[红日安全]Web安全Day10 - 重放攻击实战攻防

红日安全 / 2019-10-25 09:45:38 / 浏览数 5080 安全技术 WEB安全 顶(0) 踩(0)

本文由红日安全成员： misakikata 编写，如有不当，还望斧正。

大家好，我们是红日安全-Web安全攻防小组。此项目是关于Web安全的系列文章分享，还包含一个HTB靶场供大家练习，我们给这个项目起了一个名字叫
Web安全实战
，希望对想要学习Web安全的朋友们有所帮助。每一篇文章都是于基于漏洞简介-漏洞原理-漏洞危害-测试方法（手工测试，工具测试）-靶场测试（分为PHP靶场、JAVA靶

重放攻击

1. 漏洞简介

 首先简单看一下百度百科对重放攻击的简介：重放攻击(Replay
Attacks)又称重播攻击、回放攻击，是指攻击者发送一个目的主机已接收过的包，来达到欺骗系统的目的，主要用于身份认证过程，破坏认证的正确性。重放攻击可以由发起

2. 漏洞原理

重放攻击的基本原理就是把以前窃听到的数据原封不动地重新发送给接收方。很多时候，网络上传输的数据是加密过的，此时窃听者无法得到数据的准确意义。但如果他知道

3. 漏洞危害

重放攻击本身只是一种行为和方式，并不会直接造成系统的危害，可能在某些系统中，过多和高频次的重复会对系统造成压力。重放攻击的重要点在于重放的是可以造成目的

重放攻击主要是针对系统没有效验请求的有效性和时效性，对于多次请求执行，系统将多次响应。在重放攻击利用最多的形式中，短信轰炸算是重放攻击最直接的利用表现。

4. 常见漏洞类型

1. 短信轰炸

短线轰炸算是重放攻击中最为直接的利用形式，当系统端没有效验请求的时间差或者只在前端做请求限制的时候，可以无限的请求短信来达到短信轰炸的目的。例如，如下A

多次请求后可以在手机上看到请求到的短信

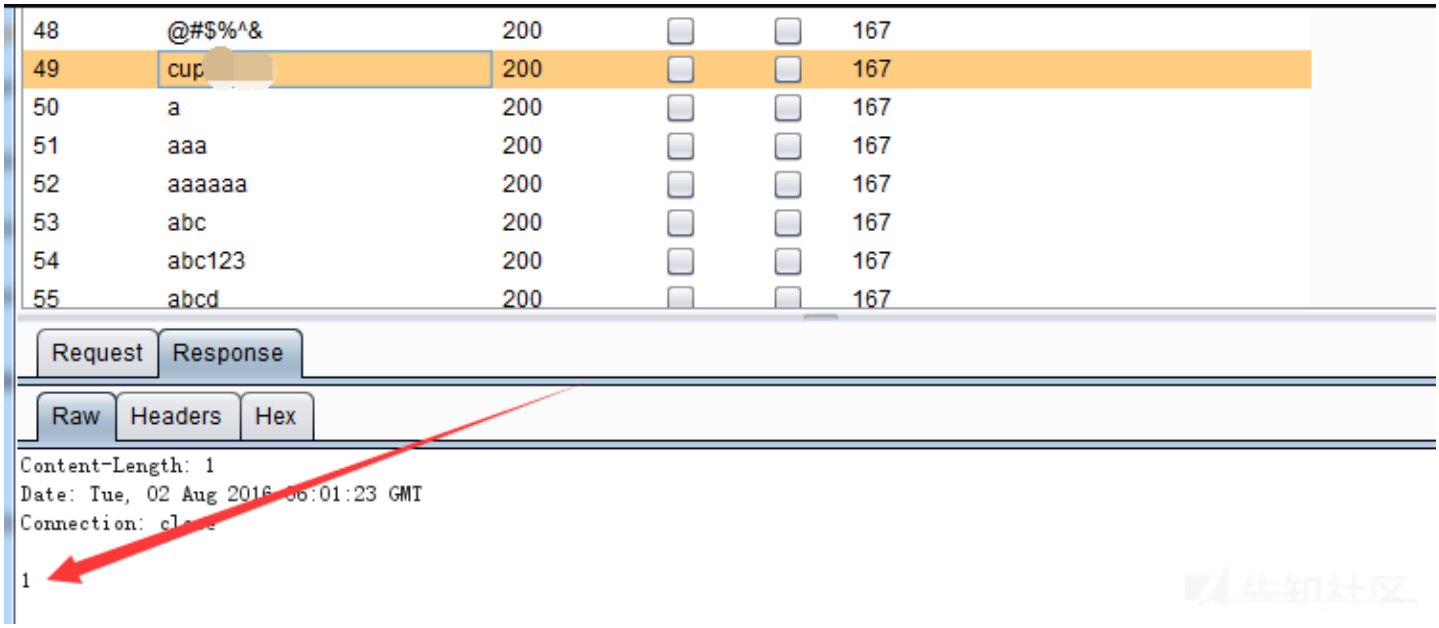
2. 暴力破解

暴力破解是重放攻击中，典型的非只重放而达到的攻击类型，而是利用重放这个动作来达到暴力破解的目的。当系统端未做请求验证和错误次数限制时，就可以根据字典或者

a. 暴力破解密码

当用户登陆时，缺少验证码或者验证码不失效，并且账号没有错误的次数限制。可以通过暴力破解碰撞密码来登录。例如此处，暴力破解原密码来登陆绑定账号。

此处验证码只判断是否存在，并不失效，且可以多次尝试绑定账号，例如如下，当返回为1的时候就是密码正确，绑定成功。

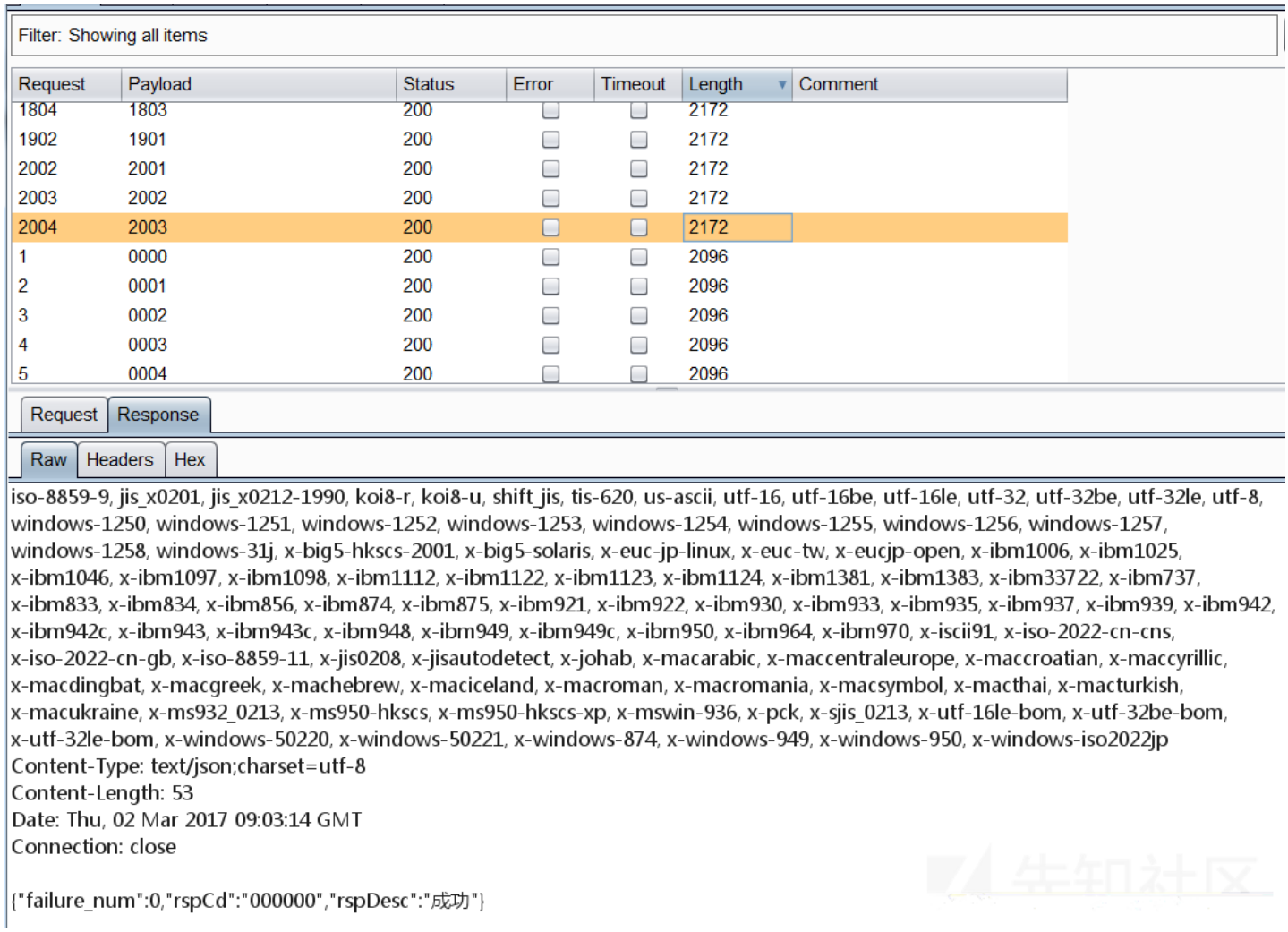


b. 暴力破解验证码

□

当我们申请修改账号密码等操作时，往往需要给手机号或者邮箱发送一个验证码，当需要修改他们或者越权操作的时候并不一定可以通过修改接收手机或邮箱来收到验证码，

对此请求多次重放后发现仍然返回修改密码失败，说明验证码可以多次使用，这种情况下很有可能是验证码在没有正确验证使用时，后台并不会失效。那么我们尝试爆破验证



c. 暴力破解参数

□ 此情况大都在尝试越权的时候，还有尝试修改某些不可知但是可预测的参数，例如此篇文章：

重置凭证可暴破

d. 暴力破解hash密码

此种暴力破解类似破解密码，但此种一般不需要考虑某些验证条件，常在获取到主机权限后，利用hash抓取工具获得，例如Windows平台的hash抓取工具：mimikaze，pwdump7等。获取到Windows的NTLM。

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:44f077e27f6fef69e7bd834c7242b040:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

常见的爆破工具：Ophrack、John the Ripper、hashcat

这种方式都需要提前准备彩虹表，当然kail上也有集成，同时也有默认字典。

3. 重放支付

这种情况往往出现在支付订单的时候，支付到最后一个请求包时，系统收到请求就会确定已支付下单。这时候在系统没有做出准确效验的时候就会根据是否支付成功的验证字

但这种情况，现在已经很少会遇到，上一次遇到还要追溯到去年初了。

4. 重放修改密码

在很多时候，我们修改密码等操作的时候，是分几步完成的，例如先验证手机验证码，跳转在修改密码。如果在最后确认修改的时候抓包多次重放，可以达到免验证来达到修

```
POST /userpwd?p=1 HTTP/1.1
Host: xxx.com

phone=13111111111&code=123456
```

当我们如上的去请求验证码效验的时候，如果通过会跳往第二个页面修改密码

```
POST /userpwd?p=2 HTTP/1.1
Host: xxx.com

phone=13111111111&pwd=123456&newpwd=123456
```

当只是简单的重置的时候，先不谈越权问题，这个包都可能造成多次修改多次重置密码。而并不用验证。

```
POST /userpwd?p=2 HTTP/1.1
Host: xxx.com

phone=13111111111&code=123456&pwd=123456&newpwd=123456
```

在修改密码的时候遇到也携带了其他的参数，例如之前的短信验证字段，那么就不一定会造成越权，但可能会有多次重放修改密码的可能。这时候如果需要修改他人密码，就

```
POST /userpwd HTTP/1.1
Host: xxx.com

email=qq@qq.com&code=123456
```

有些系统在重置密码的时候并不是需要各种验证，而是你申请修改就会给你发送重置的密码到你的注册邮箱。例如如上数据包，当验证存在邮箱的时候，只需要输入图片验证

5. 条件竞争

条件竞争是后台对共享数据读写的时候，多线程没有对共享数据执行线程锁，导致在多个线程获取到的值并不是当前线程操作的实时值，典型的例子是，一份钱买多份。

例如去年护网杯的Itshop，此处给出WP以便参考：https://www.codercto.com/a/31463.html

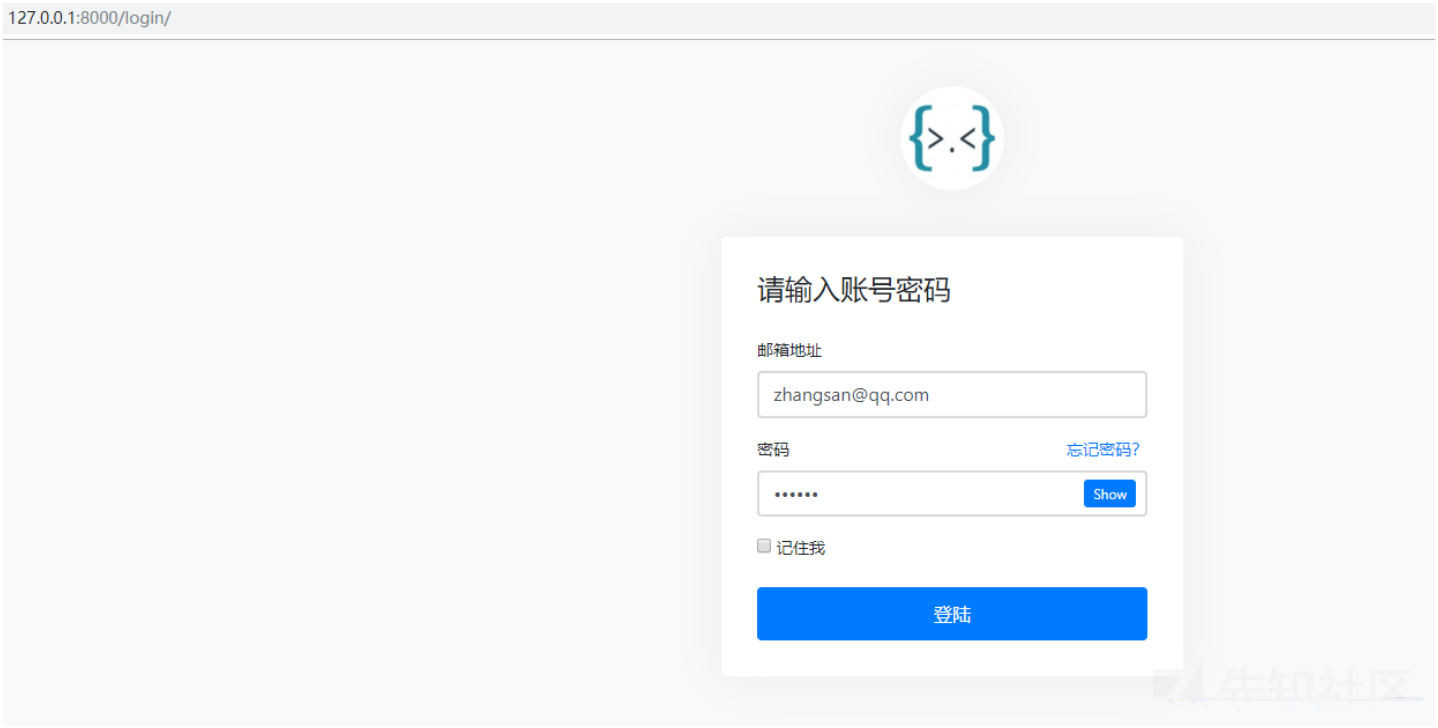
5. 漏洞靶场

漏洞环境：Django2.2、python3

此处利用的是之前写的一个bug平台，当验证会提示如下时，可以根据提示的不同来判断密码是否正确，当密码正确的时候就会跳转到内部页面。

```
def login(request):
   if request.method == 'POST':
       login_form = forms.UserForm(request.POST)
       message = '[redacted]'
       if login_form.is_valid():
           username = login_form.cleaned_data.get('username')
           password = login_form.cleaned_data.get('password')
```

```
            try:
                user = models.User.objects.get(name=username)
            except :
                message = '■■■■■■'
                return render(request, 'login/login.html', locals())

            if user.password == password:
                request.session['is_login'] = True
                request.session['user_id'] = user.id
                request.session['user_name'] = user.name
                return redirect('/index/')
            else:
                message = '■■■■■■'
                return render(request, 'login/login.html', locals())
        else:
            return render(request, 'login/login.html', locals())

    login_form = forms.UserForm()
    return render(request, 'login/login.html', locals())
```



抓包登陆，在没有验证码，且csrf_token在没有起到唯一性的时候，可以通过爆破密码登陆。

把数据包丢到Intruder中，多次爆破后发现当密码正确的时候会产生302的跳转。

201	qweqwe1234	302			363
0		200			2523
1	123456789	200			2523
2	a123456	200			2523
3	123456	200			2523
4	a123456789	200			2523
5	1234567890	200			2523
6	woaini1314	200			2523
7	qq123456	200			2523
8	abc123456	200			2523

Request Response

Raw Params Headers Hex

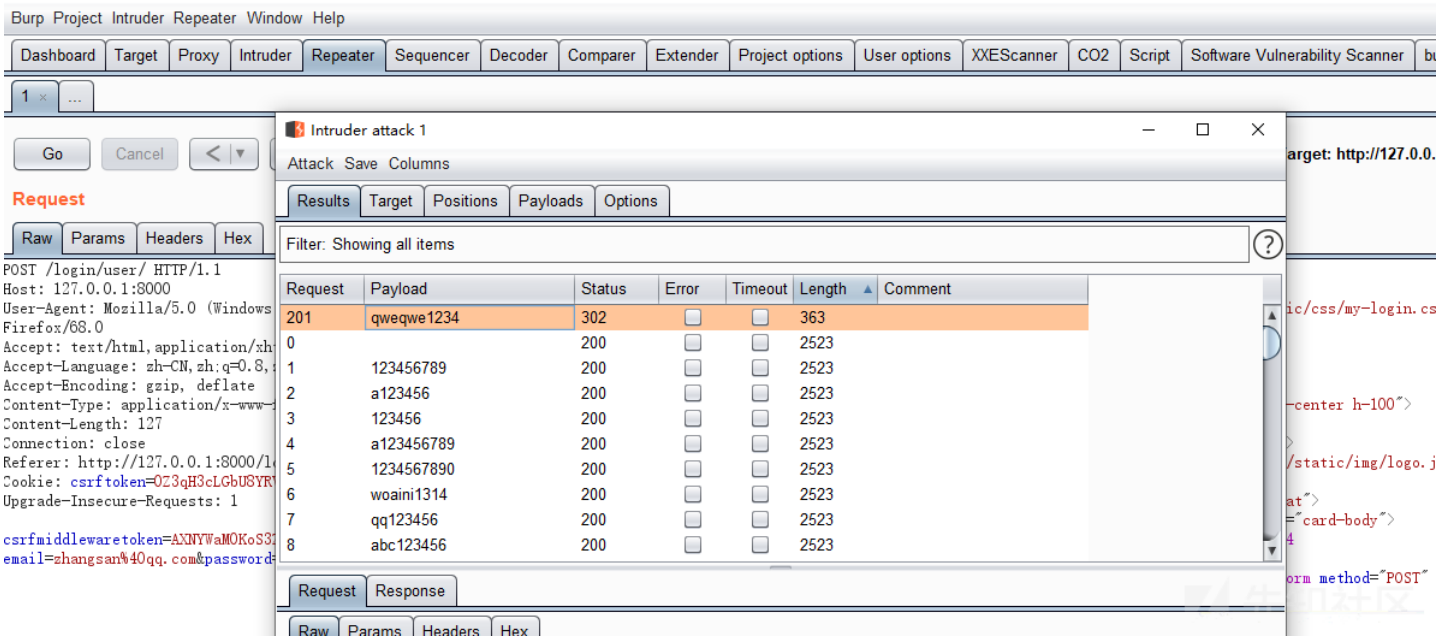
```
POST /login/user/ HTTP/1.1
Host: 127.0.0.1:8000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 128
Connection: close
Referer: http://127.0.0.1:8000/login/
Cookie: csrftoken=OZ3qH3cLGbU8YRVIYHj0a0uSiZlSAPX080V2PnpSJ2rTOWI9N6tUA4NO76nuIooY
Upgrade-Insecure-Requests: 1

csrfmiddlewaretoken=AXNYWaMOKoS3ZbbmKAuz7zsZUPdGdOwoIYFA4uZ7NfpOPgYNzZEtxPLVJWfilnXy&email=zhangsan%40qq.com&passw
ord=qweqwe1234
```

漏洞修复：添加验证码，虽然此处可以添加框架自带的验证码，但建议使用请求式验证码。如不能使用验证码也可以给账号登陆错误次数做一次限制。

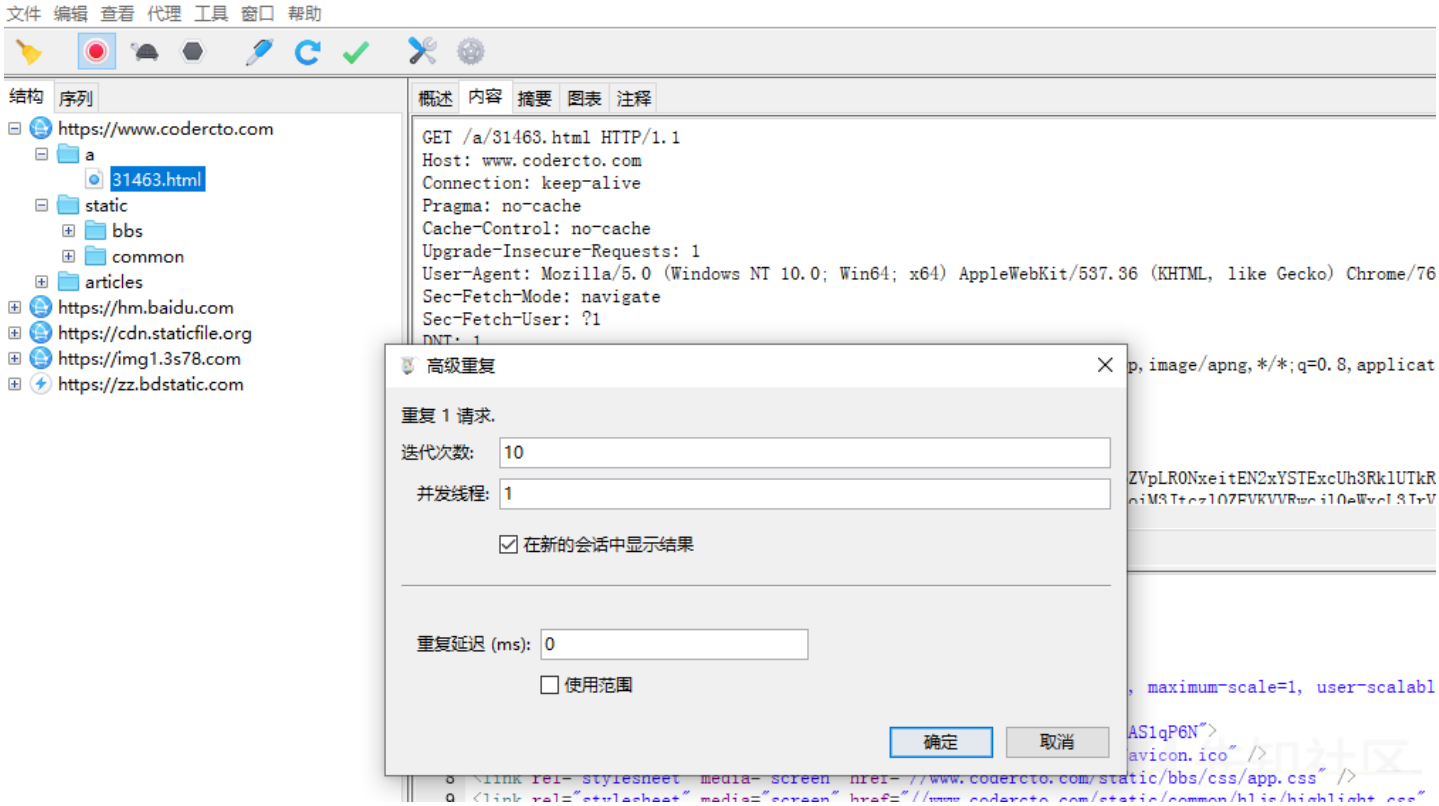
6. 漏洞测试工具

重复攻击一般采用可以抓包的工具都可以重复，例如：Charles、burp等。此处较为常用burp。因为在payload上，处理较为灵活，当然如果需要的只是重放，Charles应该

burp: https://portswigger.net/burp



Charles：https://www.charlesproxy.com/



7. 漏洞修改

1. 添加图片验证码，为了应对偏爆破类的重放攻击，添加验证字段是最简单有效的手段。当然你要保证验证是在一次使用后及时失效。
2. 限制请求次数，有些地方并不适用于添加验证码，或者不能添加验证码。这时候针对同一账户的错误次数限制就显得很有必要。例如，当错误次数连续达到五次的时候，
3. 效验验证码和用户身份，某些重放攻击是利用了手机号和验证码之间的不对应性，特别是在修改密码等处，这时候需要把验证码和请求的用户手机号做联系，当重放或者

上一篇：浅谈Windows身份认证及相关攻击方式 下一篇：SQL盲注的简单分析

1. 0 条回复
   - 动动手指，沙发就是你的了！