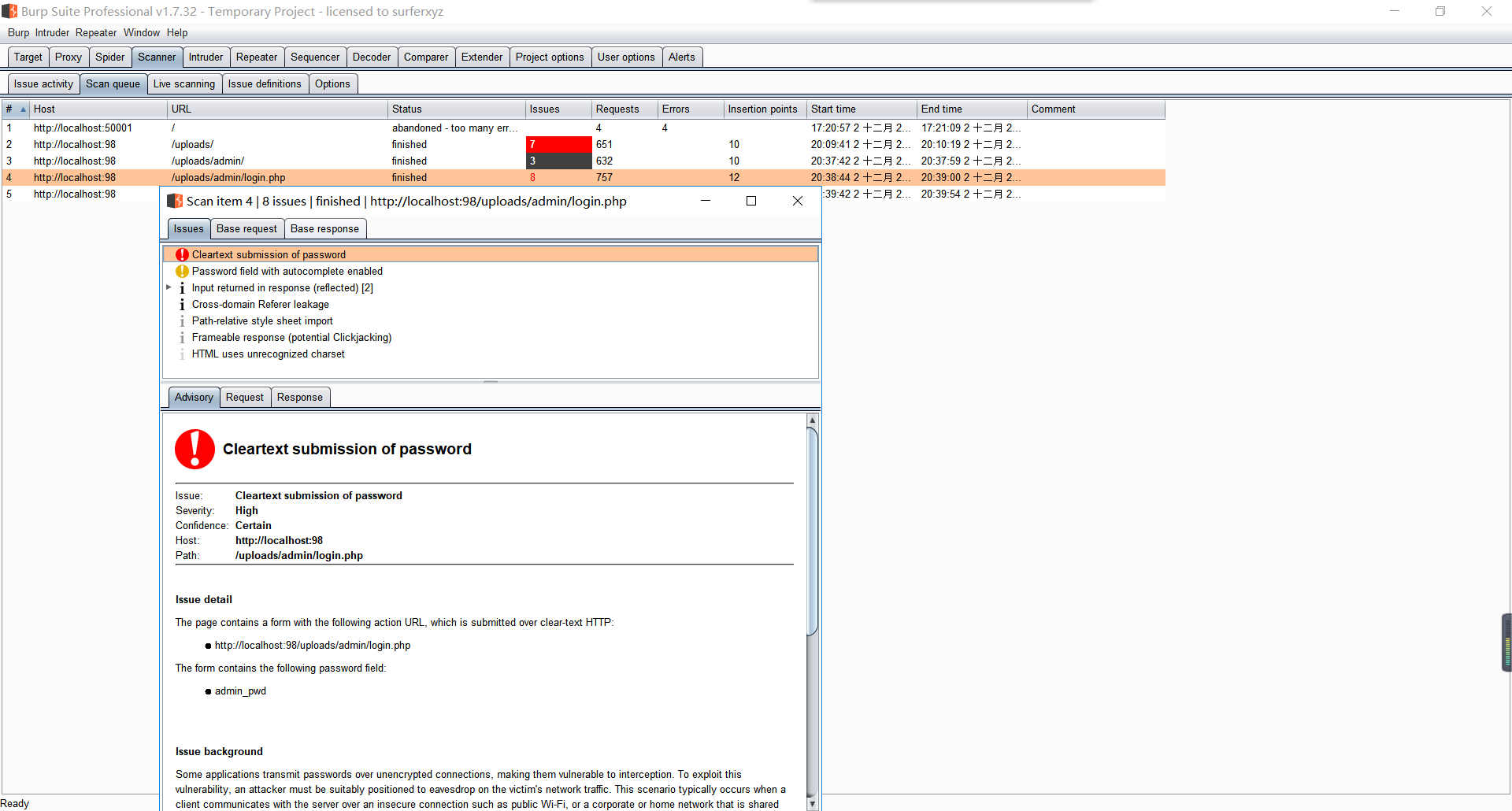
Frameable response (potential Clickjacking)【可Frameable响应（潜在的点击劫持）】

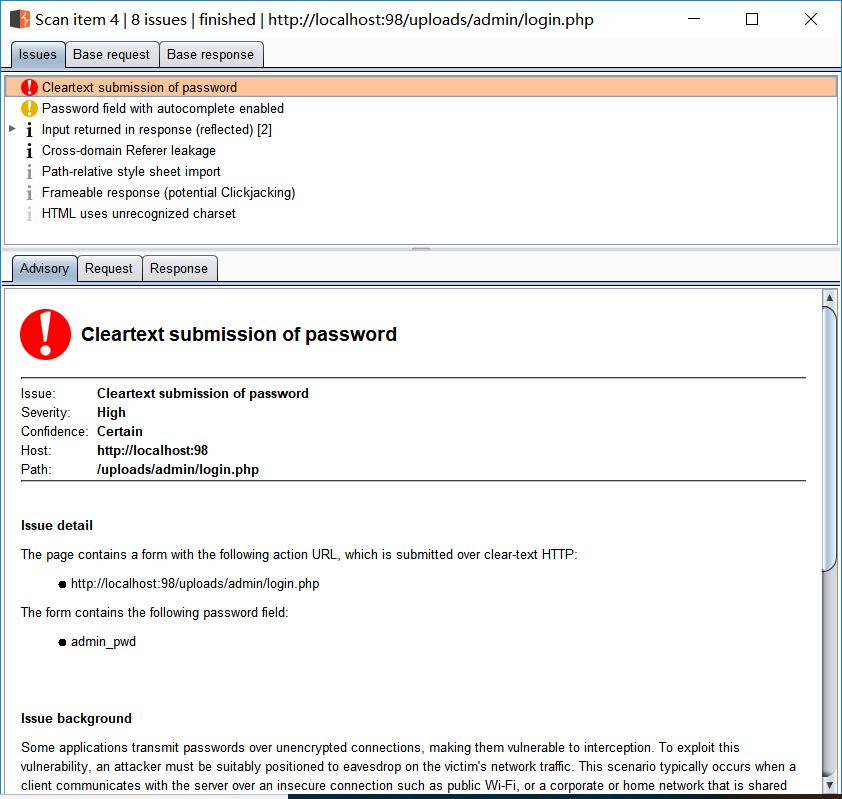
HTML uses unrecognized charset【HTML使用无法识别的字符集】

2.3对后台登录扫描漏洞：

浏览器访问http://localhost:98/uploads/admin/login.php，拦截到之后扫描

扫描结果：





根据扫描结果可知：

High级别漏洞一个：

Cleartext submission of password【明文提交密码】

Low级别漏洞一个：

password field with autocomplete enabled【密码字段自动完成启用】

Information级别漏洞五个：

Input returned in response (reflected)【相应返回的输入】

HITP TRACE method is enabled【启用HTIP跟踪方法】

Path-relative style sheet import【路径相关样式表导入】

Frameable response (potential Clickjacking)【可Frameable响应（潜在的点击劫持）】

HTML uses unrecognized charset【HTML使用无法识别的字符集】

1. 爬取网站

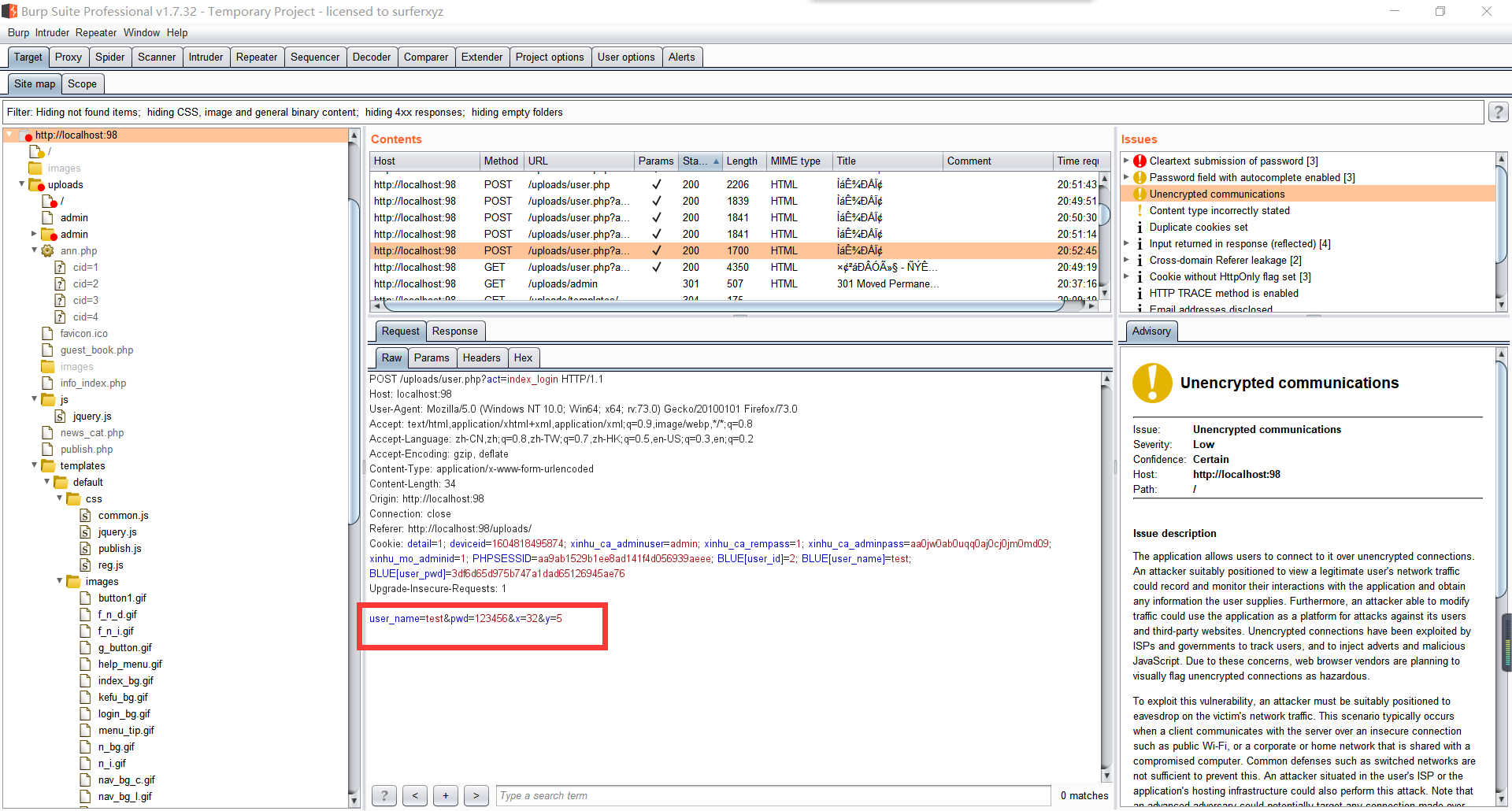
对前端登录页面（http://localhost:98/uploads/）进行爬取，其中

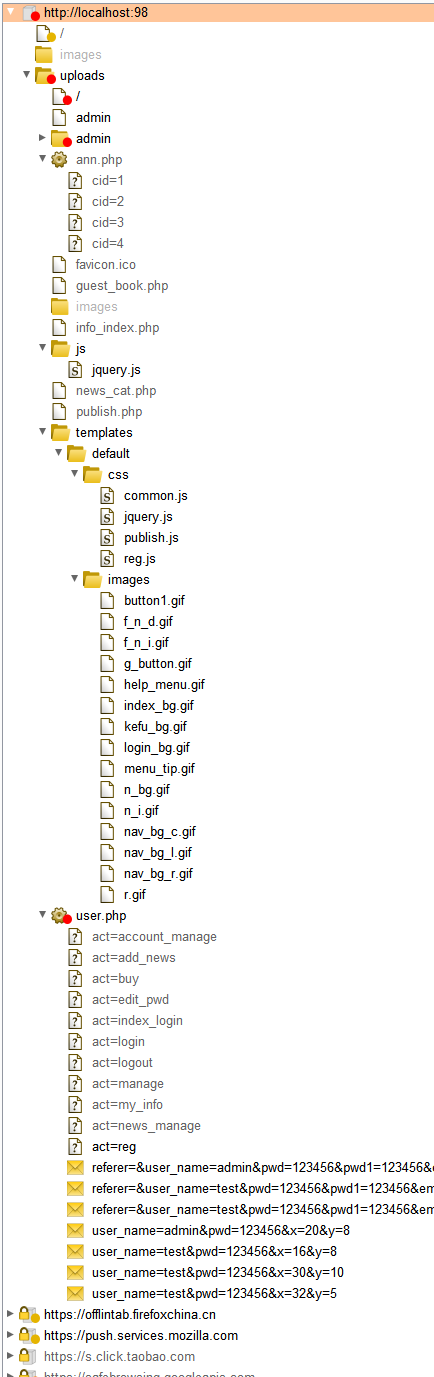
user\_name=test

pwd=123456

注意：用户为已经注册成功的用户

爬取结果：





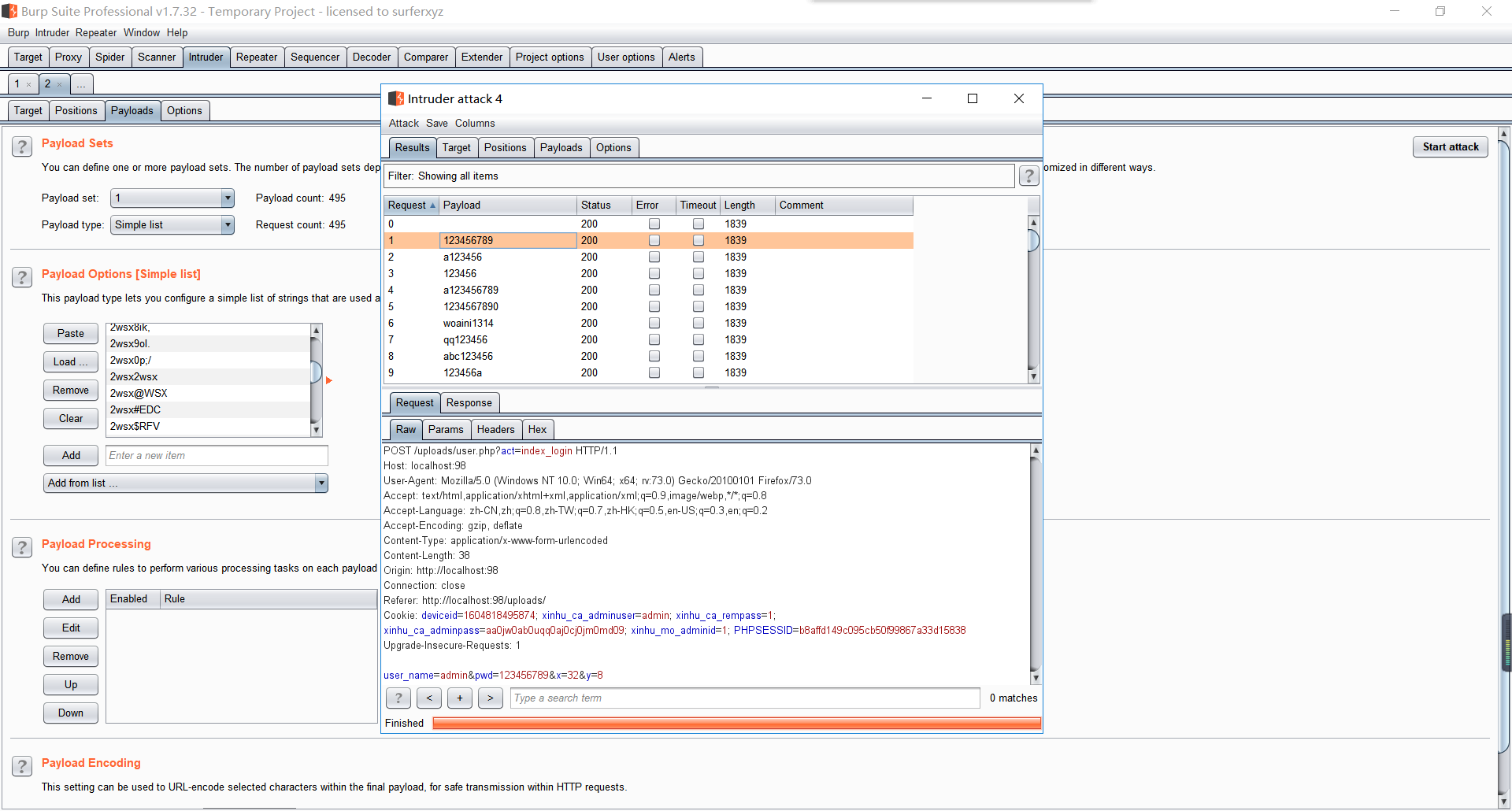
由此可以得到用户的用户名及密码，同时可以扫描到该网站的目录结果及相应的sql语句，高危

1. 自动化攻击

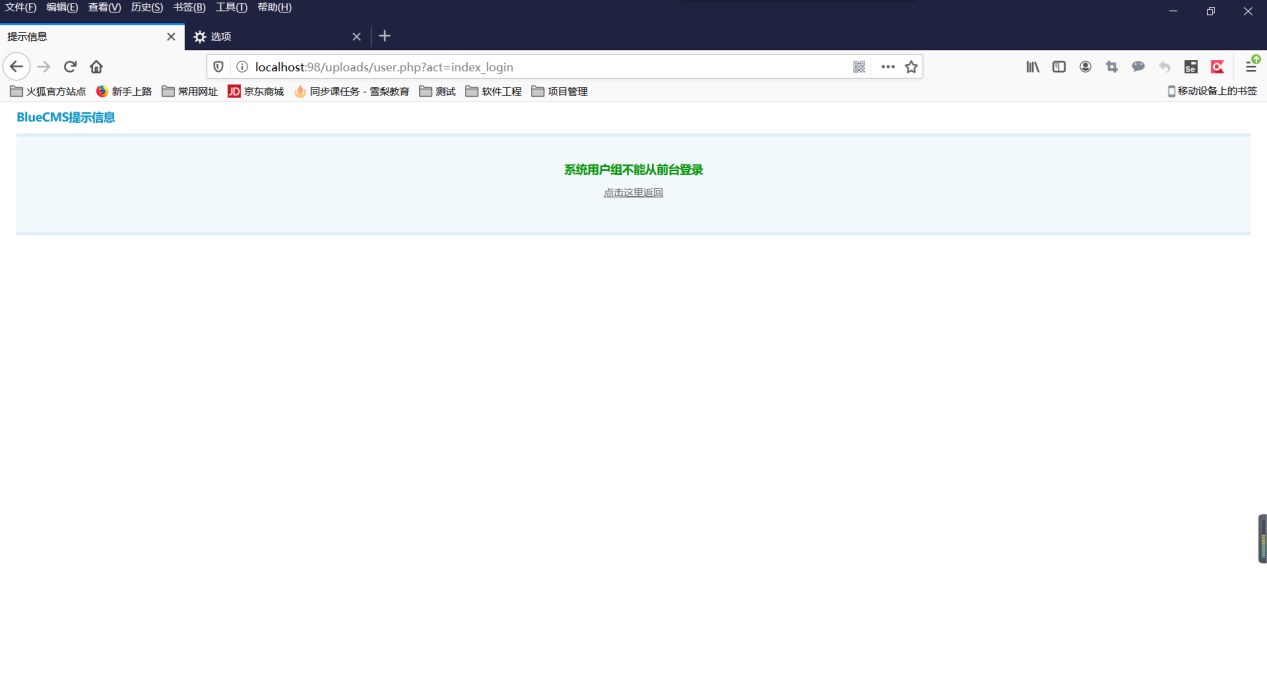
4.1攻击前端登录界面

首先用户名使用admin，密码找到自动化测试常用密码字典（这里我从网上查到了一个三年前更新过的的密码字典，仅供参考：<https://github.com/bit4woo/passmaker）>

4.1.1使用**Sniper（狙击手模式）攻击方式：**

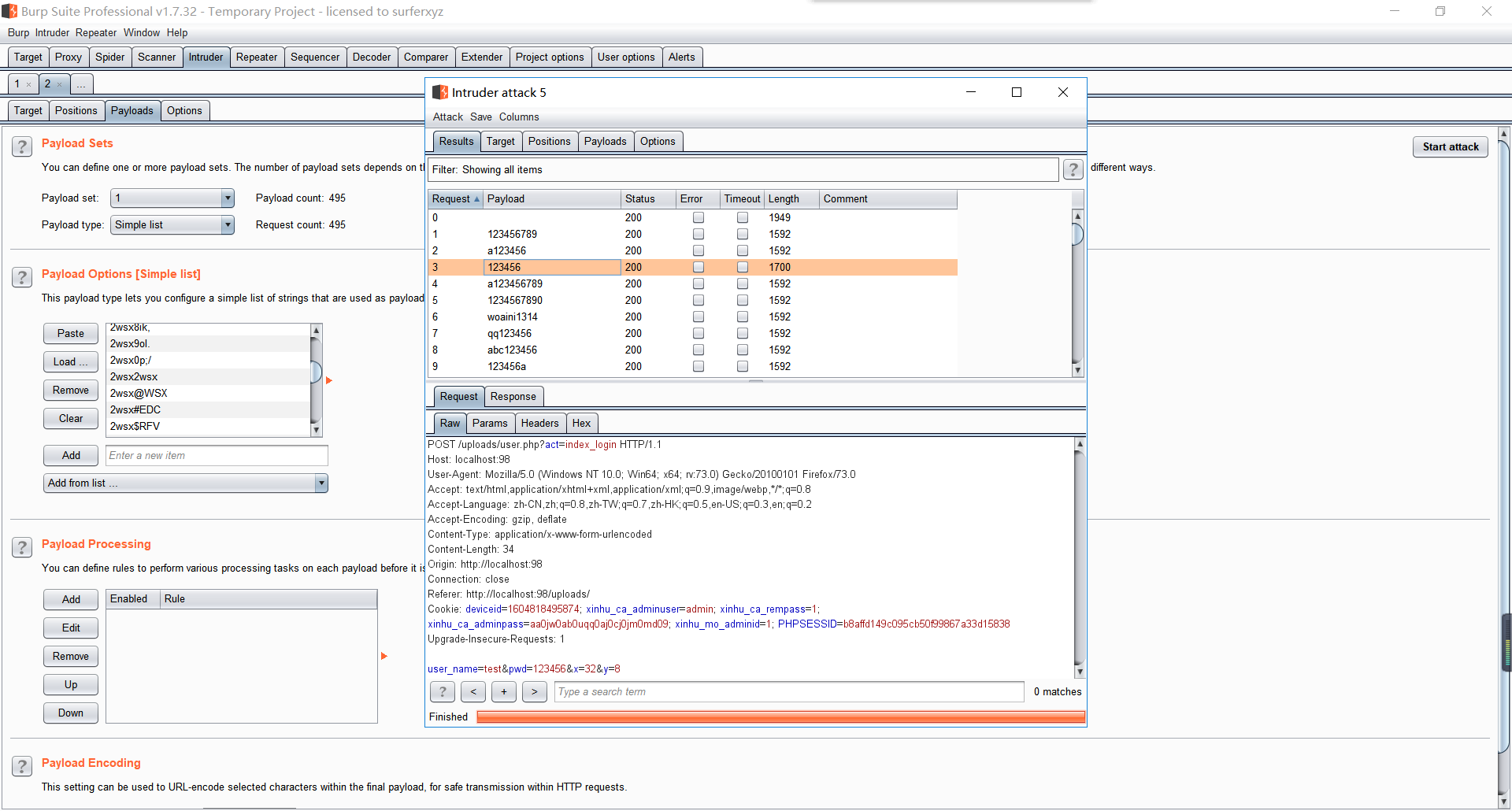


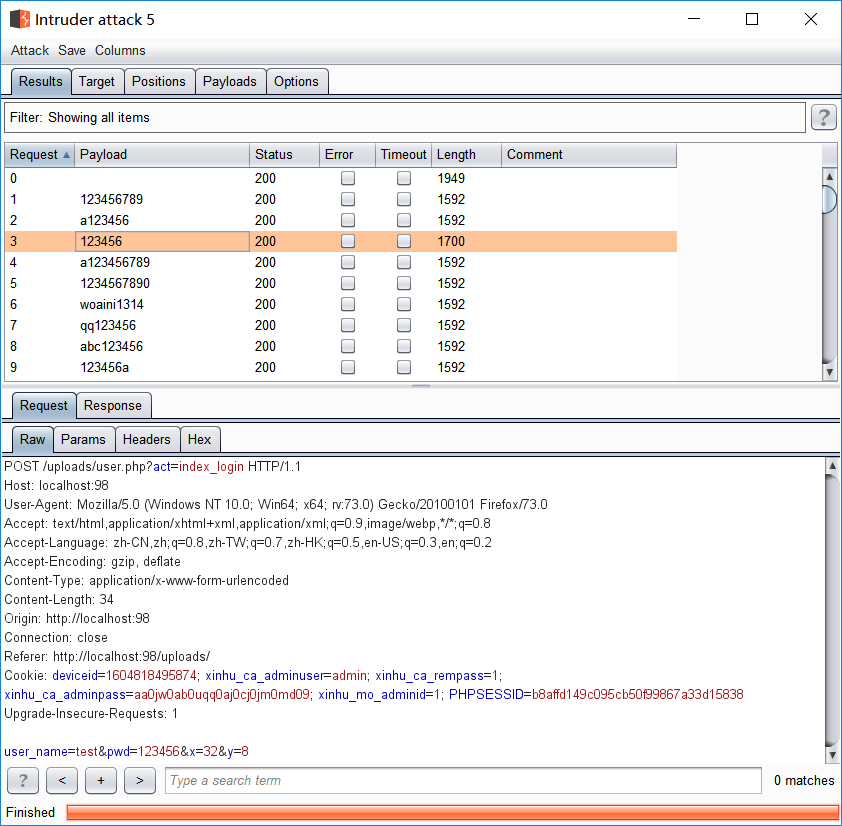
手动登陆可知admin为系统用户组，无法前台登陆



自己注册一个测试账号（用户名：test；密码：123456）

4.1.2重新使用**Sniper（狙击手模式）攻击方式进行**攻击：



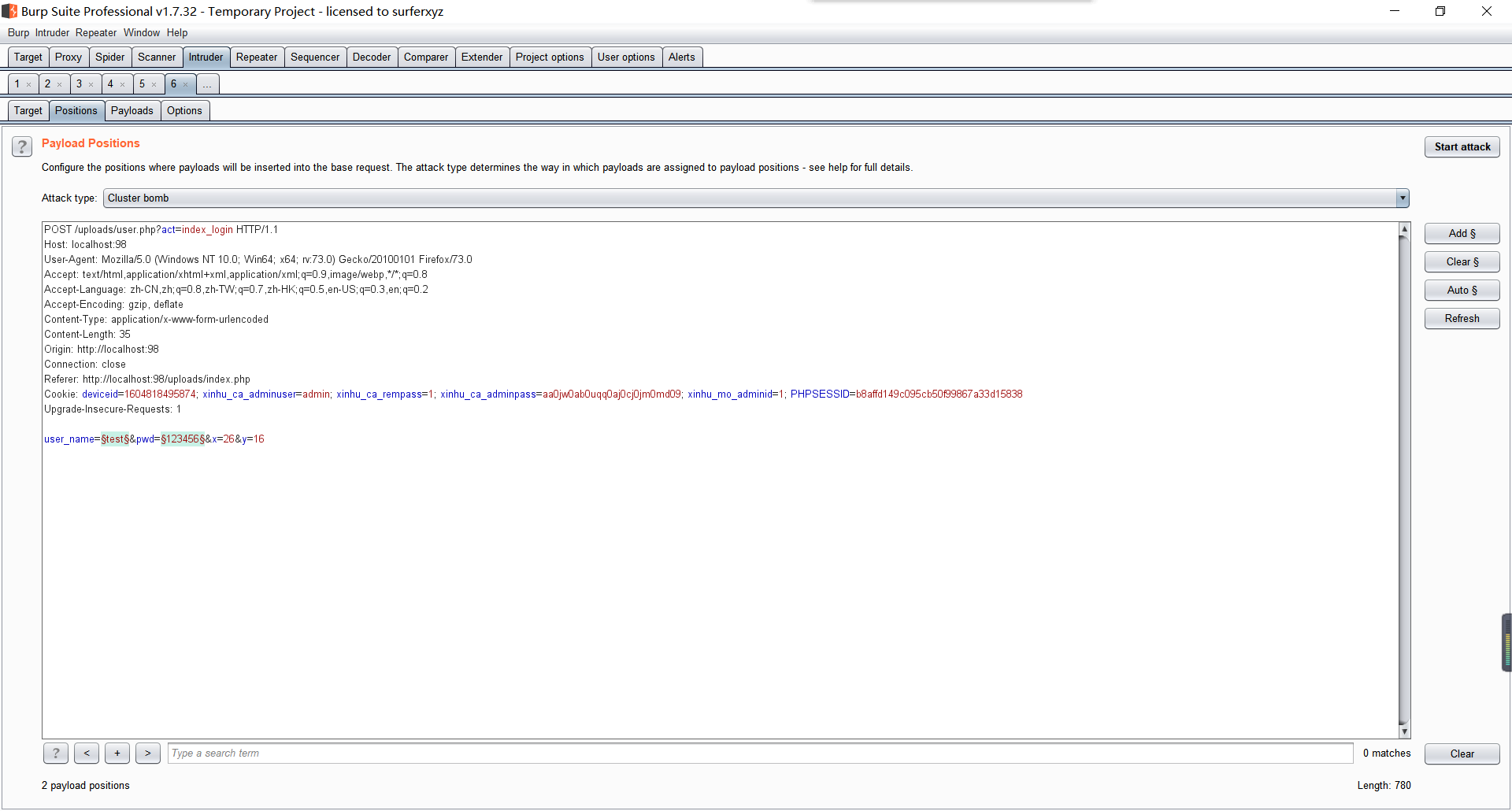


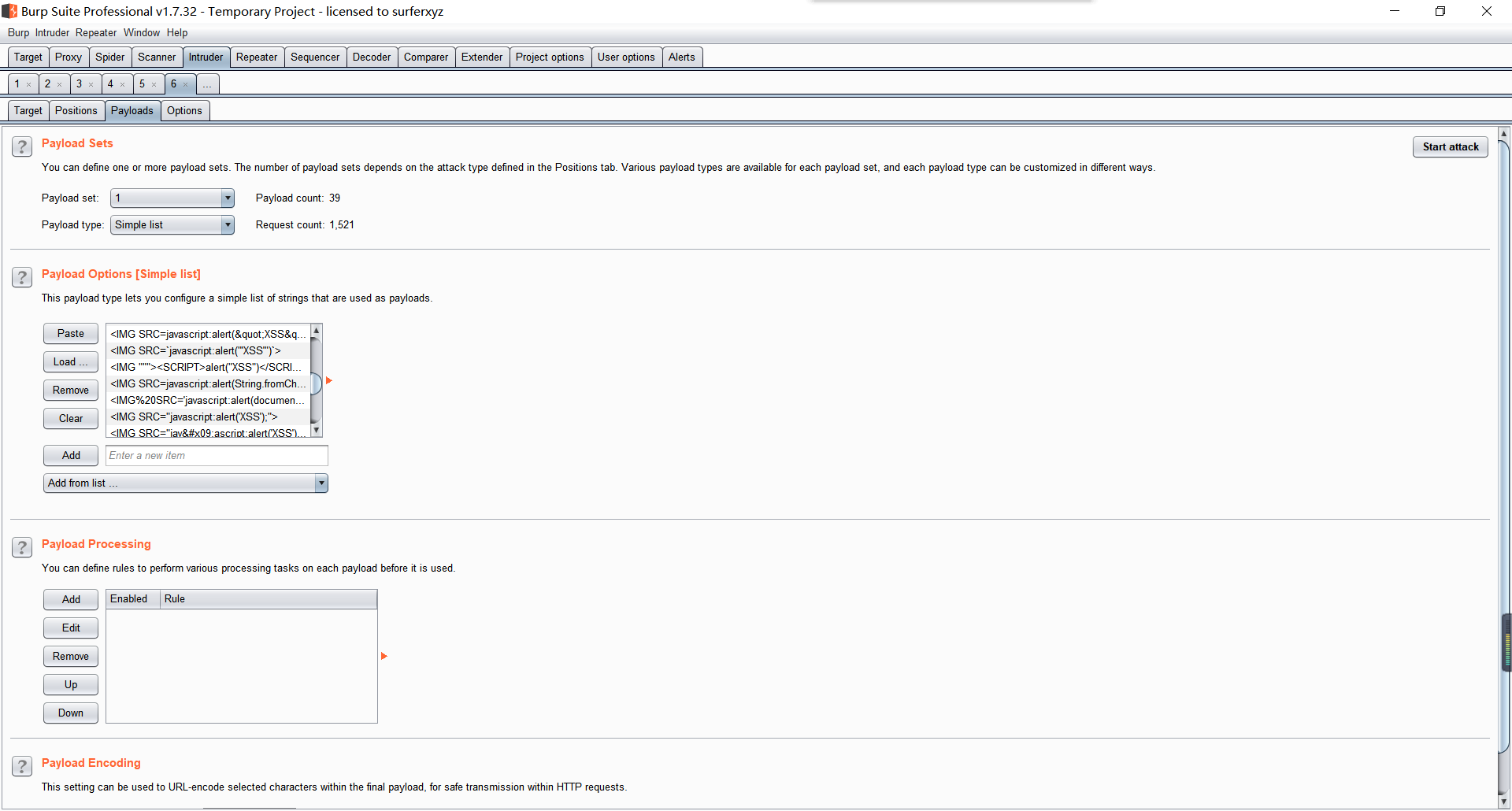
很明显，第三条攻击成功登录，与其他任意攻击结果都不相同

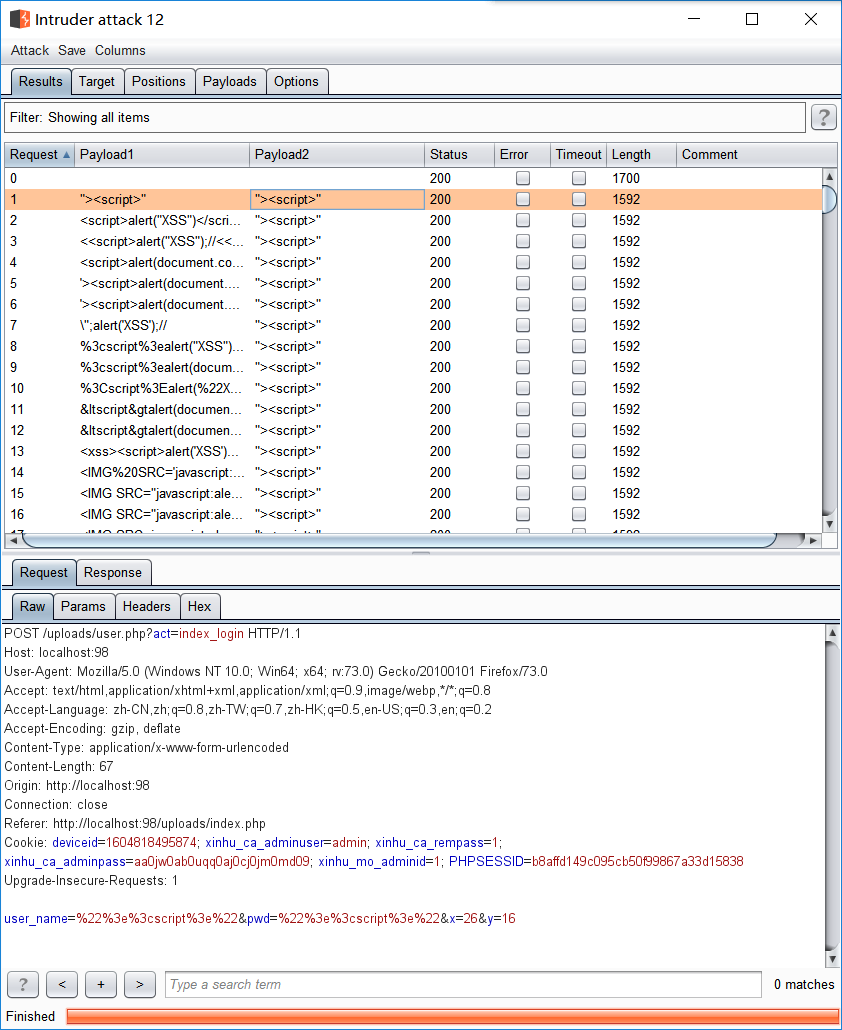
4.1.3前端登录界面XSS漏洞测试：

对后台登陆界面进行XSS攻击，在网上查到XSS攻击相关字典（字典地址：<https://github.com/ssxssx/SaiDict）>

攻击结果：



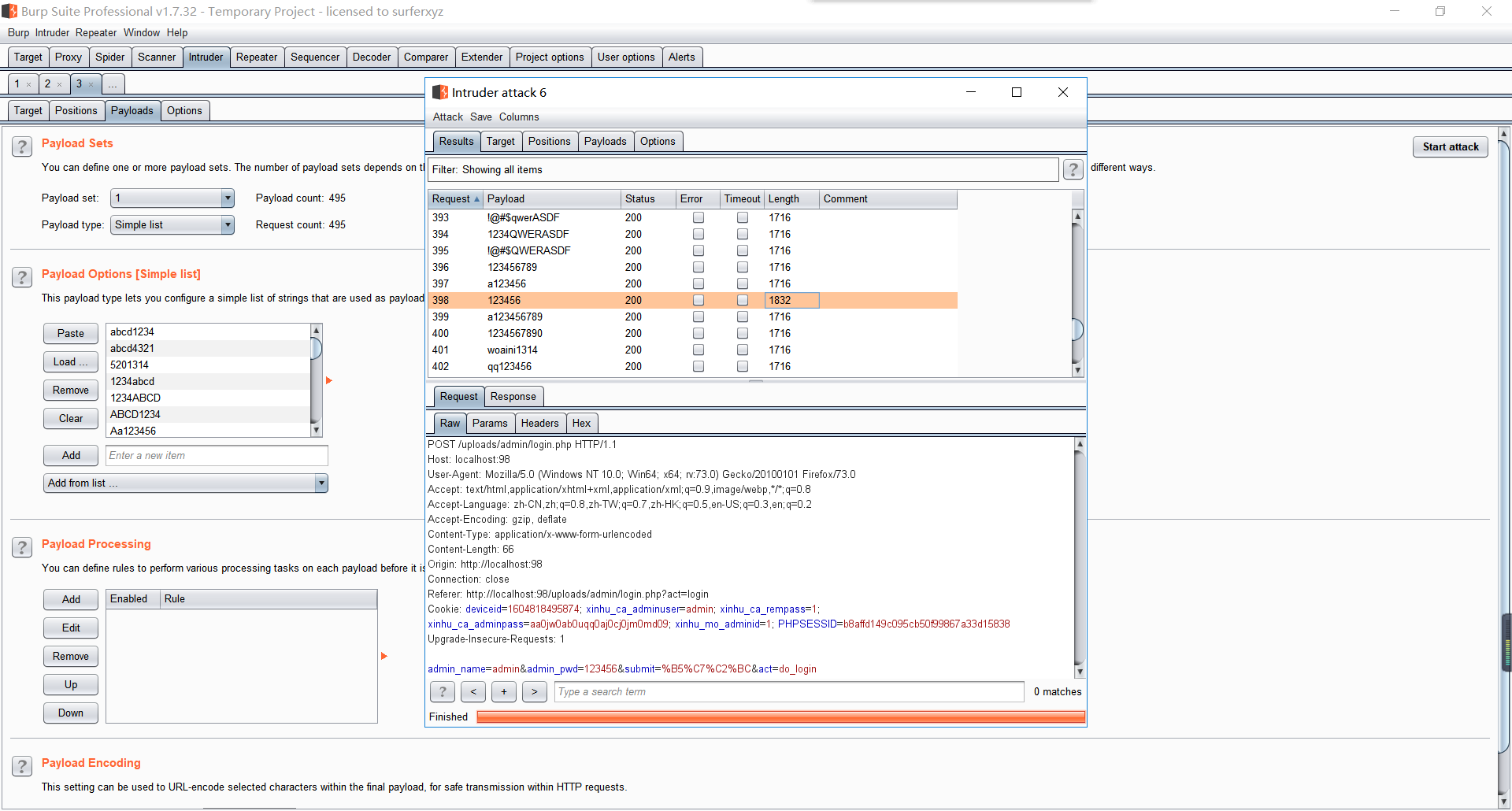


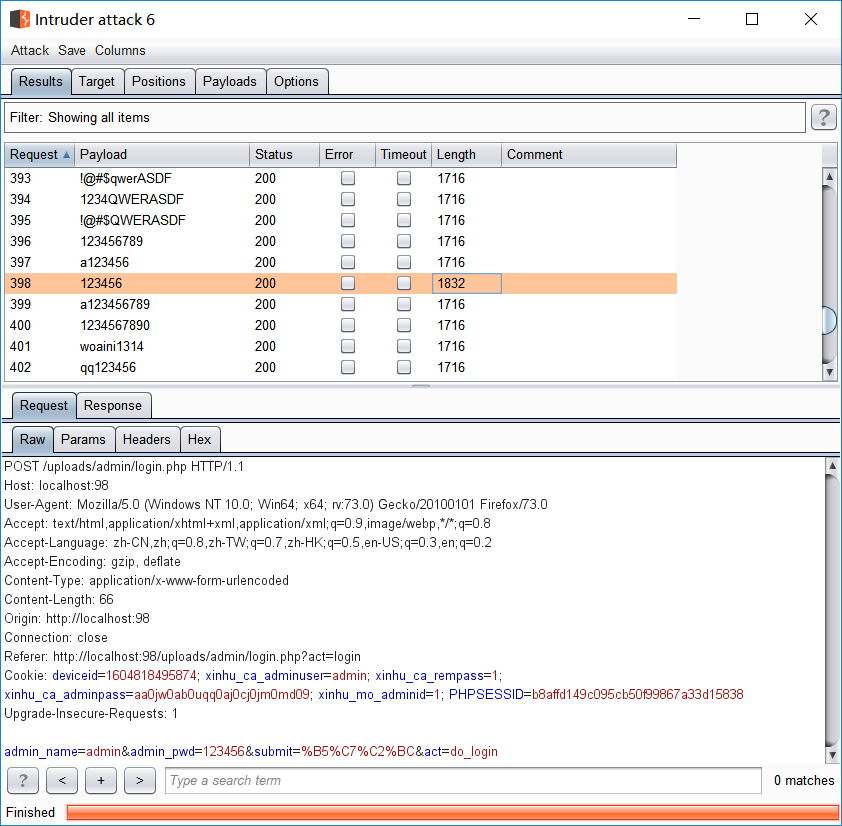


分析结果可知，所得报文长度相同，在登陆界面中，输入用户名、密码标签无XSS漏洞

4.2攻击后台登陆界面

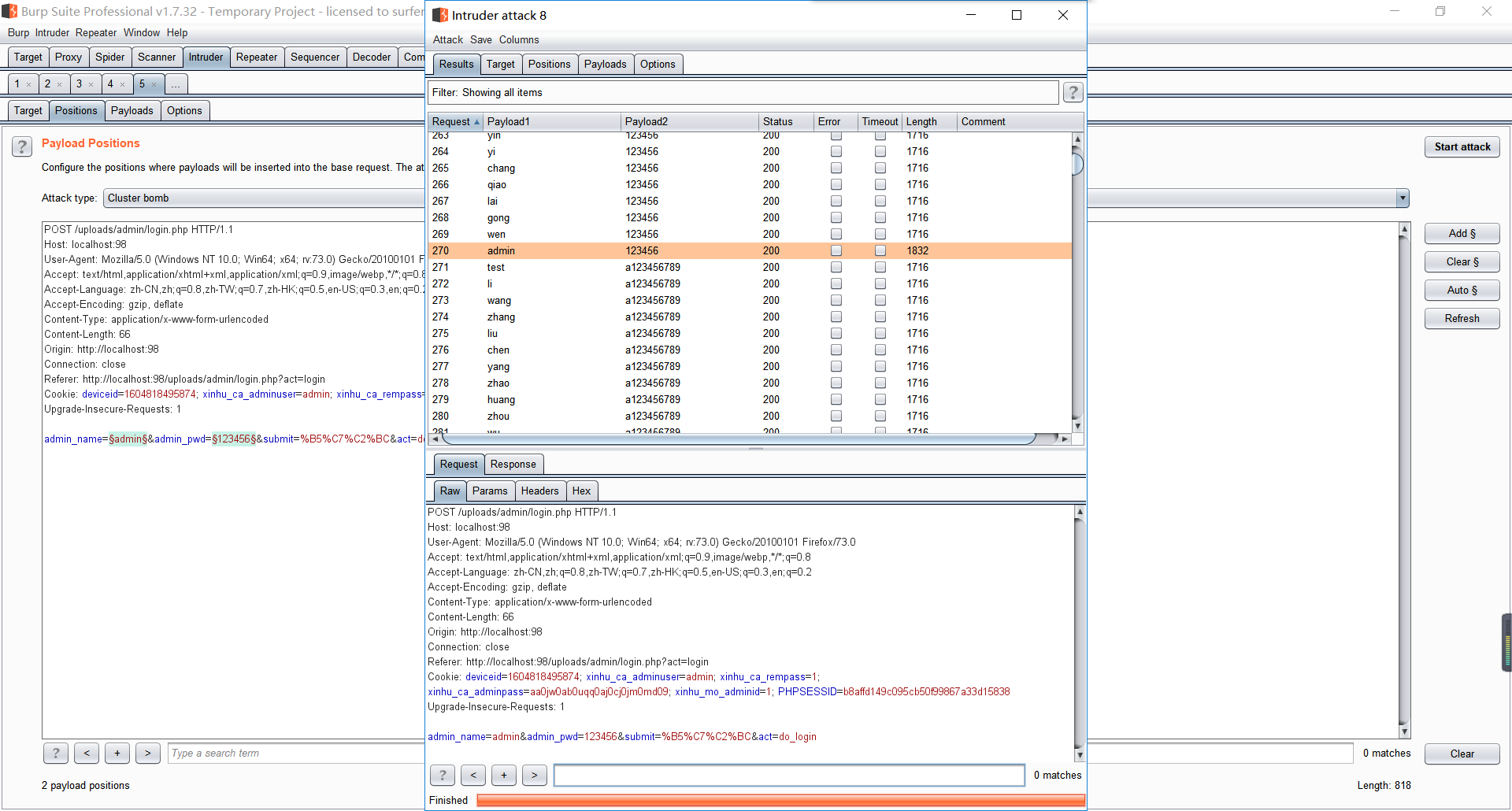
4.2.1使用**Sniper（狙击手模式）攻击方式：**

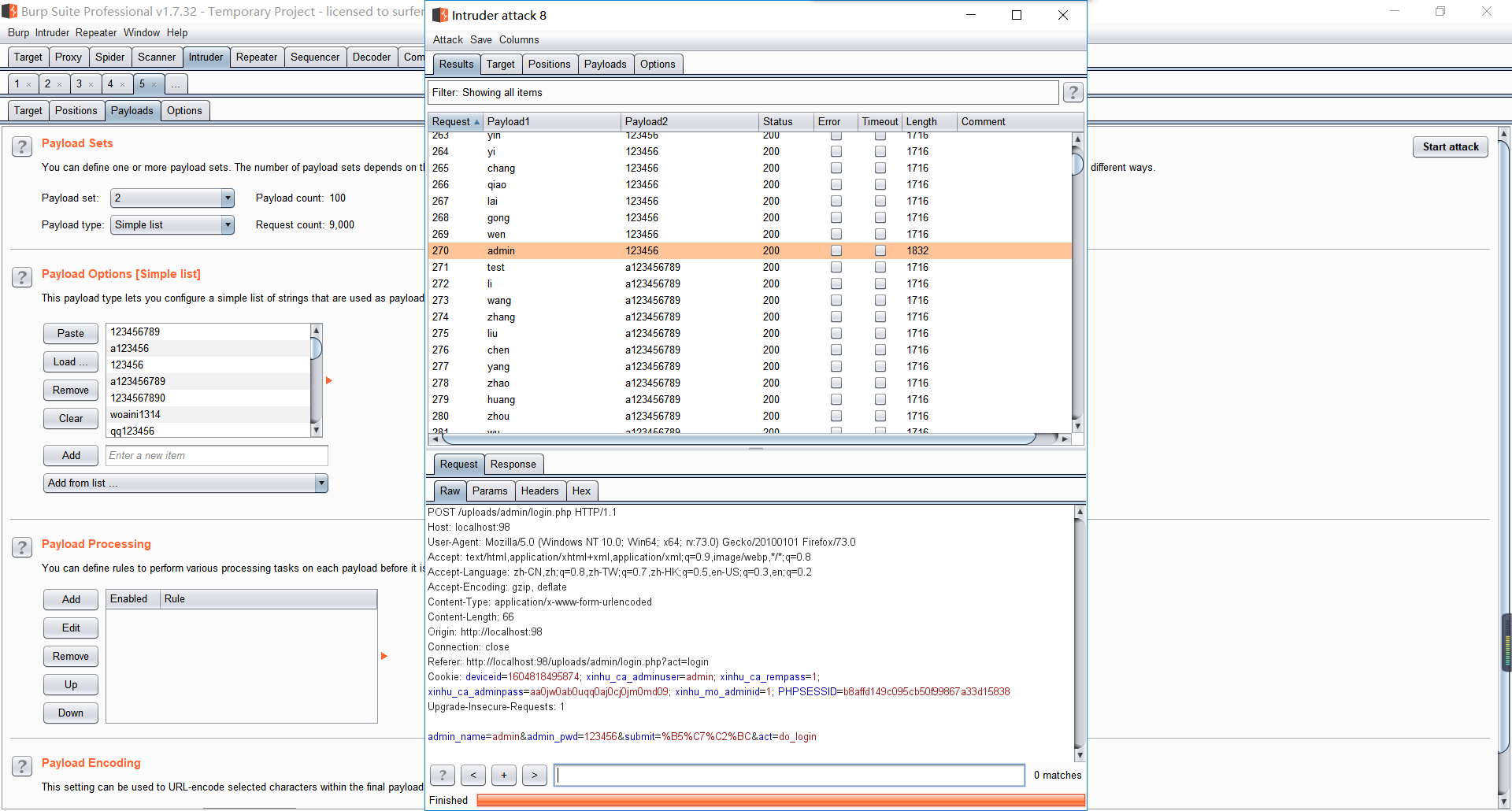


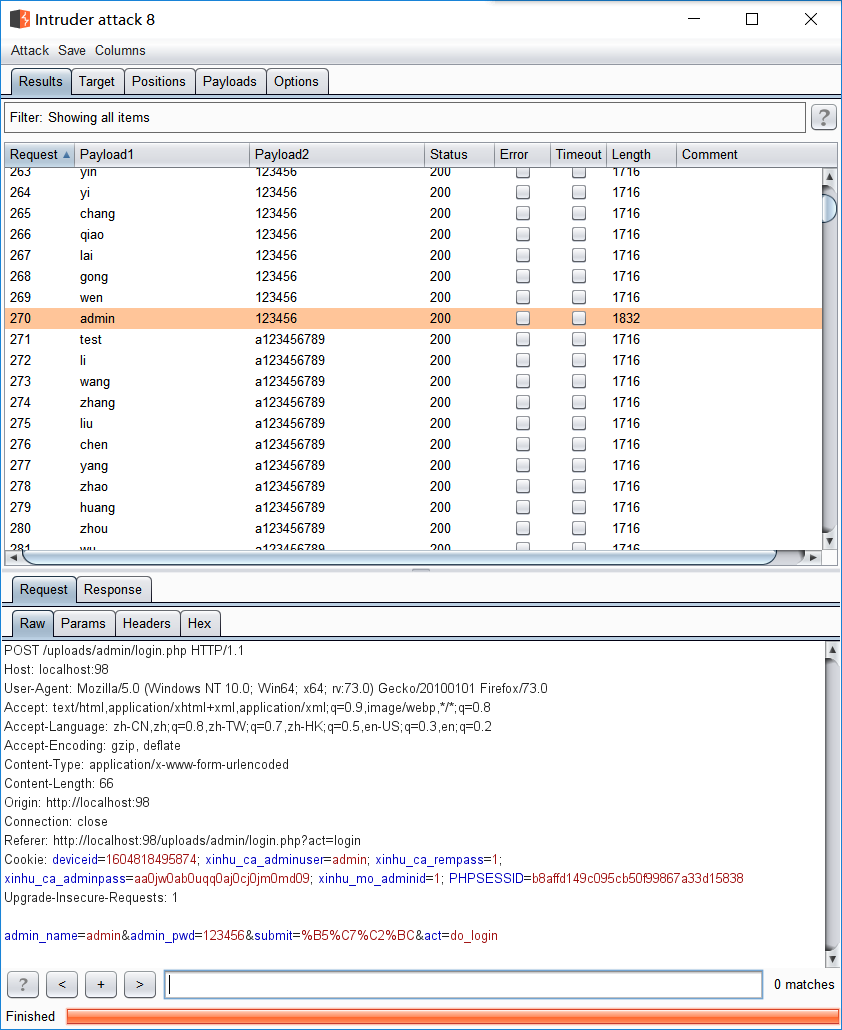


很明显，只有第398条攻击结果与其他不同，而确实是正确的用户名密码

4.2.2使用**Cluster bomb（集束炸弹模式）攻击方式：**





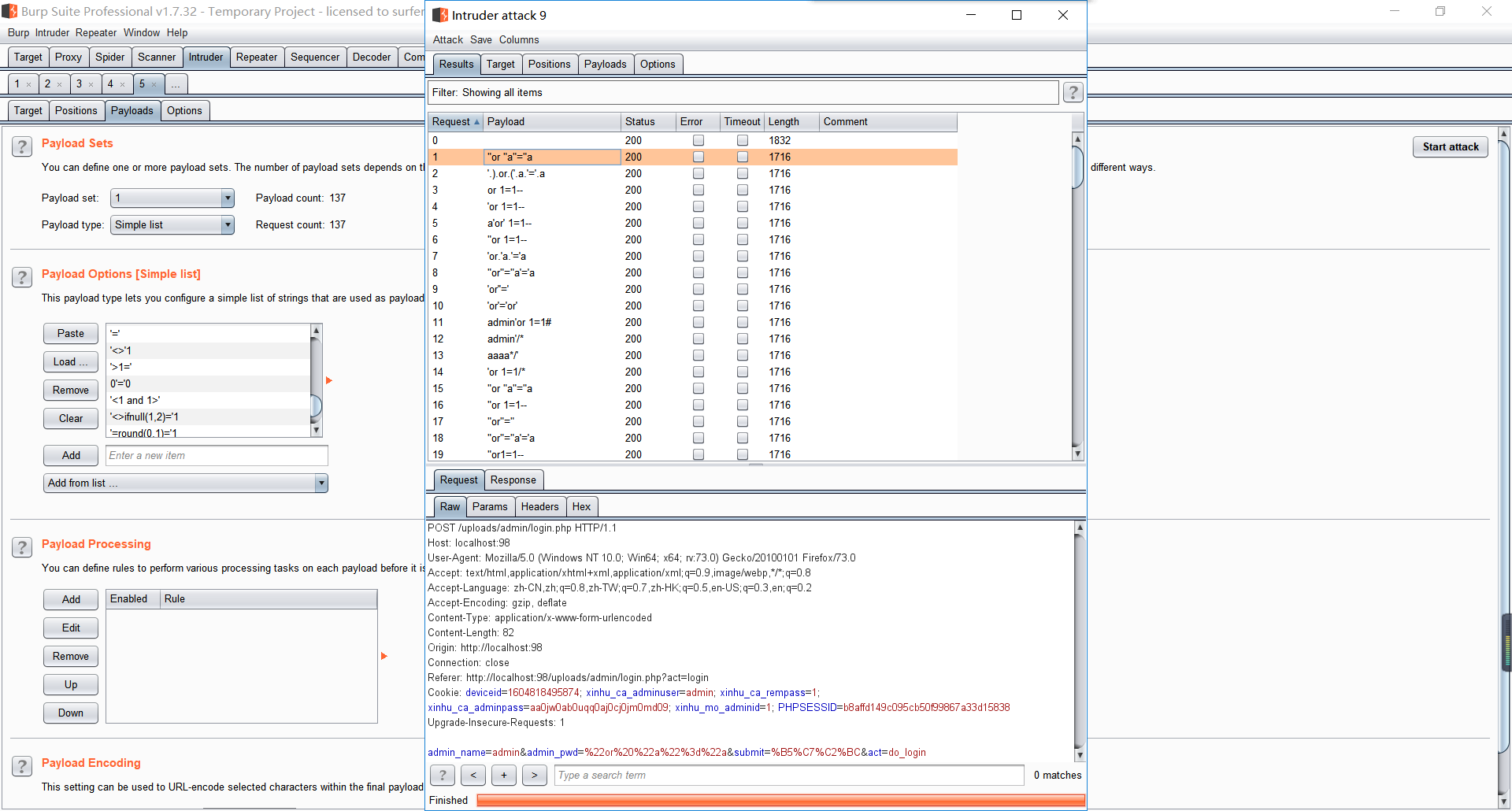


查看全部数据，找到第270条用例与其他结果不同，而用户名：admin密码：123456为正确数据，所以攻击成功

4.2.3使用sql注入方式攻击：

在网上找到sql注入密码字典，copy下来（这是网上找到的一篇SQL注入万能密码字典：<https://blog.csdn.net/weixin_41594045/article/details/102469315）>

攻击结果：



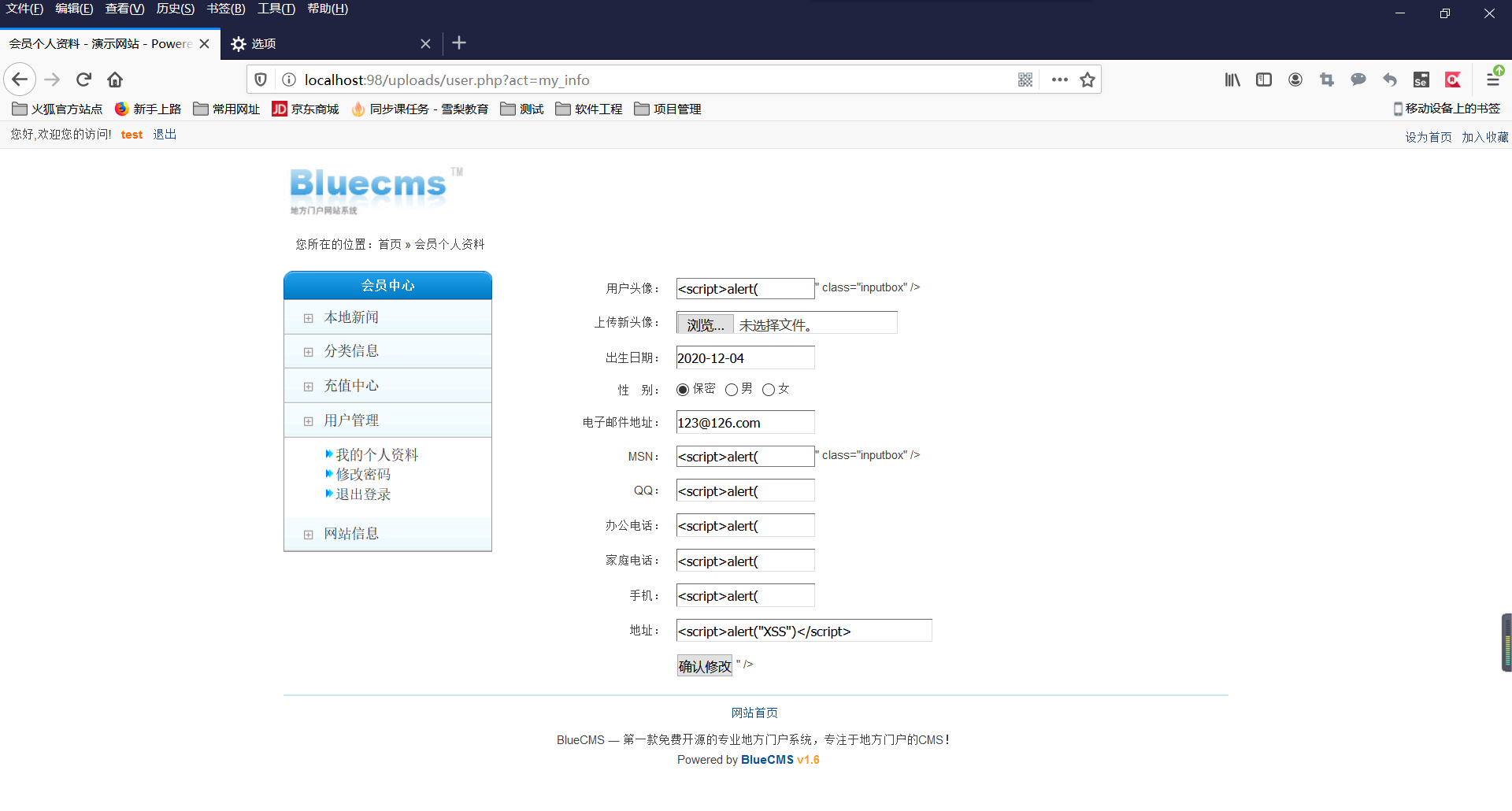
得到结果的报文长度未变化，sql注入漏洞未发现

五、对网站进行手工测试

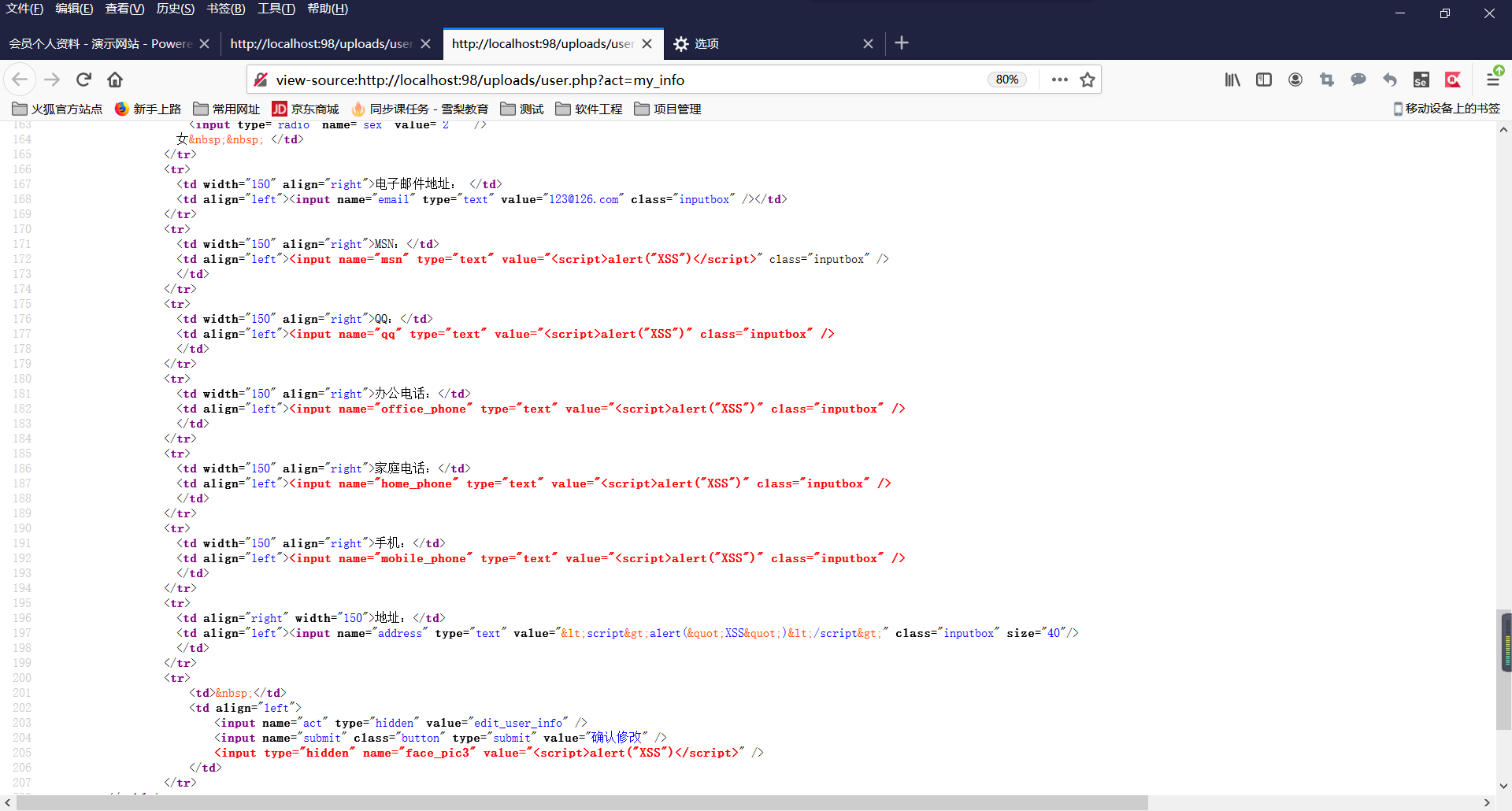
5.1XSS漏洞测试

5.1.1对前端页面的XSS漏洞测试

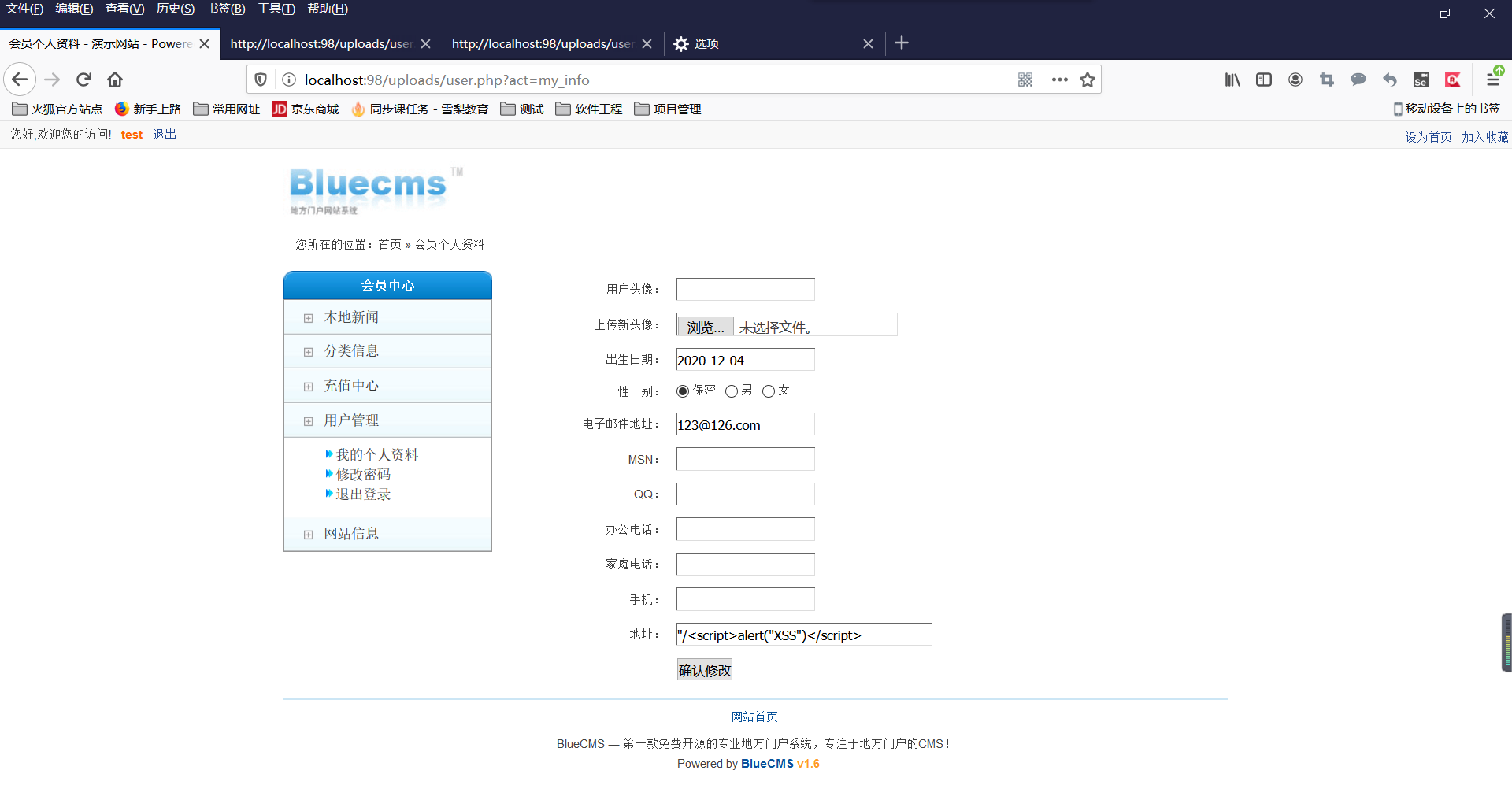
在前端页面中找到input标签，这里以【用户管理】中的【我的个人资料】为例：



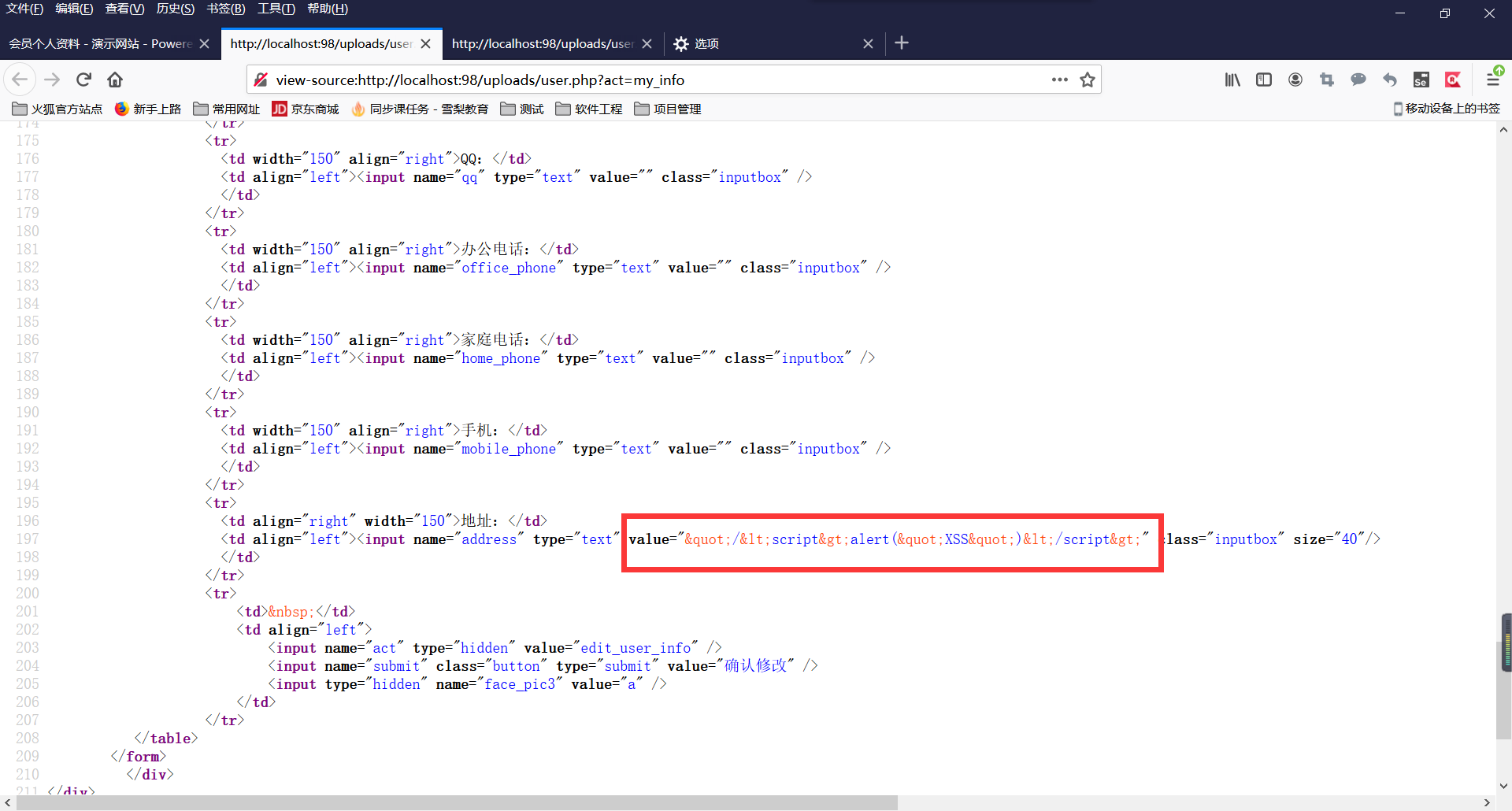
在每一个input标签内，输入XSS攻击语句，点击【确认修改】后，未弹出alert提示，查看网页源代码：



根据此情况，重新设计XSS攻击语句：



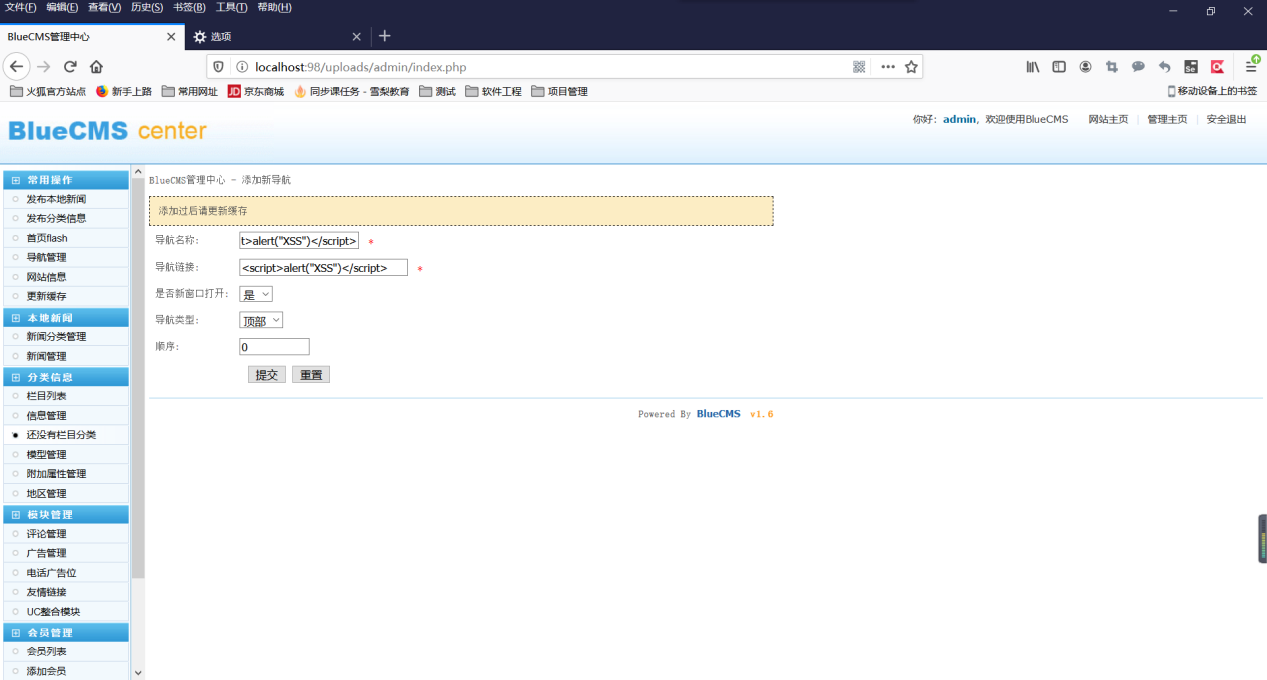
点击【确认修改】后，仍未弹出alert提示，再次查看网页源代码：



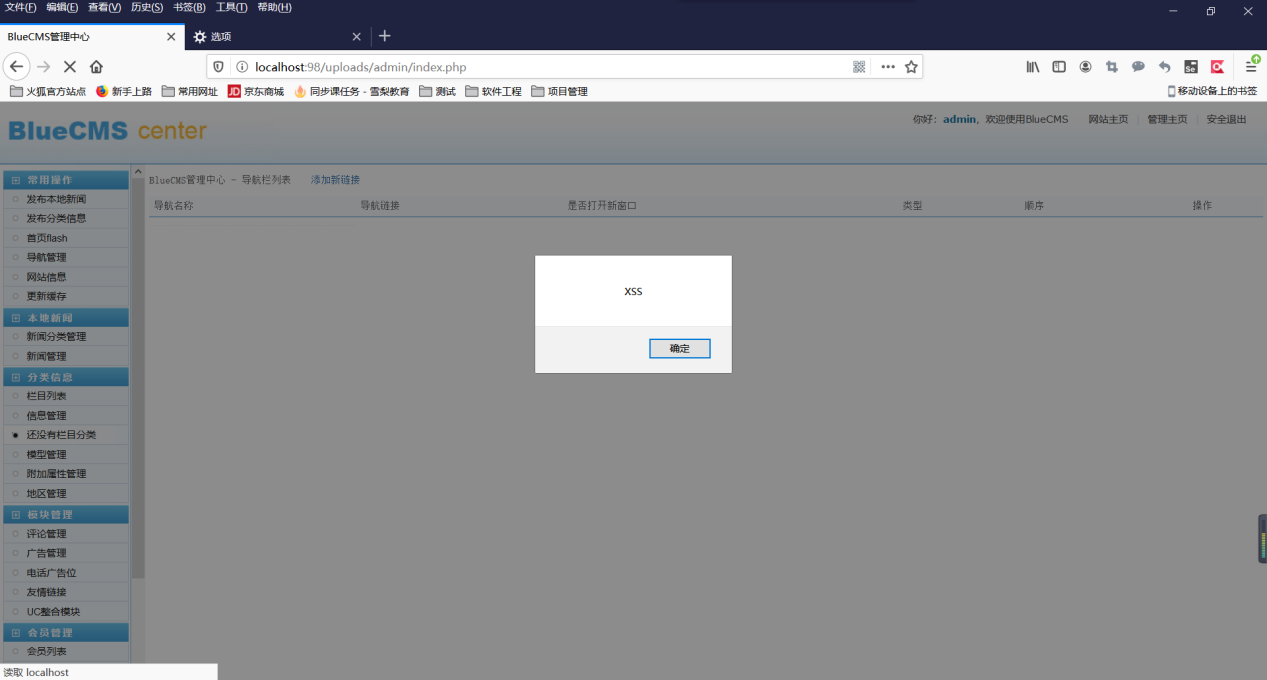
对value内容进行了符号的转码，由以上情况可得出结论：前端页面不存在XSS漏洞。

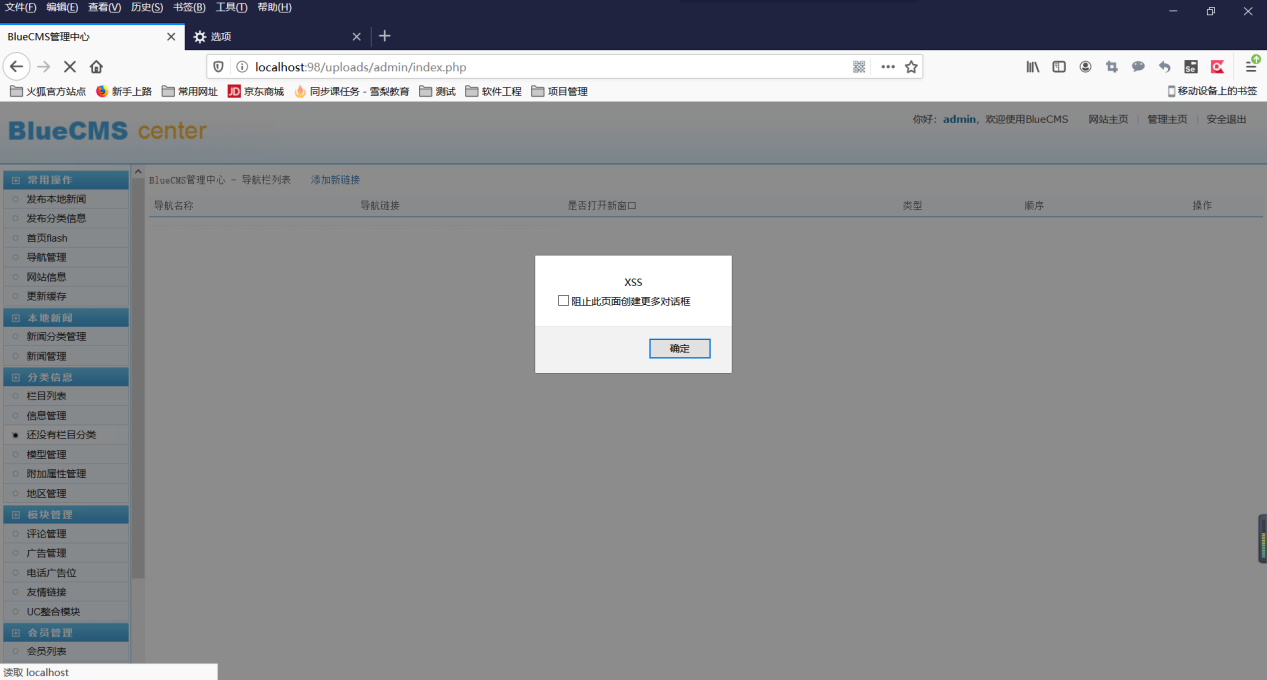
5.1.2对后台页面的XSS漏洞测试

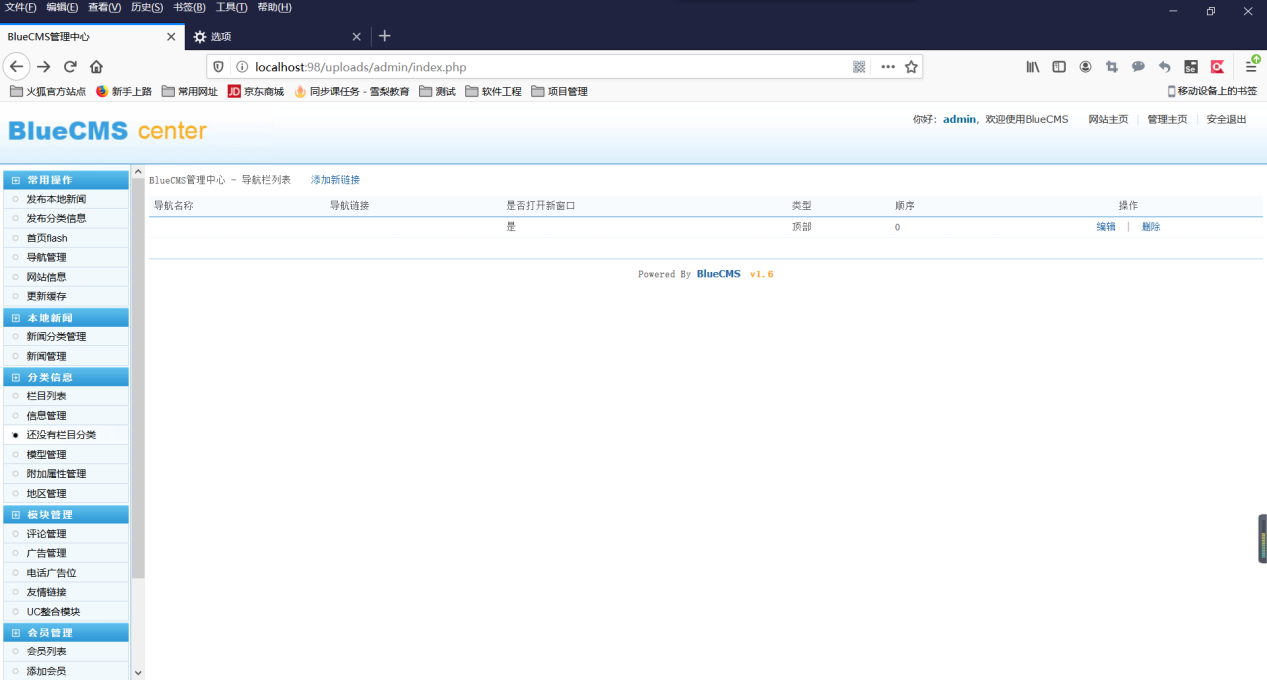
在后台界面中，找到input标签，这里以【常用操作】【导航管理】【添加新链接】中的input标签为例：

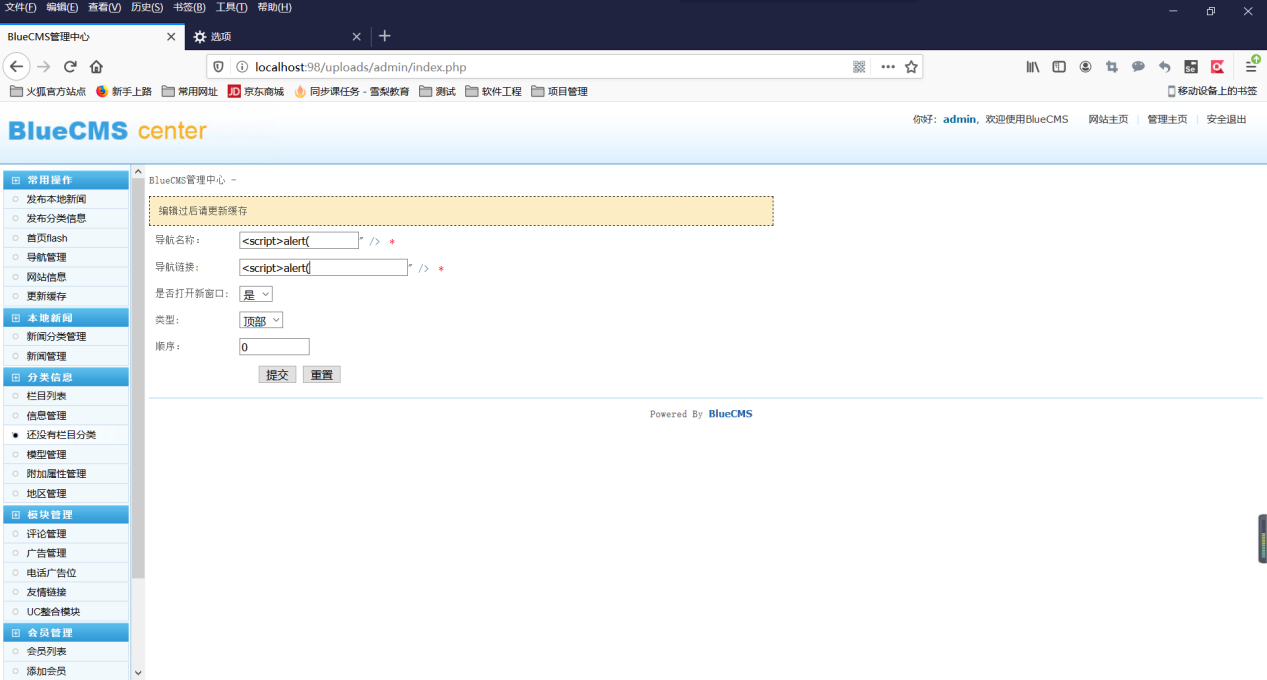


输入XSS攻击代码之后，点击提交









会弹出alert窗口，攻击成功，证明后台页面中存在XSS漏洞。

5.2SQL注入漏洞测试

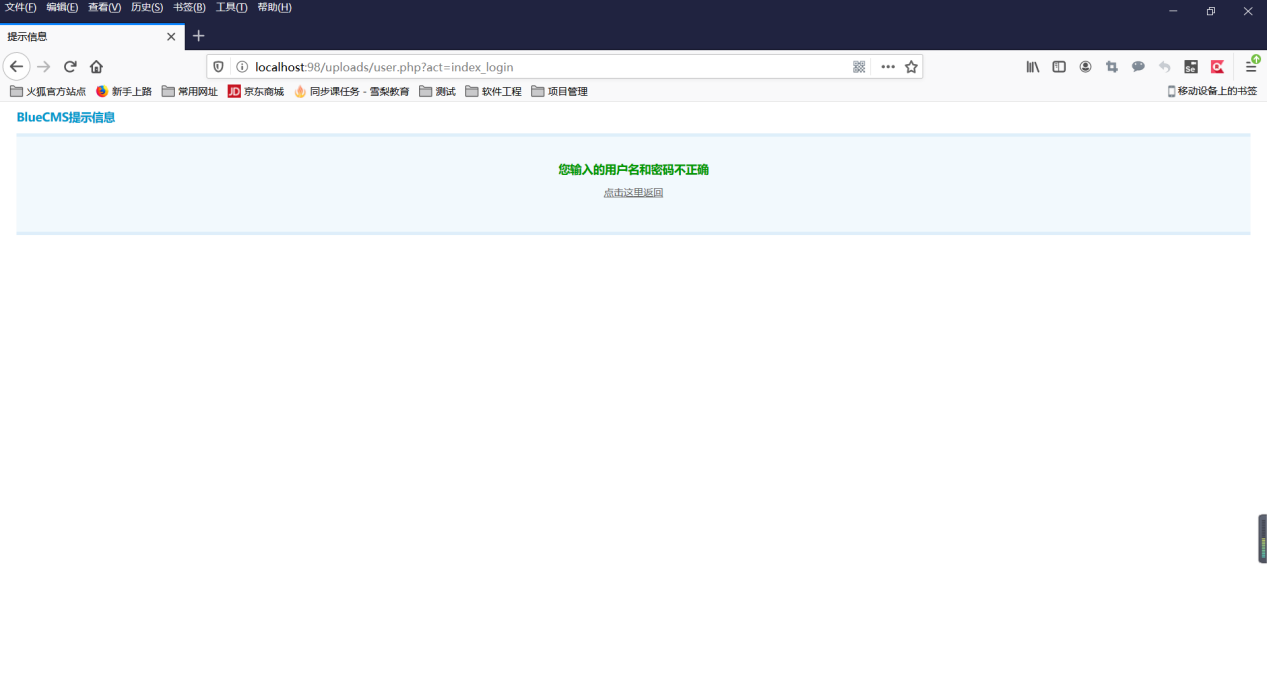
5.2.1对前端页面的SQL注入漏洞测试

在前端登录页面中找到input标签，这里以登录为例：

用户名密码输入常见的一些sql语句



点击登陆，提示：用户名密码错误

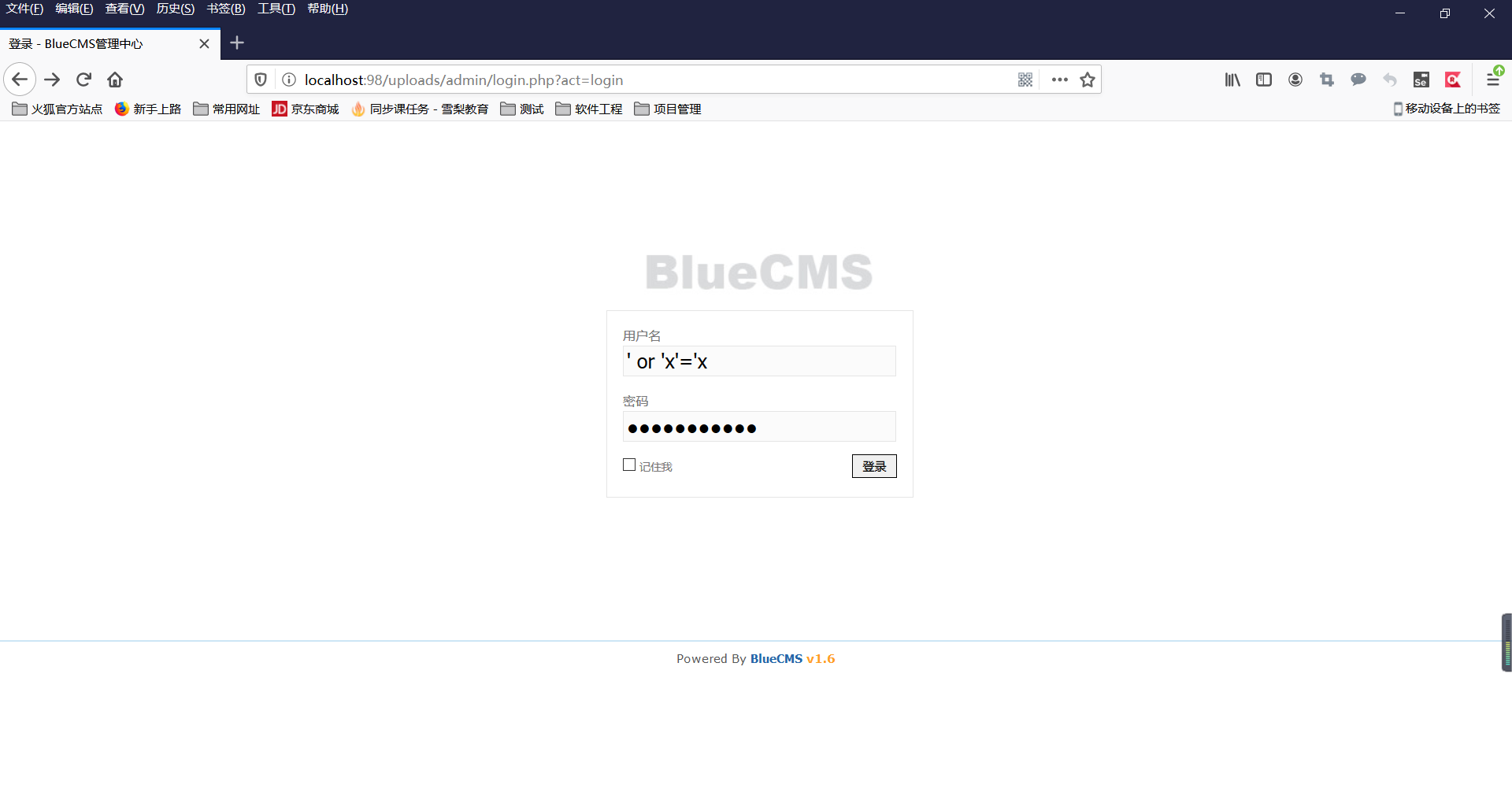


可以得出：前端未发现sql注入漏洞。

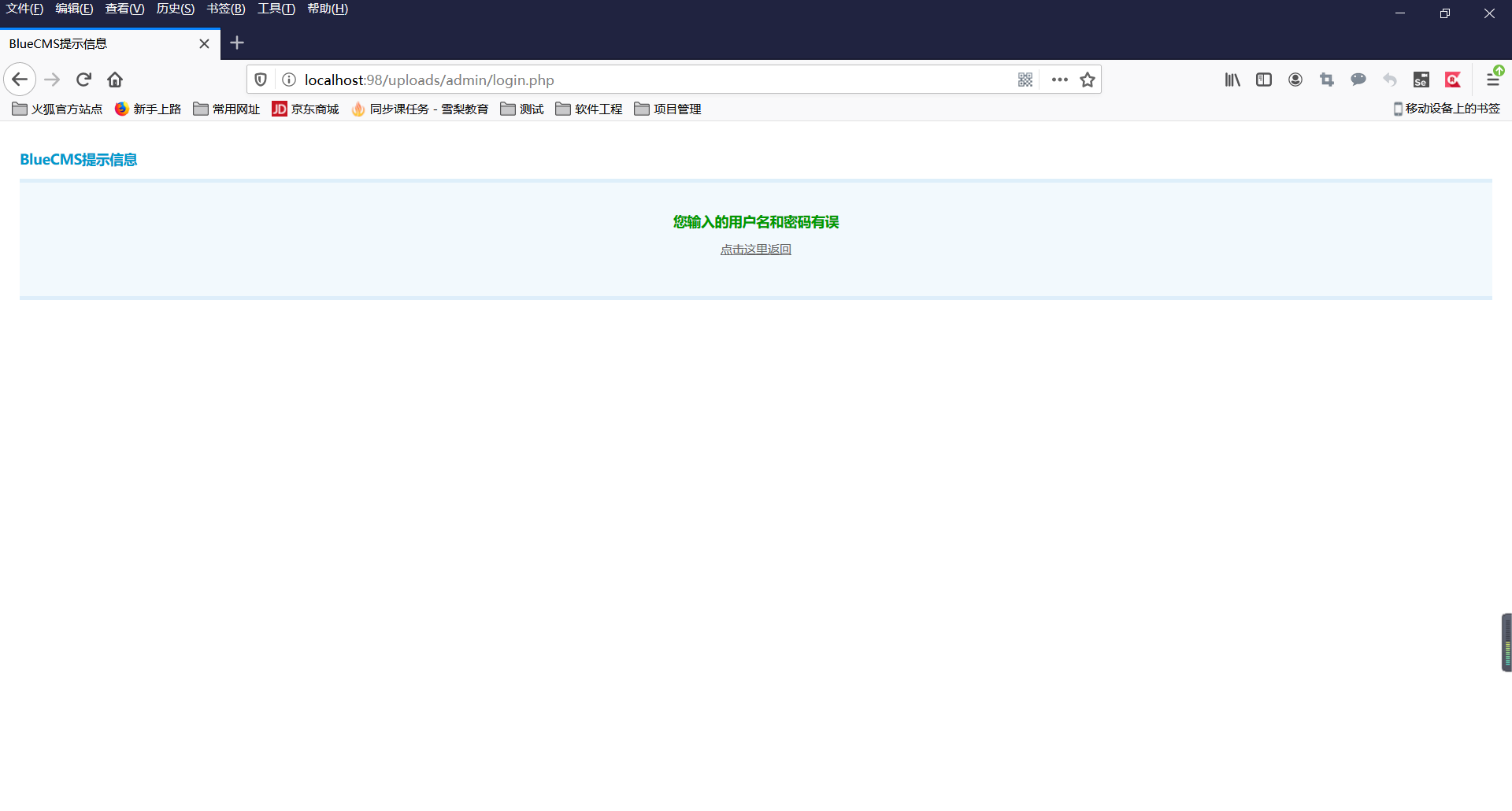
5.2.2对后台页面的SQL注入漏洞测试

在后台页面中找到input标签，这里以登录为例：

用户名密码输入常见的一些sql语句



点击登陆，提示：用户名密码错误



可以得出：后台未发现sql注入漏洞。