在不久前参加了一次众测项目，需对某厂商的系统进行漏洞挖掘

在测试一套系统时，发现了很有意思的接口，可以操作另外两个站的输出点，以此导致多处 XSS 触发

0x01：初探 Vulnerabilities

系统的 URL 是这样的：https://xxx.com/passport/?fromUrl=https://xxx.com/

显然这样的 url 容易出现 Open Redirect XSS CRLF 注入的漏洞，其次还有以下这样的参数：

redirect ref redirect_to redirect_url url jump jump_to target to link linkto domain server

比较随意的测试一番

https://xxx.com/passport/?fromUrl=https://www.baidu.com

https://xxx.com/passport/?fromUrl=https://www.baidu.com.eval.com

https://xxx.com/passport/?fromUrl=https://www.baidu.com@www.eval.com

https://xxx.com/passport/?fromUrl=javascript:alert(1)

https://xxx.com/passport/?fromUrl=xxxx%0D%0ASet-Cookie:hacker=crlf

（CRLF 注入是否存在，查看 cookie 值是否包含 hacker=crlf 字样即可）

简单测试后发现并没有触发以上的漏洞

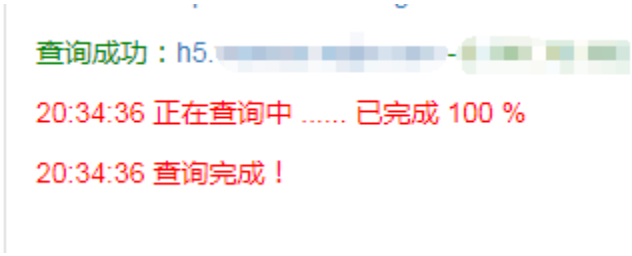
So 只能深入到系统的功能点和接口做测试

0x02：深入虎穴

这边登录至系统进一步挖掘

（PS：登录后系统提示需要填写 xxx 信息，这样的情况我个人习惯会插入”><img/src=1>进行注册）

经过前期的信息收集下，使用子域名爆破工具，成功枚举出 h5.xxx.com 这样的手机端网站



由于我们目前处于 xxx.com 这个域内的登录状态，所以我现在访问：

https://h5.xxx.com/ 也是一样处于登录状态（这种情况在大厂商中的账号登录以及一些 SSO 单点中较为常见）

点击右上方“我的” 进入个人中心页面

这边跳转进入到个人中心

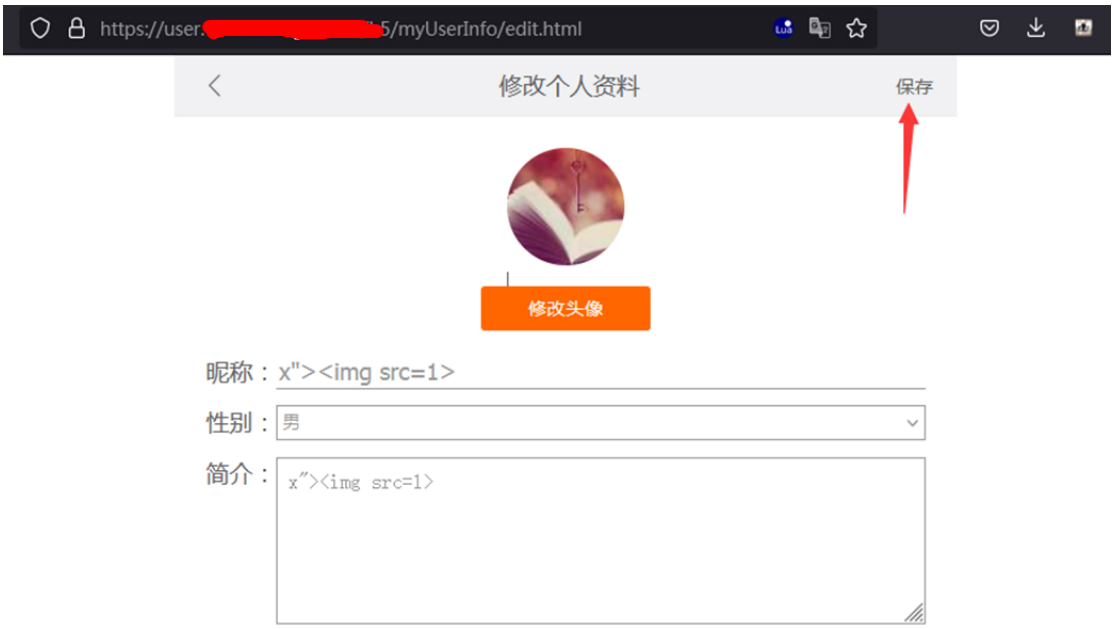
看到个人资料这一功能，这代表着 可能存在

- 修改资料导致 XSS
- 越权修改他人资料信息

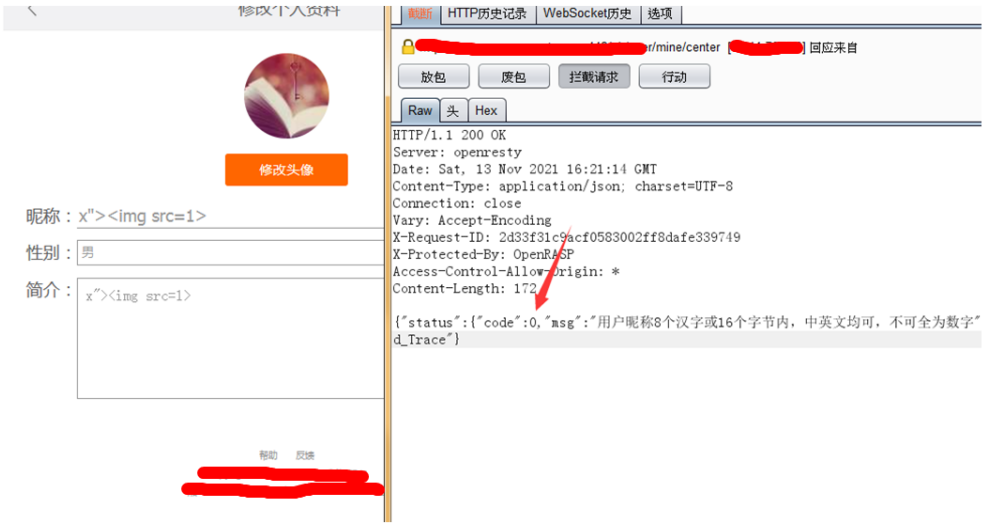
- 删除他人信息
- 替换某参数的值越权查看他人资料
  等等.....一系列的逻辑问题

这边修改个人资料并且使用 XSS 镶入页面的上下文　x"><svg/onload=alert(1)>

截图的时候忘记把 payload 改过来了 将就看看 ：)

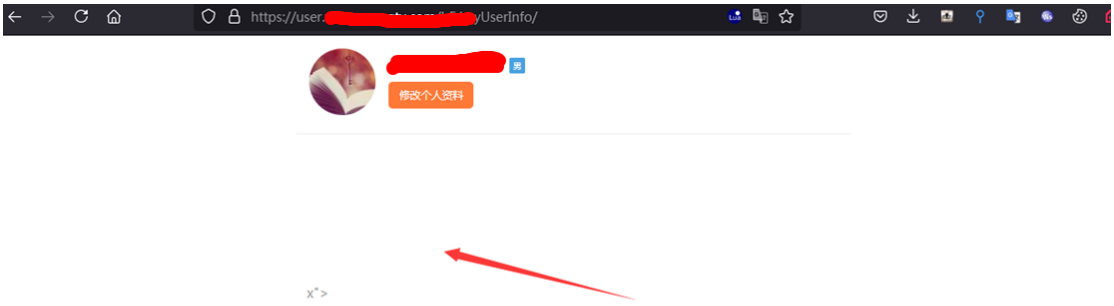


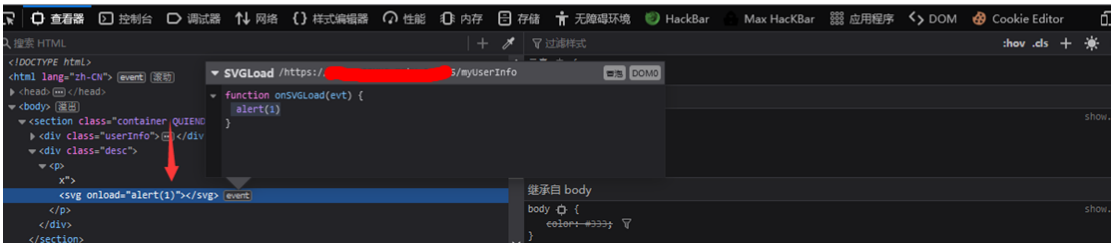
保存后，发现对用户名有限制，通过 burpsuite 修改返回包，但是无果



So 只能重新填写打一遍了，只修改简介里为 payload 昵称任意填写

最后成功在个人资料的简介里插入代码：https://user.xxx.com/h5/myUserInfo/

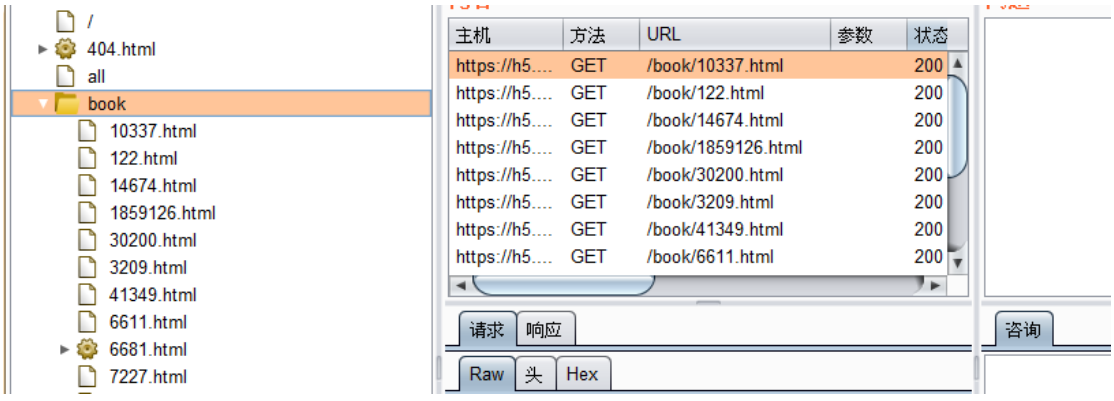




从上述来看 并没有太大用处

后续在想能否找到让他人通过我的个人资料 ID 值就可以预览到我的资料呢？

利用 BurpSuite 爬虫功能爬取了页面上的功能点接口



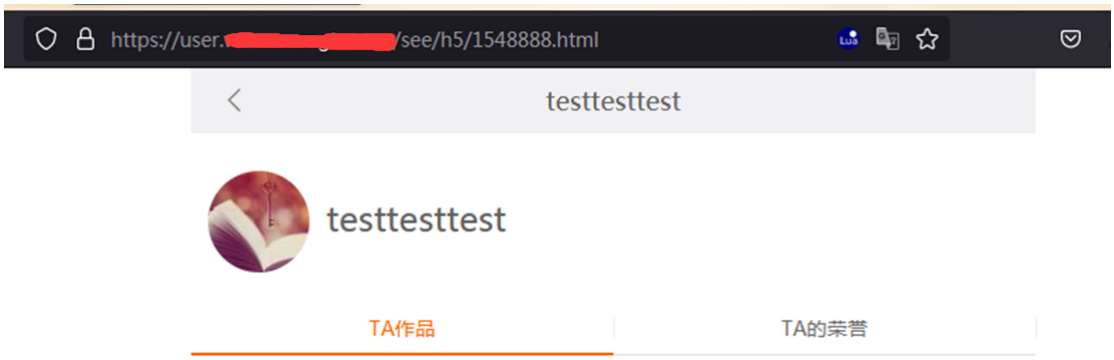
最终在 https://h5.xxx.com/book/xxxx.html 的书籍作者中，找到了可以查询笔者的接口

https://user.xxx.com/see/h5/1574.html

拼接我个人资料的 ID 值 试试水：

https://user.xxx.com/see/h5/1548888.html

本来以为很容易就挖掘到存储 XSS 了，但事实并非如此，找到个人资料接口访问后，那个简历的信息并没有展示在页面的上下文



只显示了 我当时登录系统的作者笔名

从上所述分析，现在只需要找到一处可以修改作者笔名的功能，并且不限制输入字数的，那么便有机会触发存储 XSS

经过一番接口爬取下 and 目录扫描，最终找到了一处没有任何过滤机制的网站下成功修改到了作者笔名：https://user.xxx.com/www/

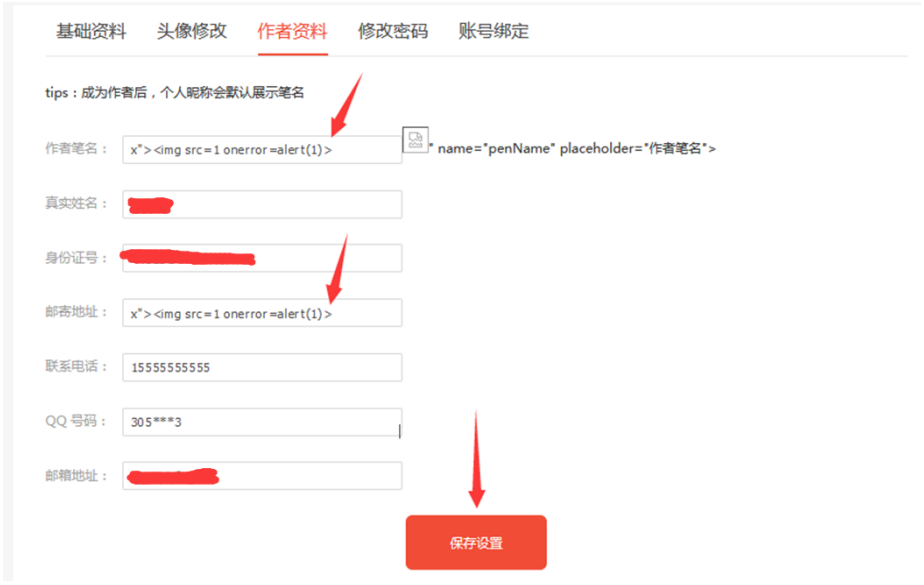
修改作者笔名的接口较为隐蔽，藏在这个作者资料中。。

（因为上述联动了几个网站，我以为这个作者资料是修改的本站账号信息，后来发现这个接口，可以让前面系统上的笔名同时生效）

https://user.xxx.com/www/userinfo/contact.html

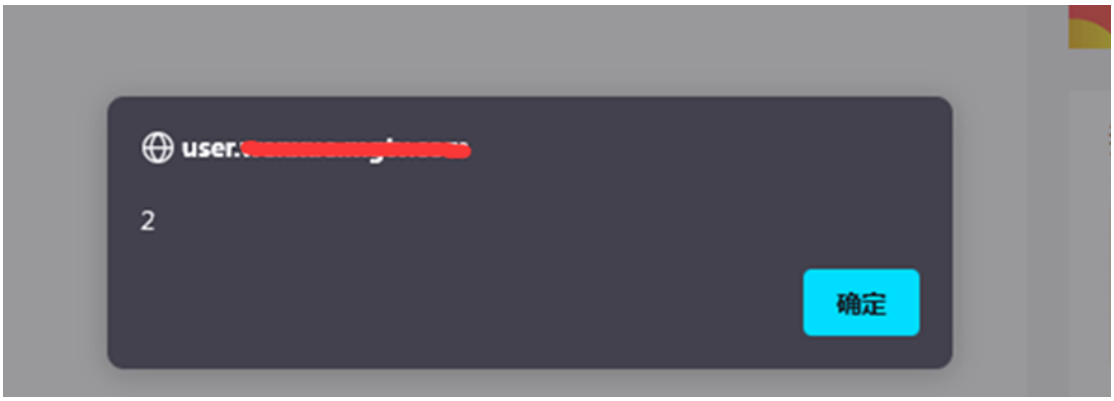
那么这里就根据规定填写资料就可以了，重点就是在作者笔名中插入 payload

点击保存设置进行盲打！



最后就是到了见证奇迹的时候了。

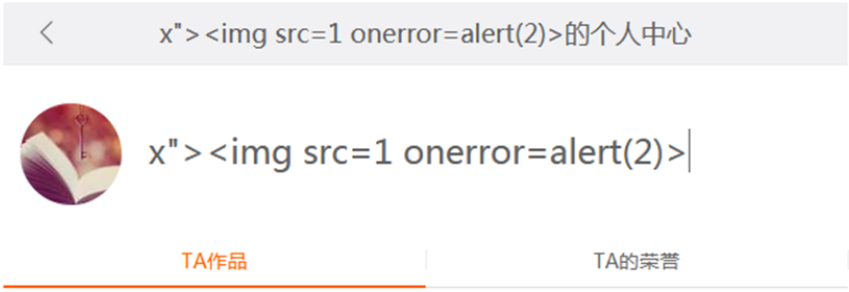
在本站点中：https://user.xxx.com/www/xxx/?xx=1548888 触发漏洞



然后，回到上述步骤中，访问所拼接获得的个人资料 url 地址

接口一：https://user.xxx.com/see/h5/1548888.html

访问后发现上下文中并没有执行 Javascript



但是通过“TA 的荣誉”可以让漏洞成功触发（也就是接口二）

接口二：https://user.xxx.com/see/h5/xxx.html?id=1548888

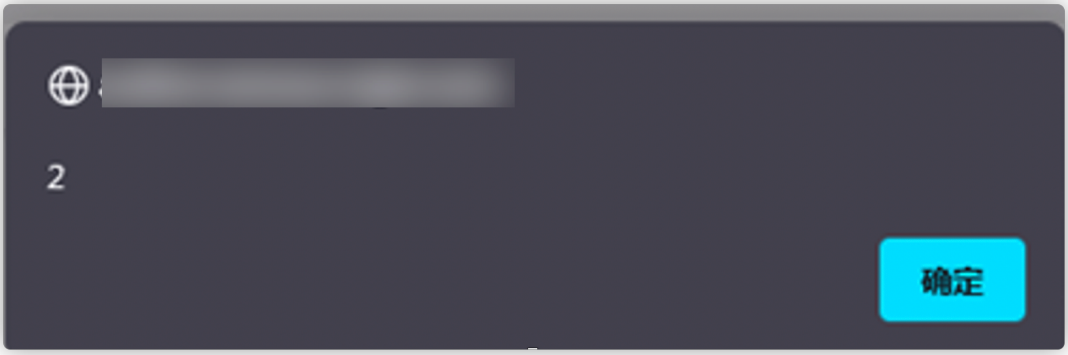
访问后成功触发漏洞

当用户在浏览我的荣誉时，这个 xss 就会触发，构造恶意 JS 即可盗取信息

回到最上面的注册系统中，它其中还包含一个后台：

https://author.xxx.com/

在一处上下文中也成功触发了



0x03：总结

该漏洞本无法利用，但细心挖掘则通过一处致命的修改功能达到想要的成果

1、初探安全漏洞

2、深入挖掘功能点

3、巧用 BurpSuite 爬取页面上的 URL

4、寻找漏洞可能会触发的位置