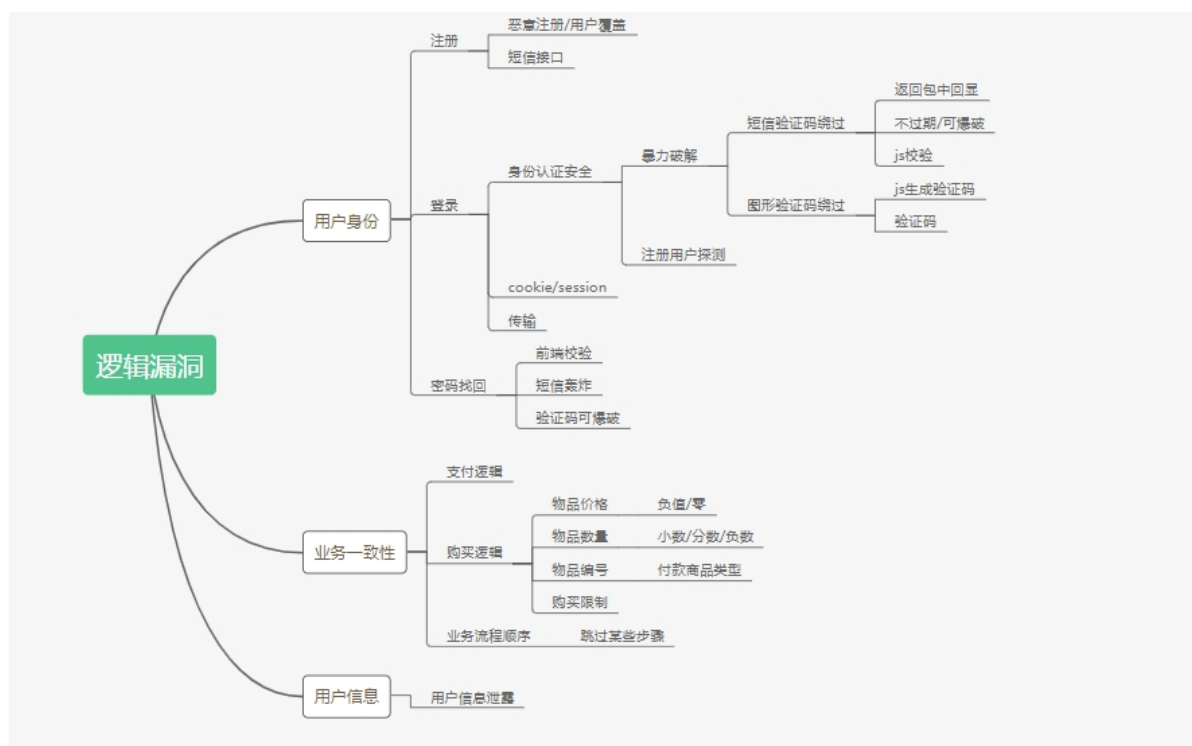
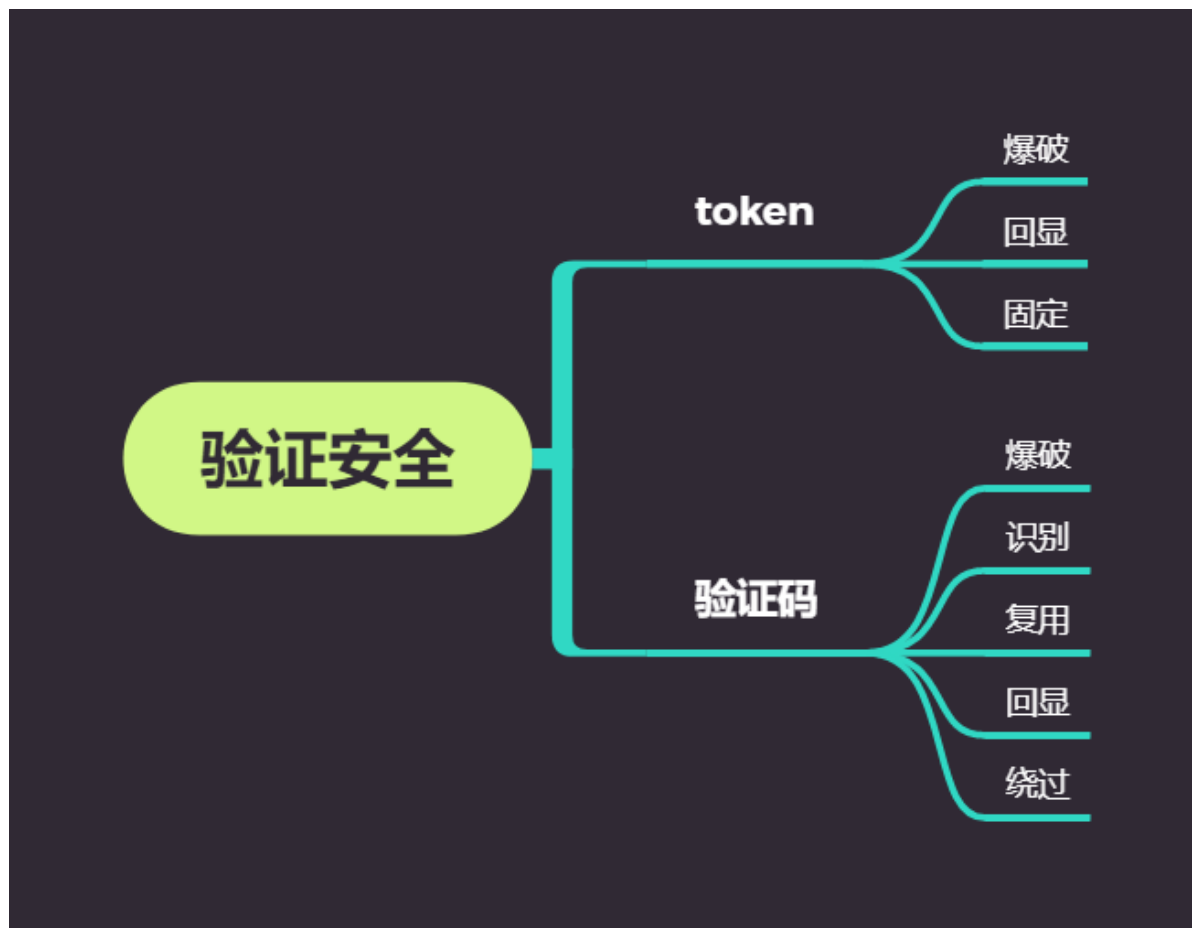


# Day36 WEB漏洞-逻辑越权之验证码与Token 及接口



## 36.1 验证码安全

### 36.1.1 分类

图片，手机或邮箱，语音，视频，操作等

### 36.1.2 原理

验证生成或验证过程中的逻辑问题

### 36.1.3 危害

账户权限泄漏，短信轰炸，遍历，任意用户操作等

### 36.1.4 漏洞

客户端回显(已讲)，验证码复用，验证码爆破(已讲)，绕过等

1. 验证码爆破：没有次数限制，验证码有效期内不变，爆破绕过验证码的方法：验证码的复用，只需要发送第一次验证码，然后通过数据包爆破密码
2. 验证码识别：用工具识别验证码
3. 复用：用上一次的验证码来绕过下一次的验证
4. 回显：验证码在前端数据包显示
5. 绕过：逻辑上的绕过，直接跳过验证码

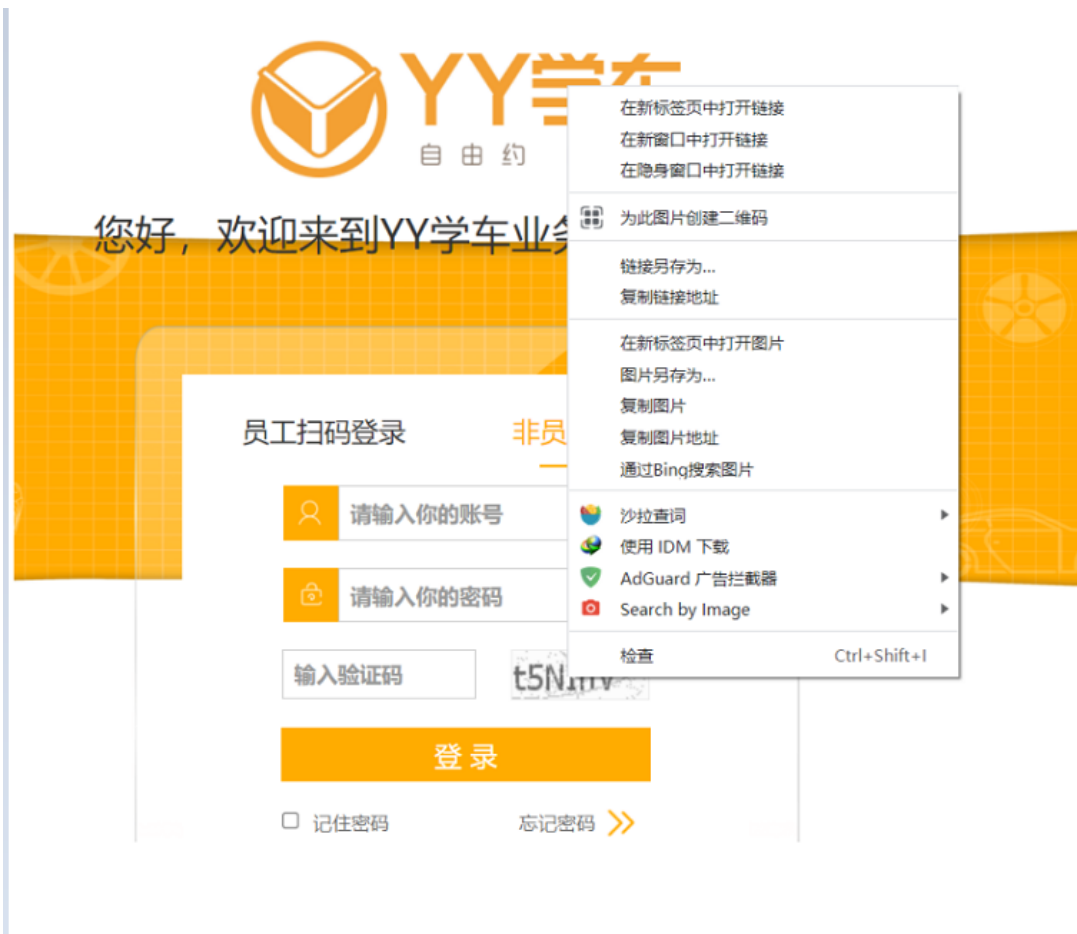
### 36.1.5 爆破防范措施



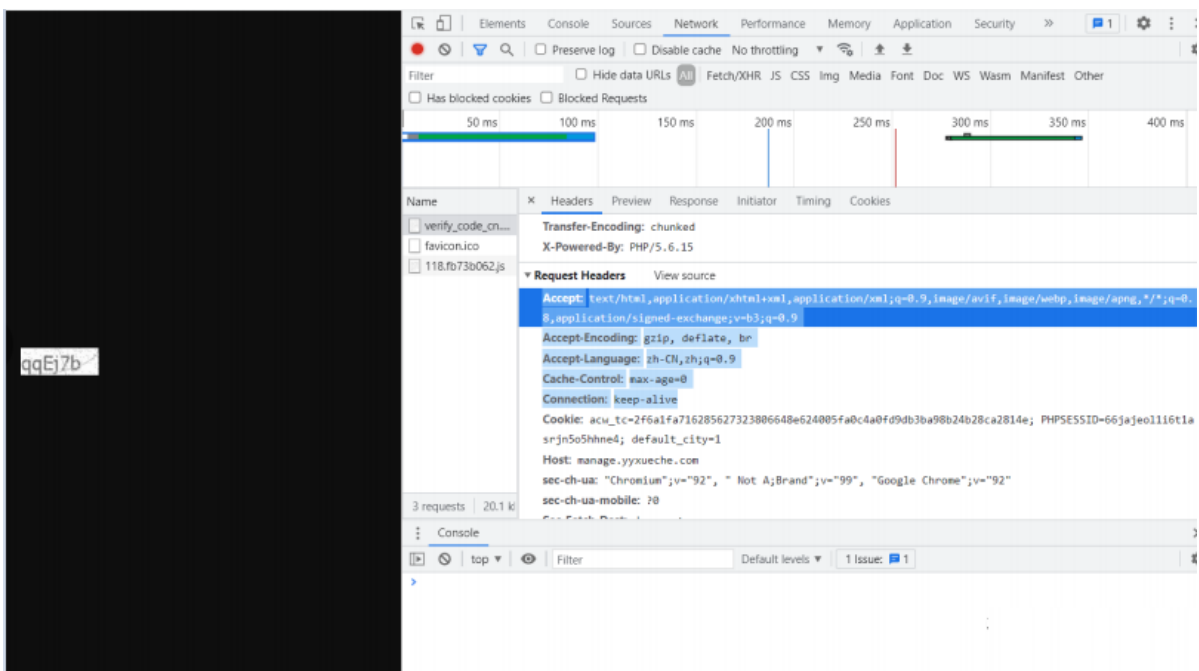
1. 如验证码输入3次就重新获取
2. 每输入一下变换一次
3. 使用特殊字符集
4. 特殊的验证码如图形，滑动，判断图形等

## 36.1.6 验证码案例分析

1 <https://manage.yyxueche.com//panel/login.php>



复制图片地址:



验证码地址, 为验证码图片的链接地址:

● 图片型

验证码地址:

https://nanage.yyxueche.com/verify\_code\_cn.php?103311

其他请求头部:

d-exchange;v=b3;q=0.9  
Accept-Encoding: gzip, deflate, br  
Accept-Language: zh-CN,zh;q=0.9  
Cache-Control: max-age=0  
Connection: keep-alive

● 自带识别引擎

识别模式

- ☒ 单个文本统一块    ☐ 单一的文本行    ☐ 一个单词    ☐ 无OSD全自动页分割  
☐ 垂直对齐文本的统一块    ☐ 可变大小文本中的一列    ☐ 无OSD或OCR的自动页面分割  
☐ 仅OSD的定位及检测    ☐ OSD模式自动页面分割    ☐ 圈内的一个单词

识别范围

- ☐ 不限定  
☐ 清晰的数字  
☒ 限定为以下字符:

0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

○ 第三方识别引擎

- ☒ 亦思验证码识别引擎    ☐ 次世代验证码识别引擎

识别库:

○ 第三方识别引擎

- ☐ 亦思验证码识别引擎    ☒ 次世代验证码识别引擎

识别库:

加载...

识别测试:

验证码图片:



获取到的验证码为:

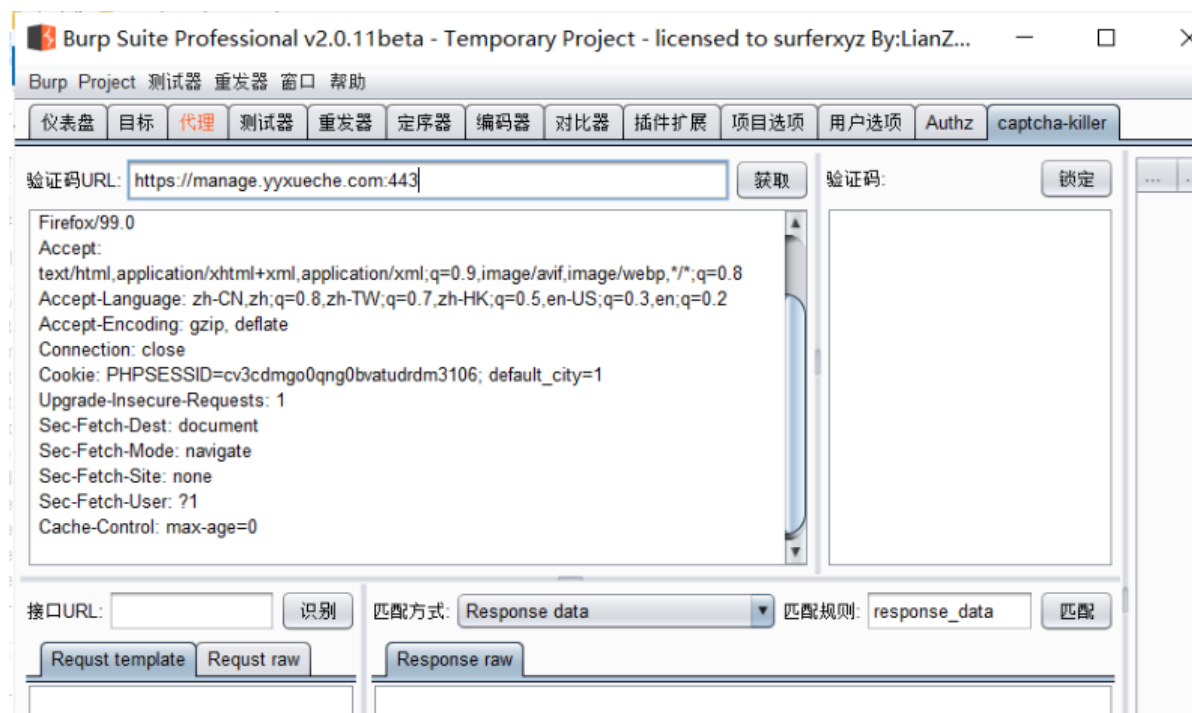
wPAeWf



- 1 总结:
- 2 右键选择验证码“复制图片地址”，访问图片地址
- 3 抓取图片地址的数据包：右键“审查元素”“Network”，刷新一下
- 4 找到请求数据包“Request Headers”复制GET, Host和Cookie之间的部分
- 5 打开工具，选择“图片型验证码识别”
- 6 将复制内容添加到“其他请求头部”，在填写“验证码地址”
- 7 根据需要调整参数、引擎等，点击“识别测试”
- 8 缺点:
- 9 识别精度有限
- 10 缺少接口，只能加载能够下载的认识库
- 11 只能识别简单的验证码
- 12 只能使用自带的发包器，不能结合到burpsuite

## 36.1.7 captcha-killer插件

浏览器访问验证数据包，抓包，右键点击captch-killer:



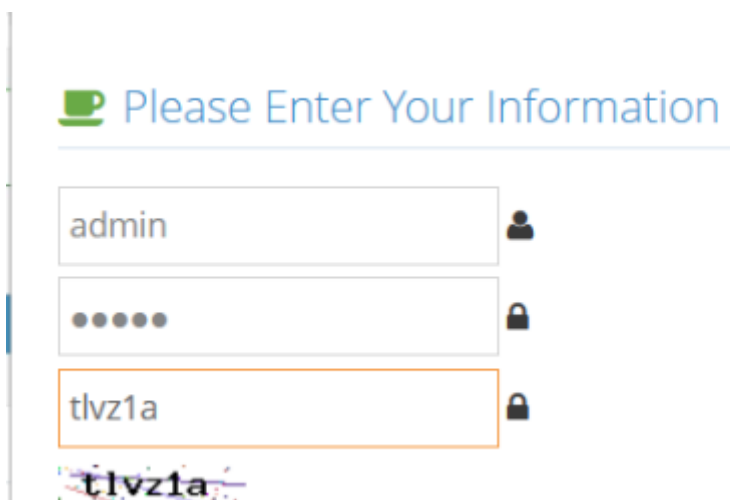


- 1 使用方法:
- 2 使用burpsuite抓包，右键发送到“captcha-killer”“Send to captcha panel”
- 3 接口URL填识别平台的接口地址
- 4 开始识别
- 5 爆破时使用该插件：“Attack type”选择“Pitchfork”，“Payload type”选择“Extension-generated”，“Extension payload generator”选择“captcha-killer”。不要多线程

### 36.1.8 验证码绕过本地及远程验证-本地及实例

#### 1.on server

随便输入账号密码，输入正确的验证码，提交，提示用户名或密码不存在：



抓包正确的验证码，查看返回，验证码正确：

Burp Suite Professional v2.0.11beta - Temporary Project - licensed to surferxyz By:LianZ...

Burp Project 测试器 重发器 窗口 帮助

仪表盘 目标 代理 测试器 重发器 定序器 编码器 对比器 插件扩展 项目选项 用户选项 Authz captcha-killer

1 x 2 x 3 x 4 x 5 x 6 x 7 x 8 x 9 x 10 x 11 x 12 x 13 x 14 x 15 x ...

发送 取消 < >

目标: http://192.168.2.56

请求

Raw 参数 头 Hex

```
POST /pikachu/vul/burteforce/bf_server.php
HTTP/1.1
Host: 192.168.2.56
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:99.0) Gecko/20100101 Firefox/99.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 59
Origin: http://192.168.2.56
Connection: close
Referer: http://192.168.2.56/pikachu/vul/burteforce/bf_server.php
Cookie: PHPSESSID=q3fb7lhv4cahk7jgmeadq05hr7
Upgrade-Insecure-Requests: 1

username=admin&password=11111&vcode=im2n65&submit=Login
```

响应

Raw 头 Hex HTML Render

```
src="../../inc/showvcode.php"
onclick="this.src='../../inc/showvcode.php'?+new Date().getTime();" />
</label>

<div class="space"></div>

<div class="clearfix">
  <label><input
class="submit" name="submit" type="submit"
value="Login" /></label>
</div>

</form>
<p> username or password is not
exists</p>

</div><!-- /.widget-main -->

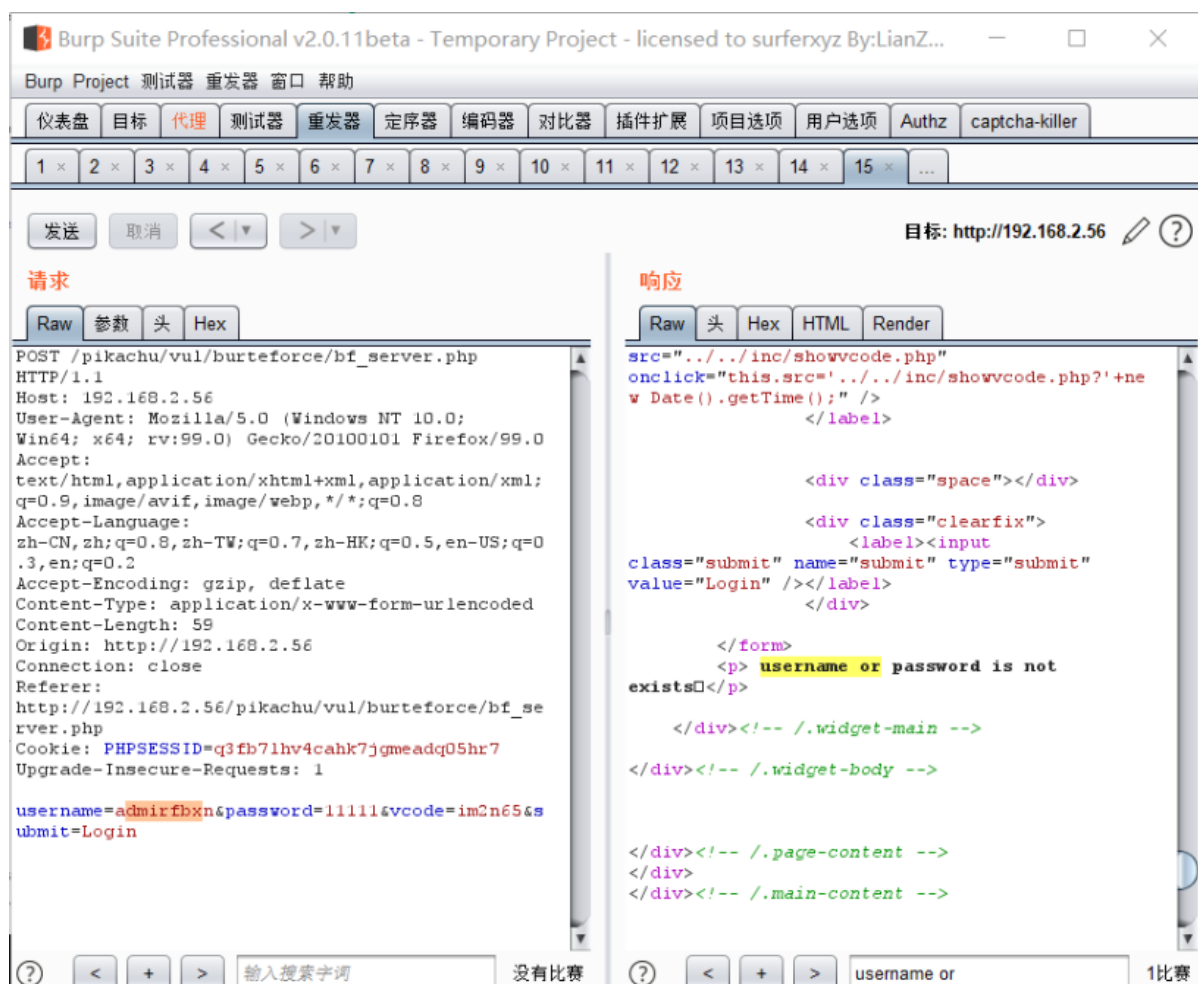
</div><!-- /.widget-body -->

</div><!-- /.page-content -->
</div>
</div><!-- /.main-content -->
```

输入搜索字词 没有比赛

username or 1比赛

修改用户名发送，还是提示用户和密码不存在，没有提示验证码错误，可以复用：

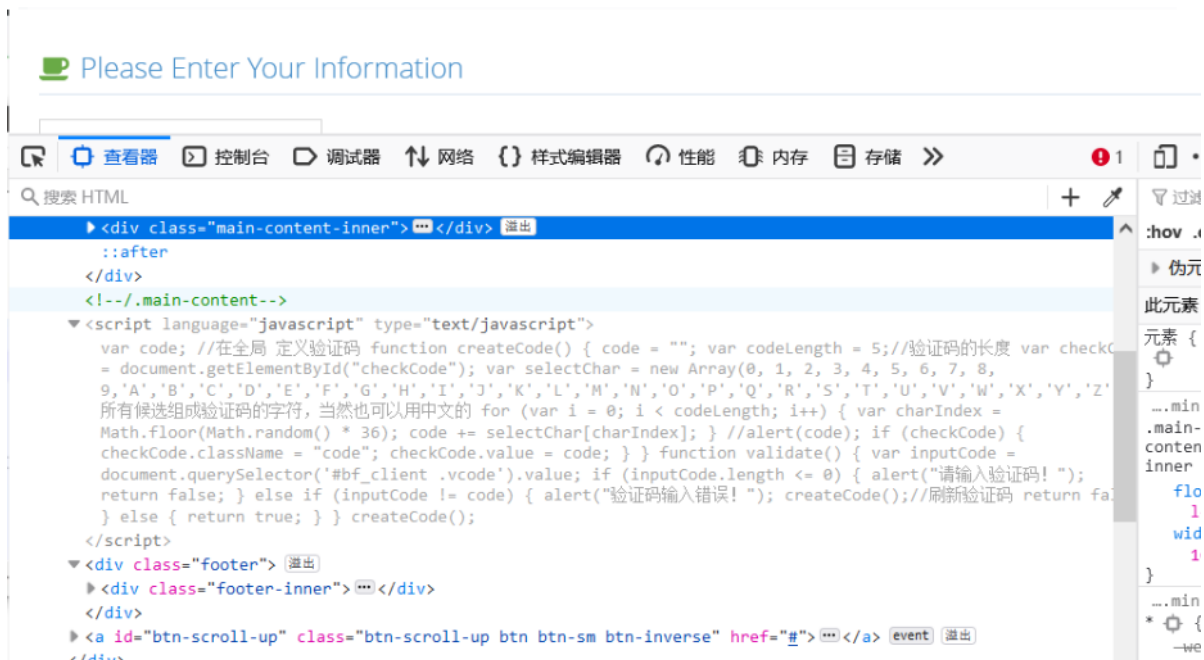


- 1 思路：将数据包的用户名修改重复发送看回显的值是否发生变化。可以看出验证码存在复用的问题。
- 2 利用：输入正确的账号和密码，用之前的验证码，发下登录成功
- 3 源代码分析
- 4 ---传递的参数都不能为空
- 5 ---post提交的验证码也要和session中的vcode相等
- 6 ---漏洞存在原因：session中的vcode没有销毁，导致下一次还能用

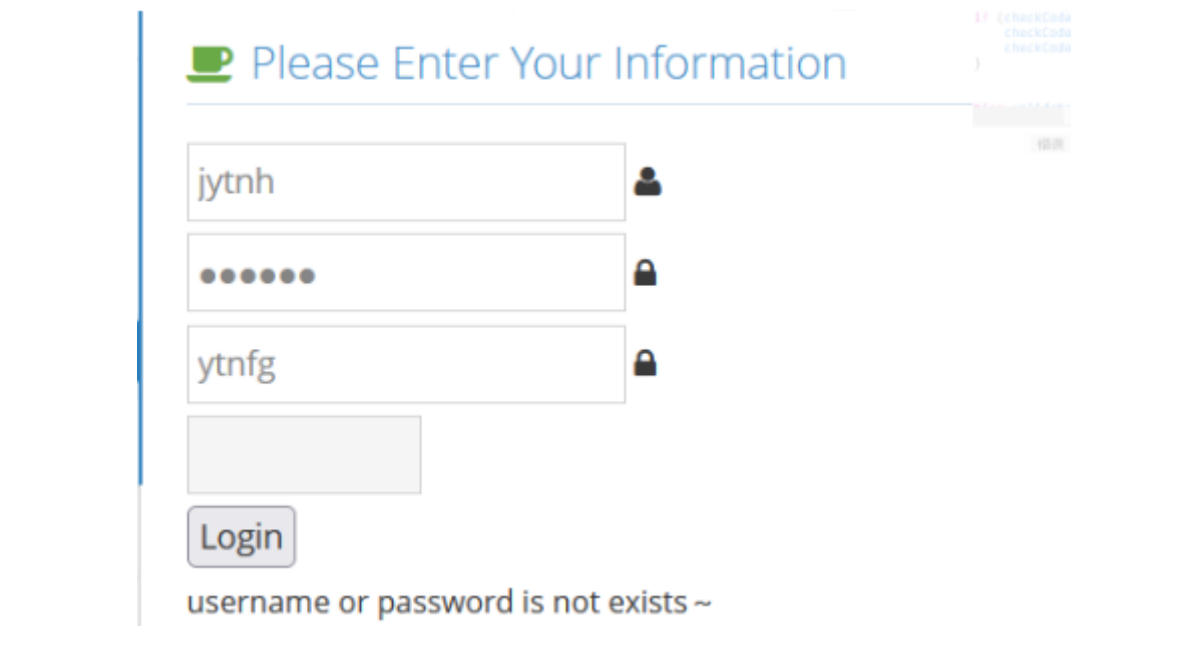
## 2.on client

查看前端源代码（验证码写入了前端）：





这里随意输入验证码绕过（前端代码验证码不执行）：



## 36.2 token 安全

### 36.2.1 Token定义

Token是服务端生成的一串字符串。当客户端第一次登录后，服务器生成一个Token并将此Token返回给客户端，Token可以代替用户名和密码作为身份的验证。每次客户端与服务器端通信，会得到新的Token，基本上述同理，主要是验证中可存在绕过可继续后续测试。

## 36.2.2 token 爆破，token 客户端回显等

1. token 爆破---token后面会跟上一个字符串，如果知道规律可以进行爆破
2. token 客户端回显---token的数据会在前端数据包 (request) 里面显示
3. token固定---虽然有token，但是可以通过上一次的token操作下一次的数据包（表面上有，实际没有）

## 36.2.3 Token靶场练习

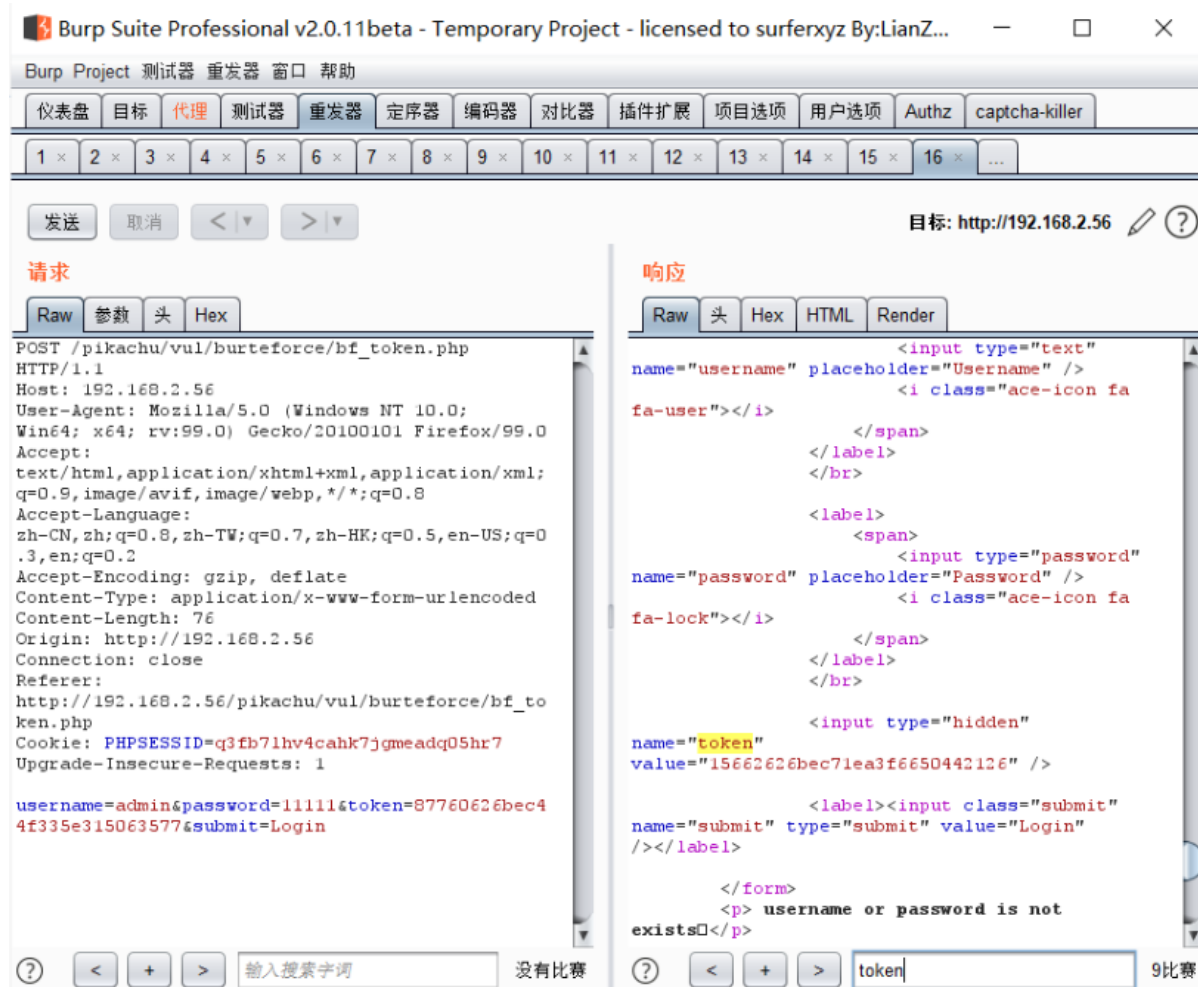
登录查看数据包：

The screenshot shows the Burp Suite Professional interface. On the left, a web application is open with a login form containing the username 'admin', a password field with five dots, and a 'Login' button. Below the button, an error message reads 'Username or password is not exist'. On the right, the 'HTTP History' tab is active, showing a single request to 'http://192.168.2.56:80'. The request details are expanded, showing a POST to '/pikachu/vul/burteforce/bf\_token.php'. The raw request text is as follows:

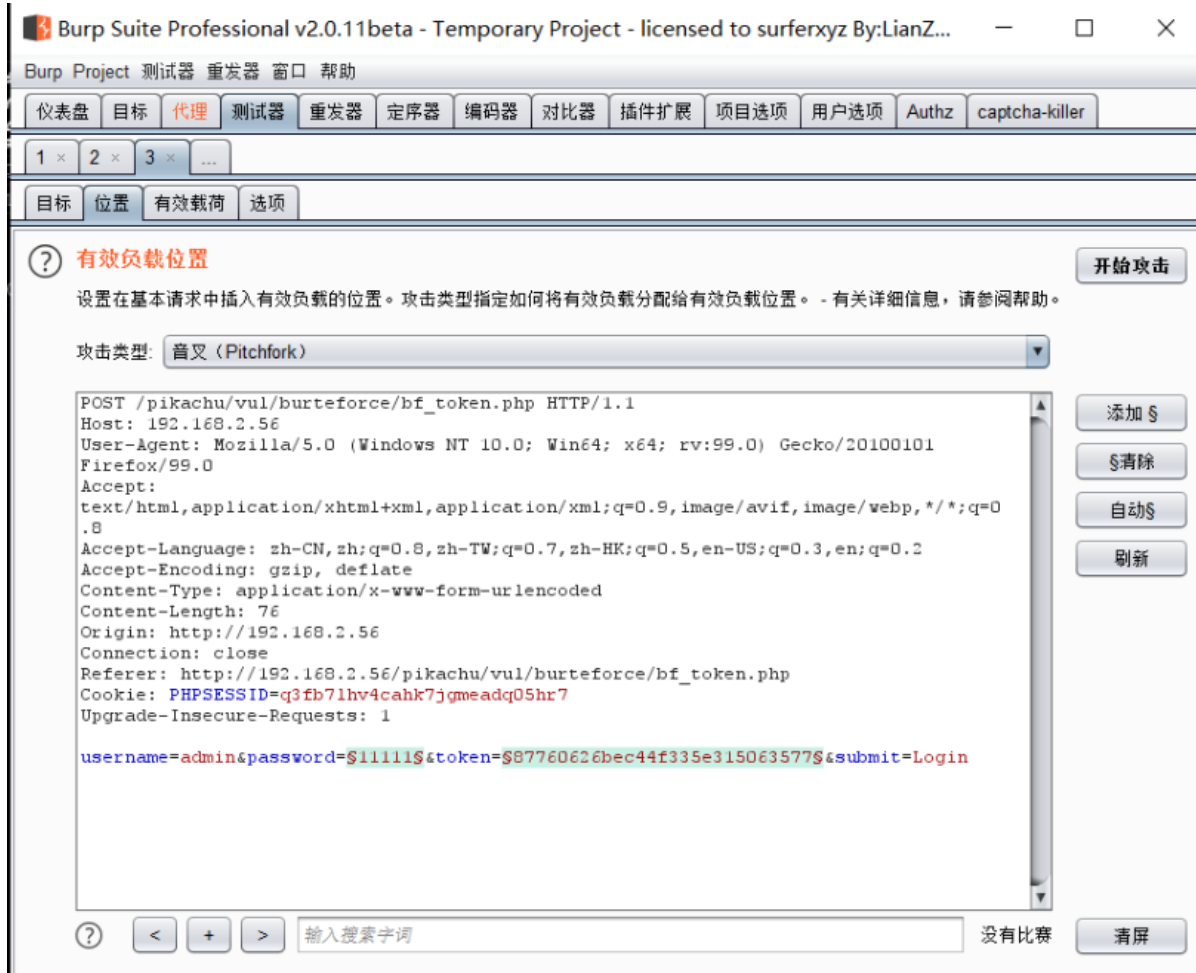
```
POST /pikachu/vul/burteforce/bf_token.php HTTP/1.1
Host: 192.168.2.56
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:99.0) Gecko/20100101 Firefox/99.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 76
Origin: http://192.168.2.56
Connection: close
Referer: http://192.168.2.56/pikachu/vul/burteforce/bf_token.php
Cookie: PHPSESSID=q3fb7lhv4cahk7jgmeadq05hr7
Upgrade-Insecure-Requests: 1

username=admin&password=11111&token=71565626beb3f51366645604168&submit=Login
```

查看前端里也有token，该token是下一次客户端token正确的匹配值：



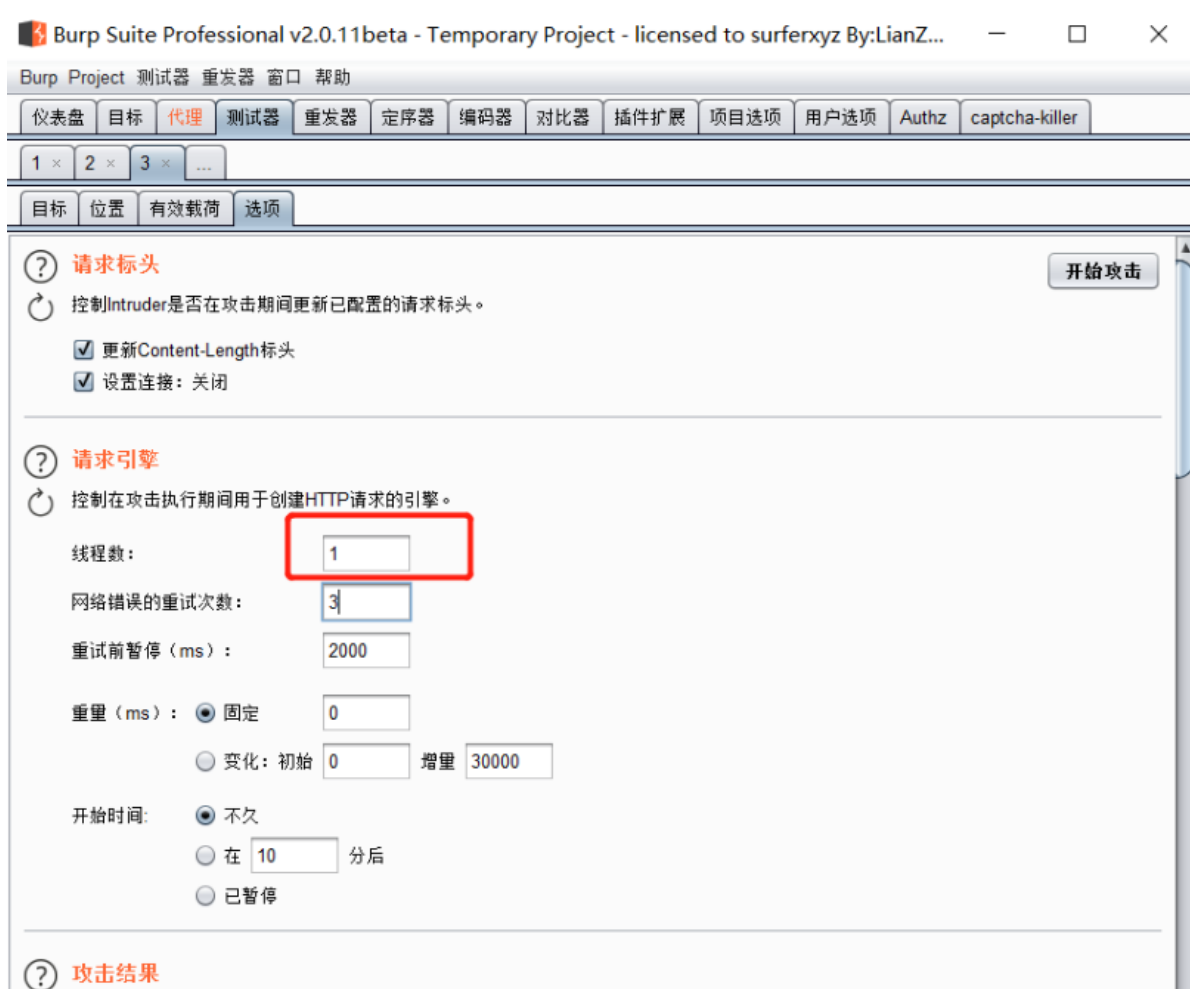
token是用来防爆破的，但是其token值输出在了前端源码中，容易被获取，因此也就失去了防暴力破解的意义。将密码和token设置为攻击参数，攻击类型选择pitchfork：



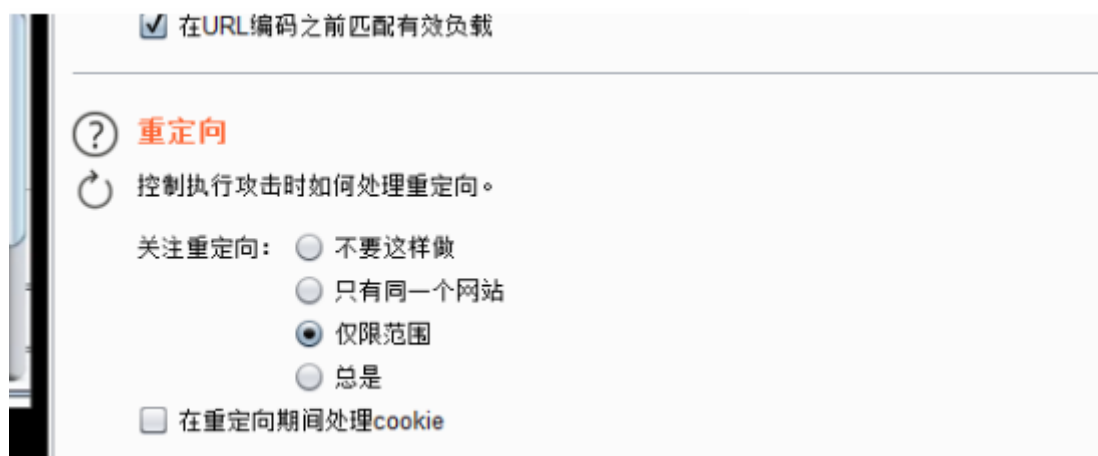
设置第一个参数，添加字典：



设置线程为1（1次只验证一个账号密码）：



设置重定向，仅限范围：



设置grep extract匹配用法，点击添加，查找到token值选择，点击ok：

提取的grep项的定义

×

?

定义要提取的项目的位置。在响应面板中选择项目时，将自动创建相应的设置。您也可以手动更改设置以有效工作。

☒ 定义开始和结束
 

☒ 从后面开始:

☐ 开始偏移:

☒ 以此分隔符结束:

☐ 以固定长度结束:

☐ 从正则表达式组中提取
 

☒ 区分大写和小写字母

☐ 排除HTTP标头
☐ 根据以下选择更新设置

重新获得回复

```

<input type="password" name="password"
placeholder="Password" />
<i class="ace-icon fa fa-lock"></i>
</span>
</label>
</br>

<input type="hidden" name="token"
value="75239626bf331a2589149077333" />

<label><input class="submit" name="submit" type="submit"
value="Login" /></label>

</form>
<p> csrf token error</p>

</div><!-- /.widget-main -->
</div><!-- /.widget-body -->

```

?

<

+

>

token

10比赛

?

Grep - Extract

↺

我们从响应中提取有用信息并将其显示在攻击结果列表中。

☒ 从响应中提取以下项目：

添加

编辑

删除

复制

至顶

From [ value="] to [ />\n\n

<labe...

这里设置第二个参数token，类型为recursive grep：



## 有效载荷集

您可以定义一个或多个有效负载集。有效负载集的数量取决于“位置”选项卡中定义的攻击类型种有效负载类型，并且可以以各种方式定制每种有效负载类型。

有效负载集:  有效载荷数量: 不明

有效载荷类型:  请求数量: 8



## 有效载荷选项[递归搜索]

此有效负载类型从先前攻击请求的响应中提取有效负载。在需要提取有用数据并递归提供漏洞项卡中定义提取grep项目。

选择“extract grep”项以获取有效负载:

From [ value=" ] to [ />\n\n <label...

攻击，发现下面的token值就是上一个的值:

Intruder attack 3

攻击 保存 列

结果 目标 位置 有效载荷 选项

过滤器: 显示所有项目

请求	Payload1	Payload2	状态	错误	Redir...	超时	长	value="	评论
0			200		0		34872	33201626bf4d3cf01d...	
1	111111		200		0		34848	21980626bf4d3e57fb...	
2	admin	21980626bf4d3e57fb90775...	200		0		34893	39940626bf4d3edcc...	
3	21	39940626bf4d3edcc75704...	200		0		34893	55767626bf4d4011b...	
4	123	55767626bf4d4011b58578...	200		0		34893	95869626bf4d40651...	
5	456	95869626bf4d4065174828...	200		0		34893	62736626bf4d41093...	
6	123456	62736626bf4d4109337681...	200		0		34869	98520626bf4d41517...	
7		98520626bf4d41517b9053...	200		0		34848	11420626bf4d41f139...	
8	12121212	11420626bf4d41f13989232...	200		0		34893	48270626bf4d42710...	

请求 响应

Raw 参数 头 Hex

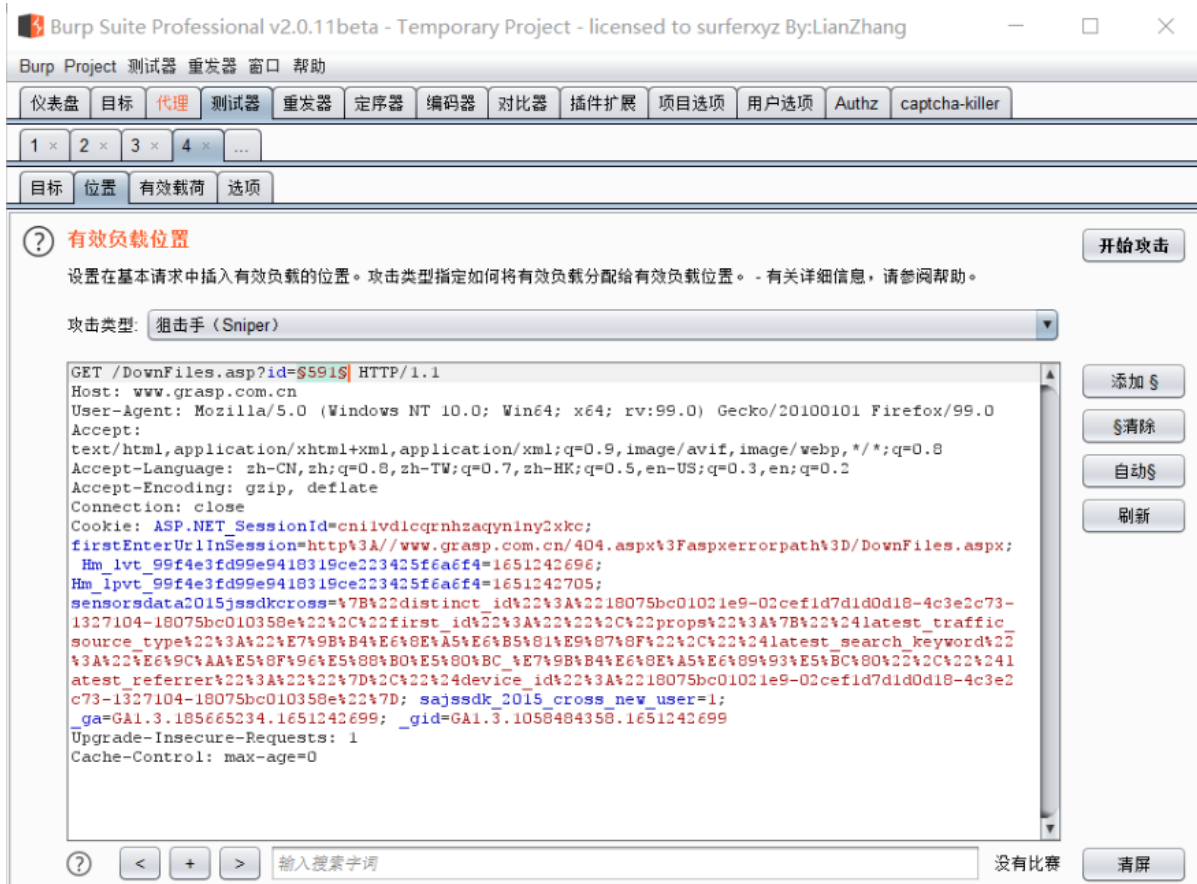
POST /pikachu/vul/burteforce/bf\_token.php HTTP/1.1  
Host: 192.168.2.56  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:99.0) Gecko/20100101 Firefox/99.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8  
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2  
Accept-Encoding: gzip, deflate  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 74  
Origin: http://192.168.2.56  
Connection: close  
Referer: http://192.168.2.56/pikachu/vul/burteforce/bf\_token.php  
Cookie: PHPSESSID=q3fb7lhv4cahk7jgmeadq05hr7  
Upgrade-Insecure-Requests: 1  
  
username=admin&password=456&token=95869626bf4d406517482854945&submit=Login

输入搜索字词 没有比赛

### 36.3 本地某URL下载接口ID值调用遍历测试-实例

```
1 url:www.grasp.com.cn/DownFiles.asp?id=591
```

## 抓包发送给intruder，设置id为变量:



### 设置攻击类型和攻击范围:





查看结果，看是否可以发现其他用户的信息，用来水平越权：

Request	Payload	Status	Error	Timeout	Length	Comment
752	752	302	<input type="checkbox"/>	<input type="checkbox"/>	444	
753	753	302	<input type="checkbox"/>	<input type="checkbox"/>	444	
754	754	302	<input type="checkbox"/>	<input type="checkbox"/>	444	
808	808	302	<input type="checkbox"/>	<input type="checkbox"/>	444	
854	854	302	<input type="checkbox"/>	<input type="checkbox"/>	444	
864	864	302	<input type="checkbox"/>	<input type="checkbox"/>	444	
229	229	302	<input type="checkbox"/>	<input type="checkbox"/>	490	
230	230	302	<input type="checkbox"/>	<input type="checkbox"/>	512	
163	163	302	<input type="checkbox"/>	<input type="checkbox"/>	516	
164	164	302	<input type="checkbox"/>	<input type="checkbox"/>	516	
165	165	302	<input type="checkbox"/>	<input type="checkbox"/>	516	
166	166	302	<input type="checkbox"/>	<input type="checkbox"/>	516	

## 36.4 Callback自定义返回调用安全-漏洞测试-实例

### 36.4.1 什么是callback

一般而言，函数的形参是指由外往内向函数体传递变量的入口，但此处加了callback后则完全相反，它是指函数体在完成某种使命后调用外部函数的出口！这时候应该明白什么叫"回调"了吧，也就是回头调用外部函数的意思。

## 36.4.2 漏洞利用

callback参数可以更改，可以和跨站漏洞结合，在网页源代码搜索传递的参数，如果存在，意味着URL传递的参数会在网页的前端回显，那么，也意味着可以构造XSS漏洞（测试有没有过滤，完不完整）

- 1 上述在实战中如何做到漏洞发现-bp功能点
- 2 原理：逻辑漏洞挖功能点和参数值（关键的参数：id，callback，filename，uid等等）

## 36.5 保存所需数据包

抓包发送到爬虫（这里使用右键点击tools中的find references）：



查看抓到的结果：

http://192.168.2.56/pikachu/vul/burteforce/bf\_server.php 的参考

源	主机	URL	状态	长	请求时间
Target	http://192.168.2.56	/pikachu/vul/burteforce/bf_client.php	200	35348	21:25:57 29 四月 2022
Target	http://192.168.2.56	/pikachu/vul/burteforce/bf_client.php	200	35348	21:25:59 29 四月 2022
Target	http://192.168.2.56	/pikachu/vul/burteforce/bf_client.php	200	35348	21:26:04 29 四月 2022
Proxy	http://192.168.2.56	/pikachu/vul/burteforce/bf_client.php	200	36529	21:11:44 29 四月 2022
Proxy	http://192.168.2.56	/pikachu/vul/burteforce/bf_client.php	200	36529	21:17:11 29 四月 2022
Proxy	http://192.168.2.56	/pikachu/vul/burteforce/bf_client.php	200	36529	21:17:37 29 四月 2022
Scanner	http://192.168.2.56	/pikachu/vul/burteforce/bf_form.php	200	35056	20:59:20 29 四月 2022
Scanner	http://192.168.2.56	/pikachu/vul/burteforce/bf_form.php	200	35056	22:56:35 29 四月 2022
Scanner	http://192.168.2.56	/pikachu/vul/burteforce/bf_form.php	200	35101	22:56:44 29 四月 2022
Scanner	http://192.168.2.56	/pikachu/vul/burteforce/bf_form.php	200	35101	22:56:44 29 四月 2022

请求 响应

Raw 参数 头 Hex

POST /pikachu/vul/burteforce/bf\_client.php HTTP/1.1  
Host: 192.168.2.56  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:99.0) Gecko/20100101 Firefox/99.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8  
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2  
Accept-Encoding: gzip, deflate  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 55  
Origin: http://192.168.2.56  
Connection: close  
Referer: http://192.168.2.56/pikachu/vul/burteforce/bf\_client.php  
Cookie: PHPSESSID=q3fb71hv4cahk7jgmeadq05hr7  
Upgrade-Insecure-Requests: 1

没有比赛

真实目录路径，搜索关键字（这里可以使用 discover content模块过滤搜索内容，相当于帮你筛选）：

内容搜索: http://192.168.2.56/pikachu/vul/burteforce/

控制 设定 网站地图

过滤器: 显示所有项目

按请求类型过滤

- ☐ 仅在范围内显示项目
- ☐ 仅显示请求的项目
- ☐ 仅显示带参数的请求

按MIME类型过滤

- ☒ HTML
- ☒ Script
- ☒ XML
- ☒ CSS
- ☒ 其他文字
- ☒ 图片
- ☒ Flash
- ☒ 其他二进制

按状态代码过滤

- ☒ 2xx [成功]
- ☒ 3xx [重定向]
- ☒ 4xx [请求错误]
- ☒ 5xx [服务器错误]

按搜索关键词过滤

token

- ☐ 正规表现
- ☐ 区分大写和小写字母
- ☐ 搜索不匹配的项目

按扩展名过滤

表示:

非表示:

按注释过滤

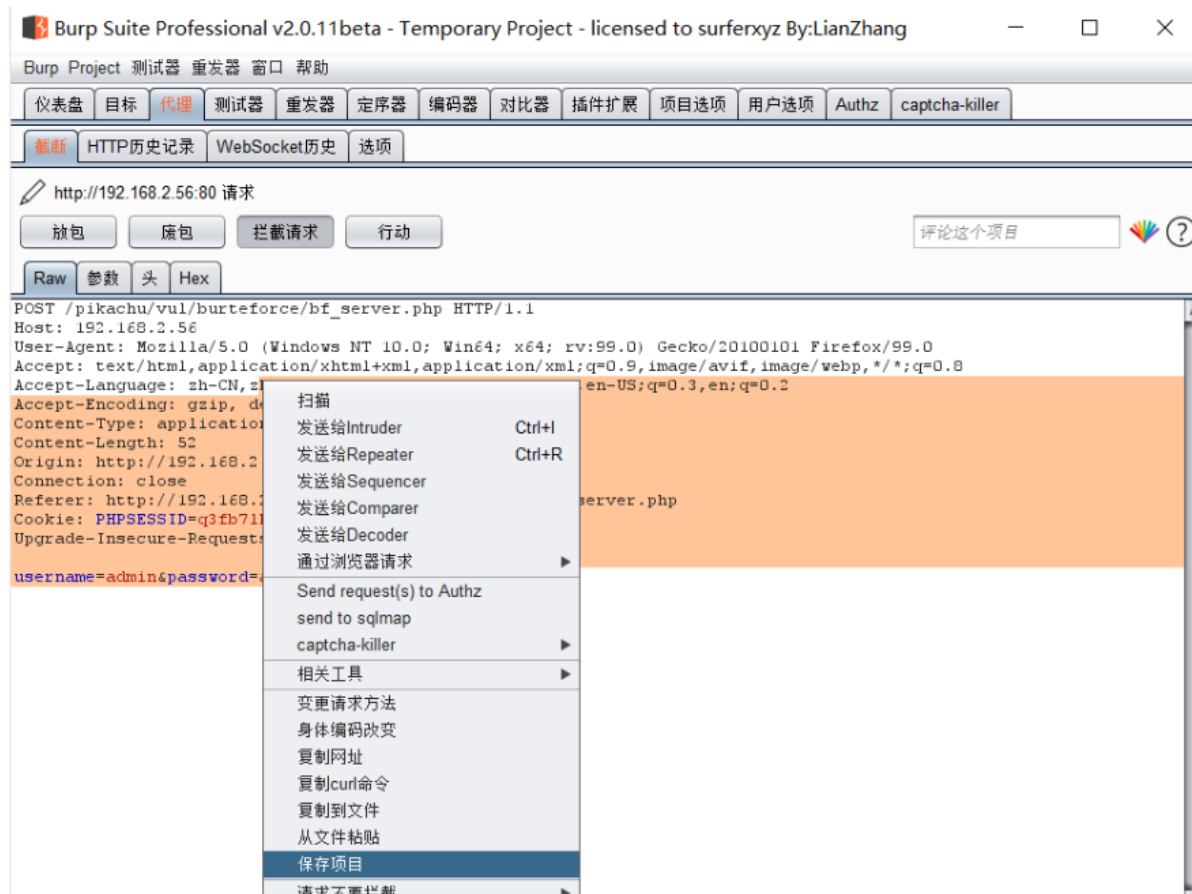
- ☐ 仅显示已评论的项目
- ☐ 仅显示彩色项目

全部显示 全部隐藏 撤消更改

## 内容

主机	方法	URL	参数	状态	长	MIME类型	标
http://192.168.2.56	GET	/		200	2607	HTML	G
http://192.168.2.56	GET	/pikachu/		200	35829	HTML	G
http://192.168.2.56	GET	/pikachu/assets/css...		200	117109	CSS	
http://192.168.2.56	GET	/pikachu/assets/css...		200	78254	CSS	
http://192.168.2.56	GET	/pikachu/assets/css...		200	397241	CSS	
http://192.168.2.56	GET	/pikachu/assets/css...		200	121503	CSS	
http://192.168.2.56	GET	/pikachu/assets/css...		200	792	CSS	
http://192.168.2.56	GET	/pikachu/assets/font...		200	27767	CSS	

导出数据包 (save bom) :



资源:

- 1 <https://www.lanzous.com/i1z2s3e>
- 2 <https://www.cnblogs.com/nul1/p/12071115.html>
- 3 <https://github.com/c0ny1/captcha-killer/releases/tag/0.1.2>  
<https://github.com/bit4woo/reCAPTCHA/releases/tag/v1.0>

