接口文档下的渗透测试

原创六号刃部 酒仙桥六号部队
2020-10-14原文

这是 **酒仙桥六号部队** 的第 **89** 篇文章。
全文共计1978个字，预计阅读时长7分钟。

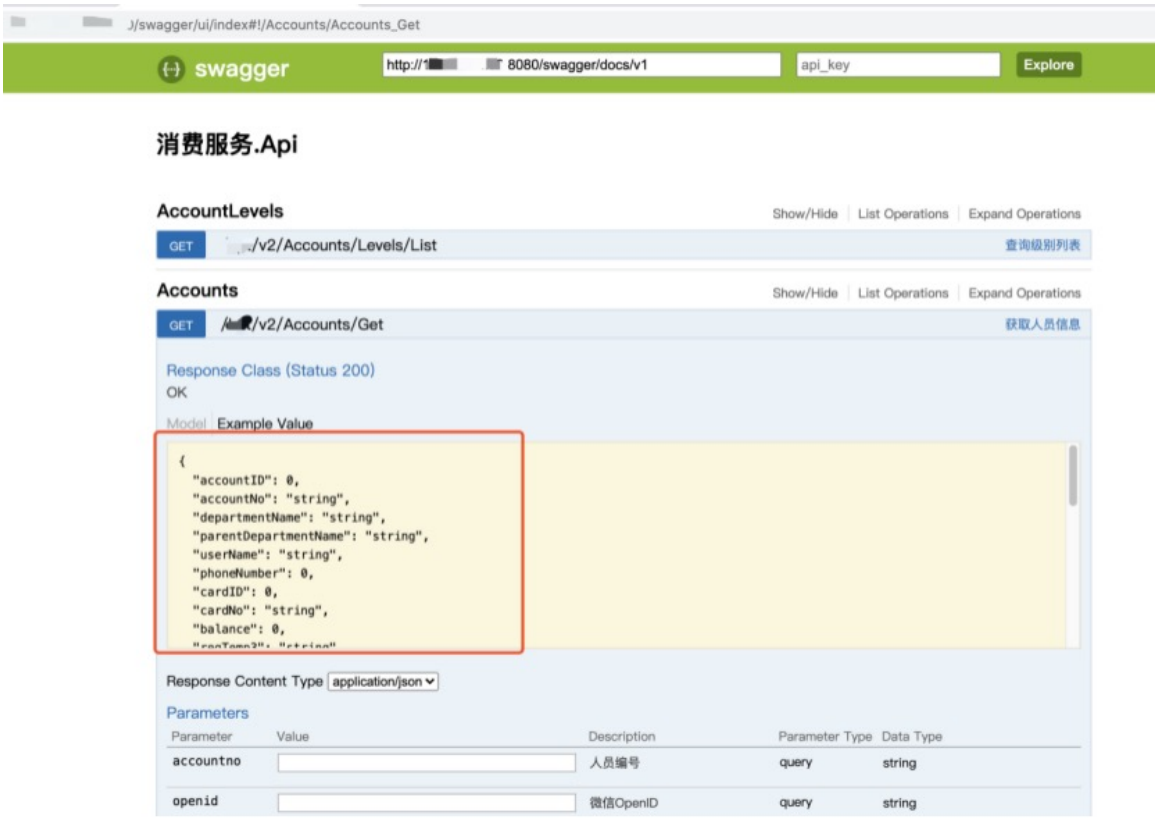
接口文档背景

随着前后端分离架构的优势越来越明显，前后端分离的应用场景也越来越广，如今前后端分离已成为互联网项目开发的业界标准使用方式，而为了前后端程序员在实际开发中能够有统一的接口文档去调试，因此也随着衍生出了很多API接口文档以及调试工具，如swagger、docway、yapi、Web Api HelpPage等。再结合之前挖掘SRC以及甲方工作中也发现过多处这种问题，汇总案例，输出一篇Swagger UI接口文档下的测试文章。

认识Swagger

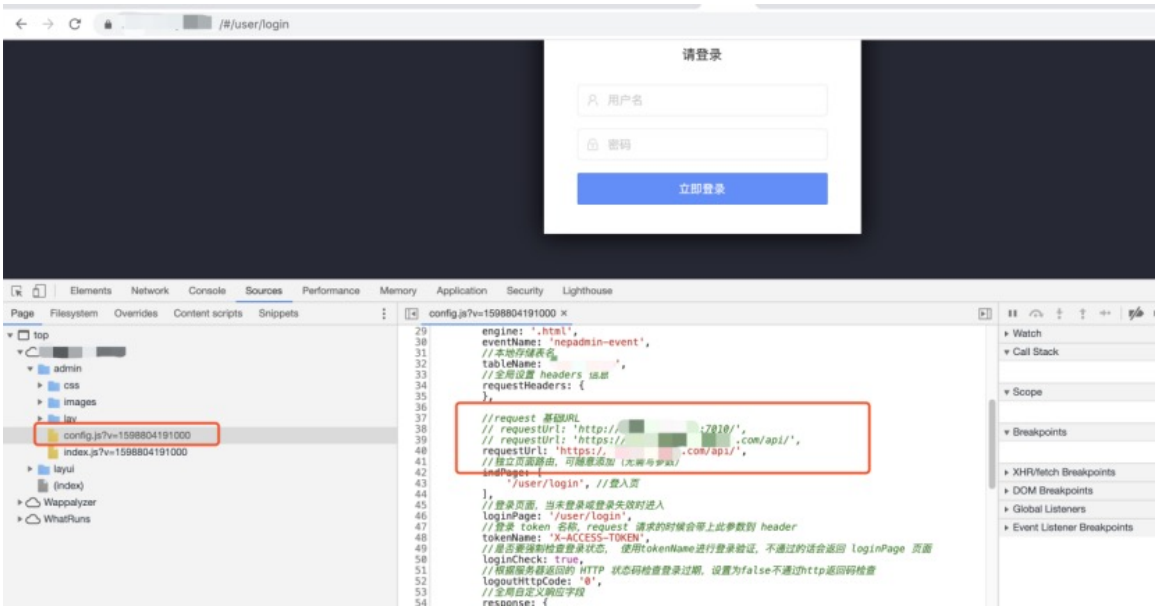
Swagger是一个规范和完整的框架，用于生成、描述、调用和可视化RESTful风格的Web服务，JAVA在金融机构开发语言的地位一直居高不下，而作为JAVA届服务端的大一统框架Spring，便将Swagger规范纳入自身的标准，建立了Spring-swagger项目，所以在实际测试环境中，基于spring框架的swagger-ui接口展示及调试文档页面最为常见。我们先来看某个Swagger UI页面，如图所示，接口中存在查询用户信息、上传文件等多个敏感操作。

在每个接口中也有详细的参数介绍，包括参数类型等，再也不用去fuzz接口参数了，直接根据参数类型构造参数就完事了~



如何发现Swagger UI

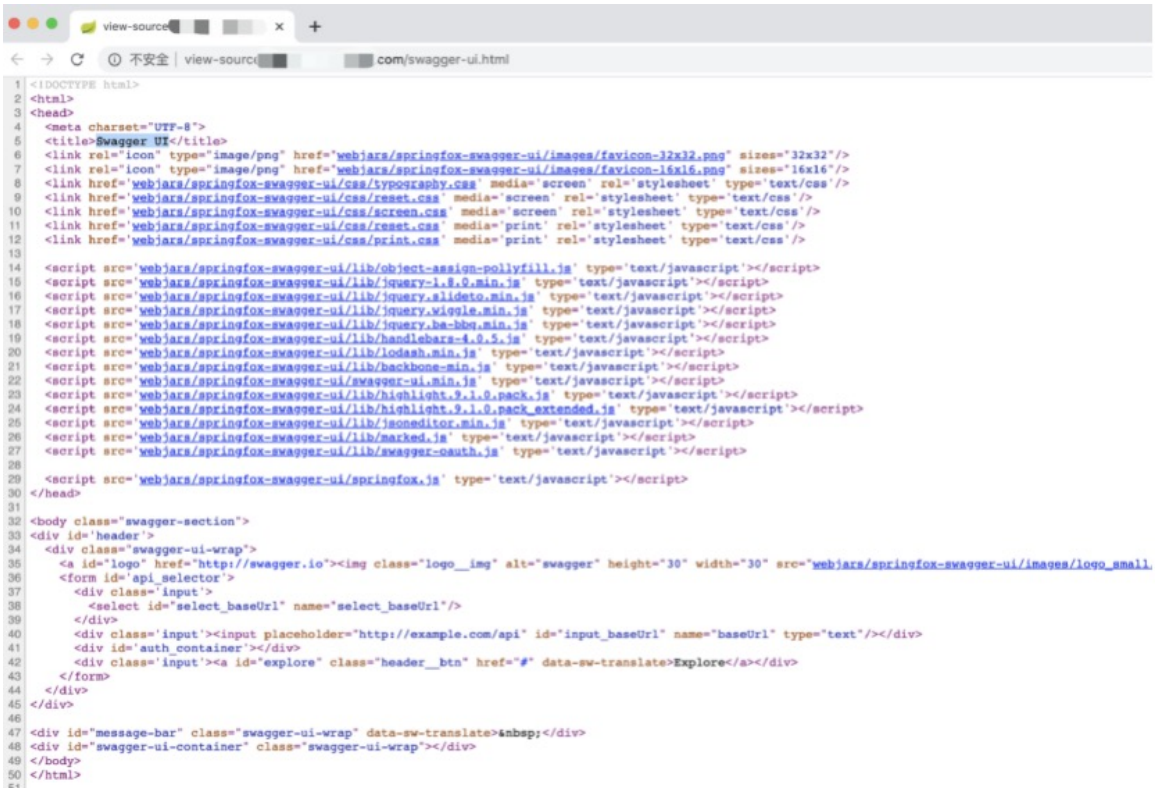
1.通过js查找在网站的config等关键词js文件中查找：



2.通过路径字典爆破以下为搜集到swagger接口常见路径，亲测匹配率很高。

/swagger//api/swagger//swagger/ui//api/swagger/ui//swagger-ui.html//api/swagger-ui.html//user/swagger-ui.html//swagger/ui//api/swagger/ui//libs/swaggerui//api/swagger ui//swagger-resources/configuration/ui//swagger-resources/configuration/security/

3.根据swagger组件特征固定title：Swagger UI



从功能找到切入点

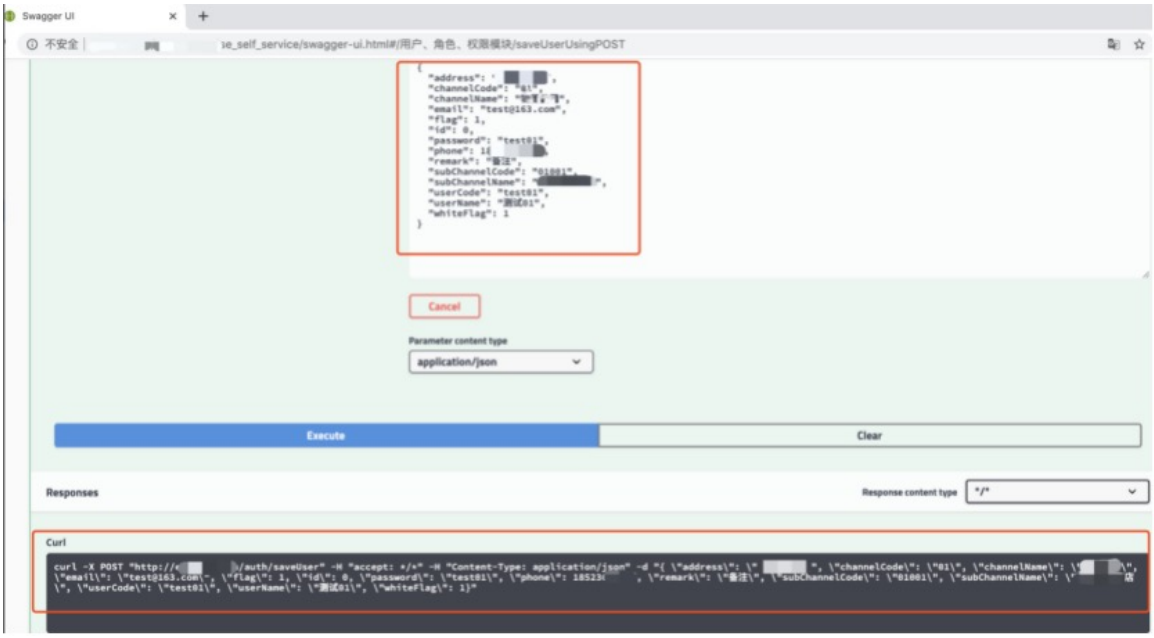
在我们找到Swagger UI页面后，应快速浏览所展示的接口功能，根据功能点由高风险到低风险依次进行安全测试。常见的接口安全测试点如下：接口越权（若接口文档对应的Web应用提供注册功能，可以用低权限用户tok

en尝试水平越权查询修改其他用户信息，或者垂直越权尝试进行管理员操作）接口SQL注入（针对所有查询接口）接口未授权访问（重点针对管理员模块，如对用户的增删改查）任意文件上传（针对上传接口进行测试）测试信息泄露（重点针对用户、订单等信息查询接口，以及一些测试数据等）。

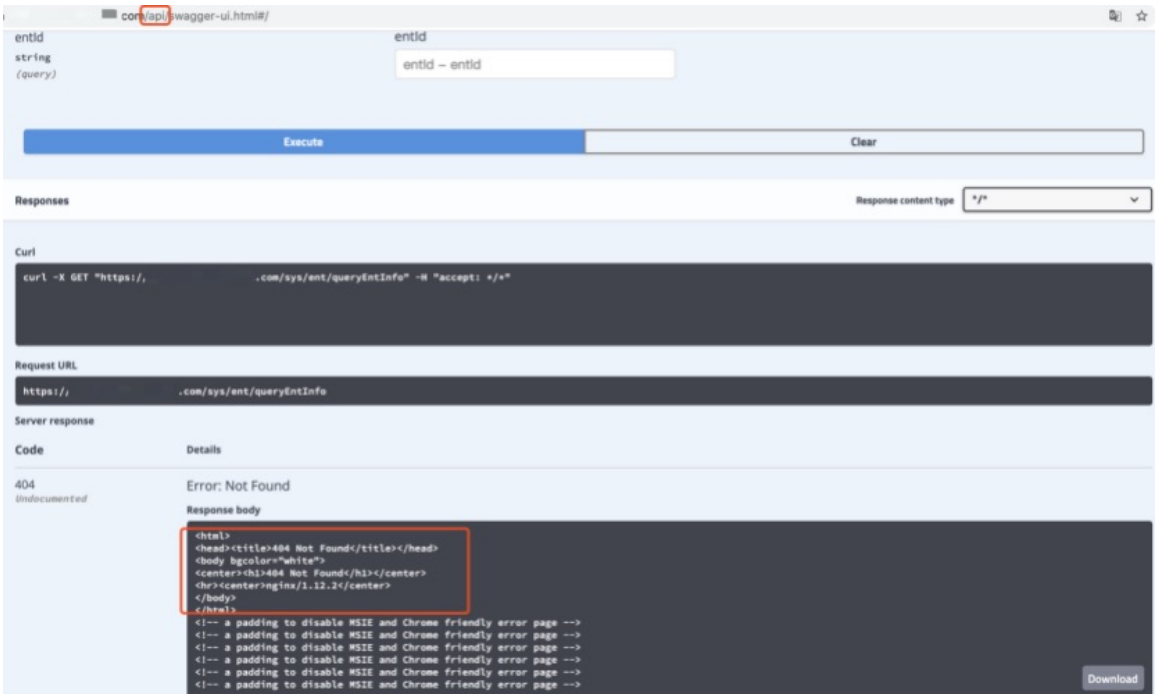
案例：

1.越权

如下所示，在某个泄露的Swagger UI页面中发现管理员添加用户模块以及分配权限模块。



这里有个小坑点，如果swagger页面地址不是直接拼接在域名之后，如图所示，直接请求很容易404。

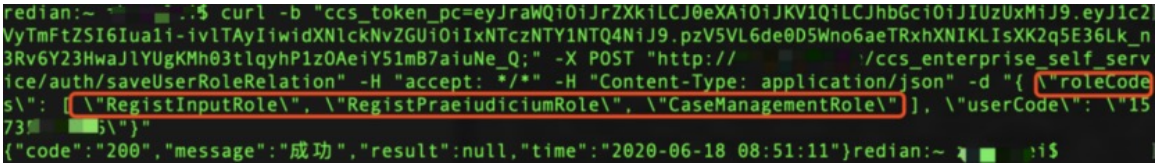


所以一般需要添加swagger-ui.html之前的URI地址（示例为/api），才可以正常进行访问。

而此处管理员添加用户接口也存在这种情况，将curl指令拷贝出来，添加上缺失的URI地址，直接测试访问发现存在身份认证。

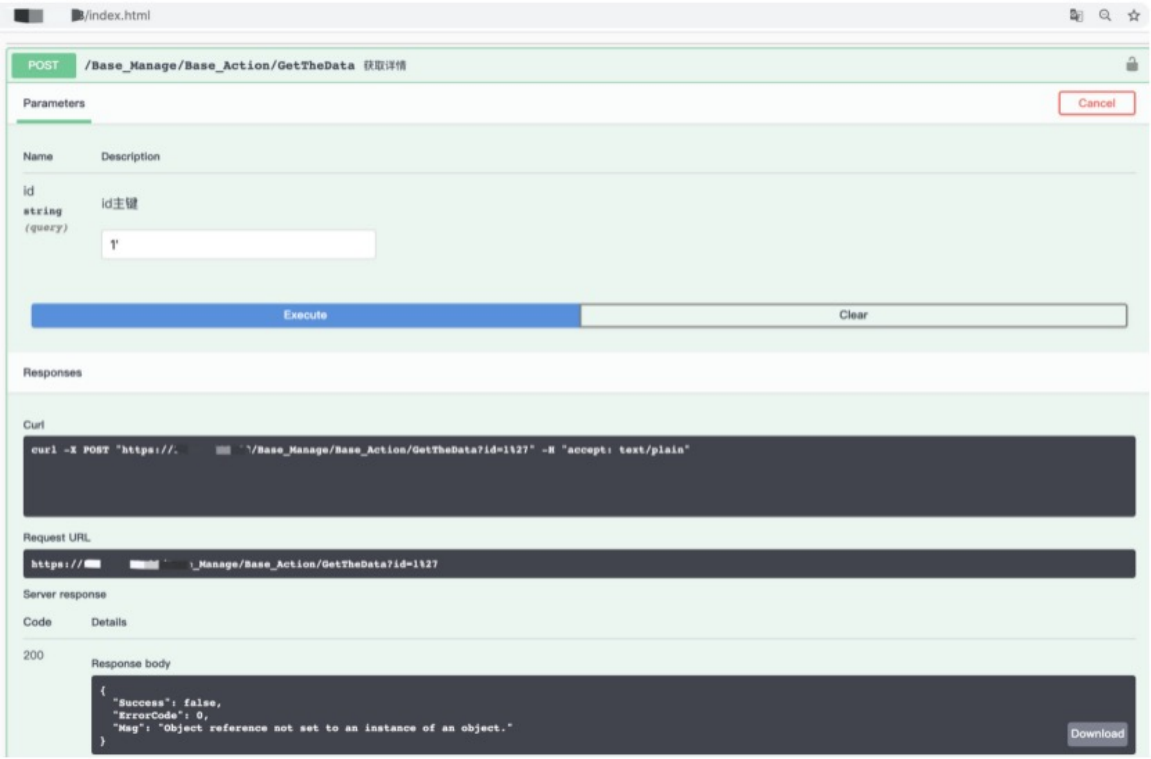
但好在此swggaer对应的web应用提供了注册功能。尝试利用注册的低权限用户cookie去访问，查看是否可以进行垂直越权操作，将登录后的cookie添加在curl请求-b参数中，再去访问，成功垂直越权至管理员身份添加用户。

再利用泄露的权限管理接口，为用户成功添加管理员模块权限。



2.接口SQL注入

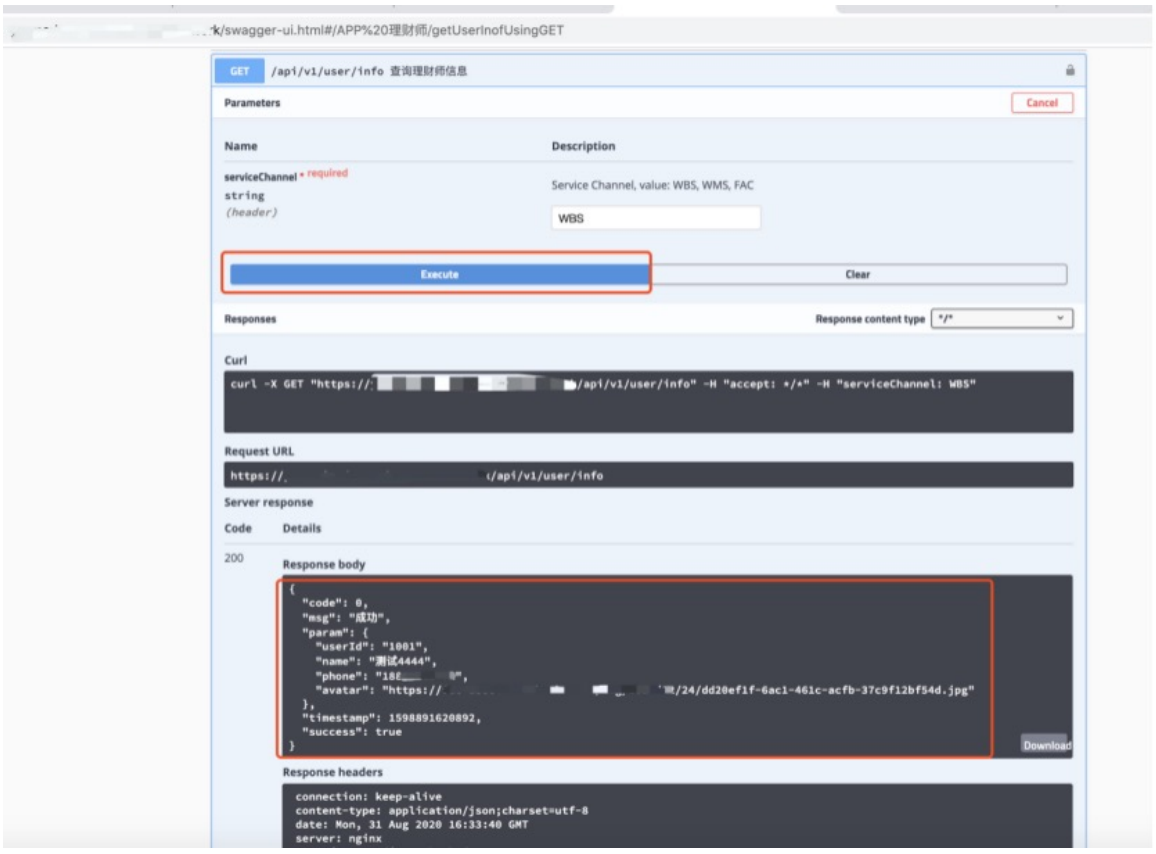
根据查询接口的参数进行正常SQL注入即可，这里不再进行演示。

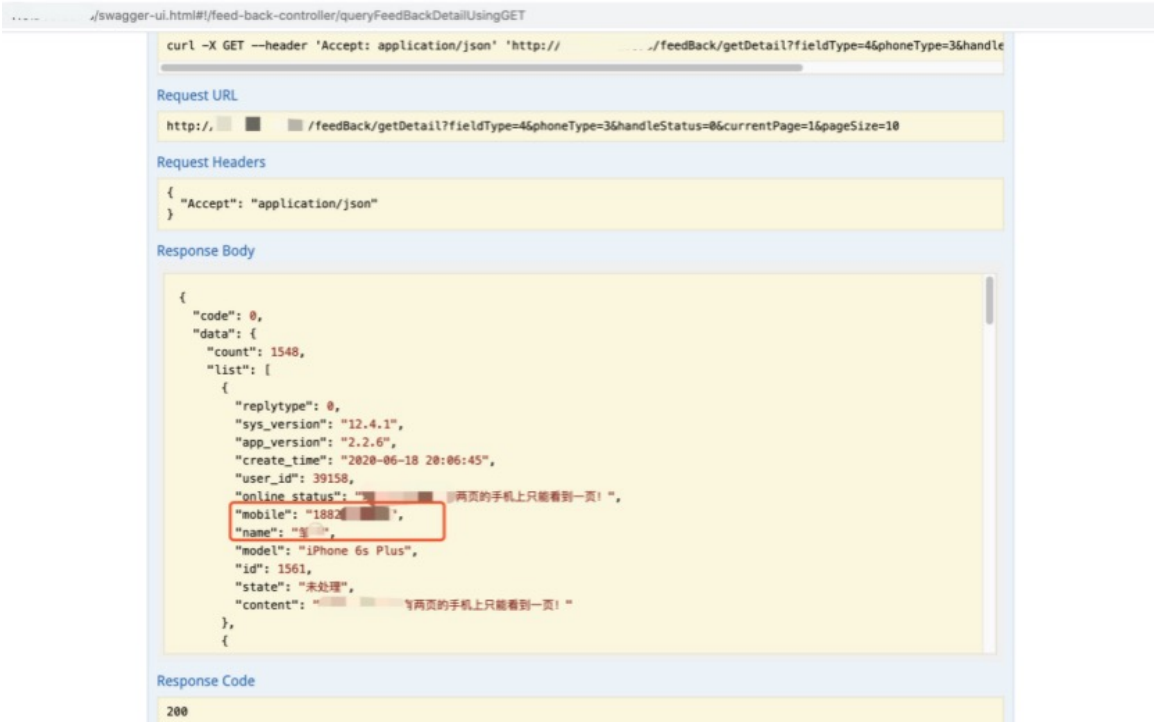


3.未授权访问

很多开发为了对接口测试方便，便取消了身份会话认证，这也导致了未授权访问出现的最为频繁，想象一下几十个功能接口光溜溜的躺在你的面前~

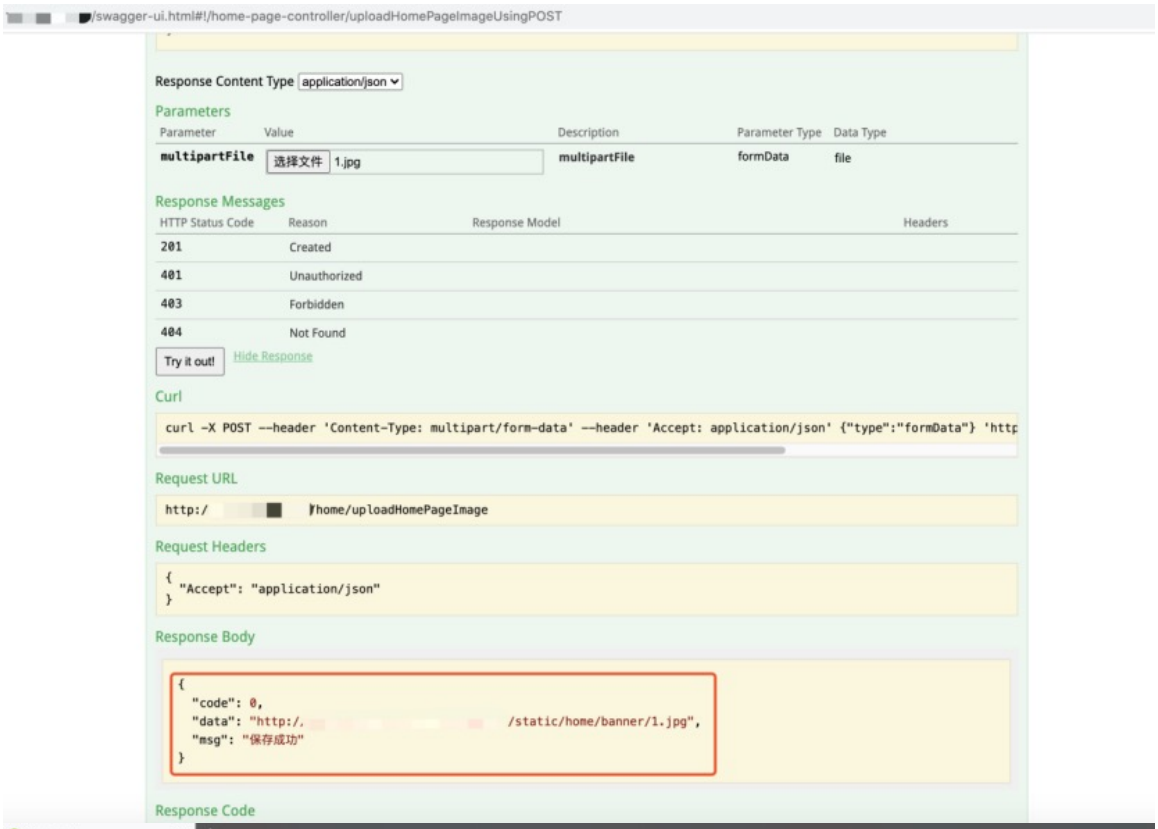
最常见的便是未授权查询接口，只需要小小的点击一下Excute，想要的东西便会出来。

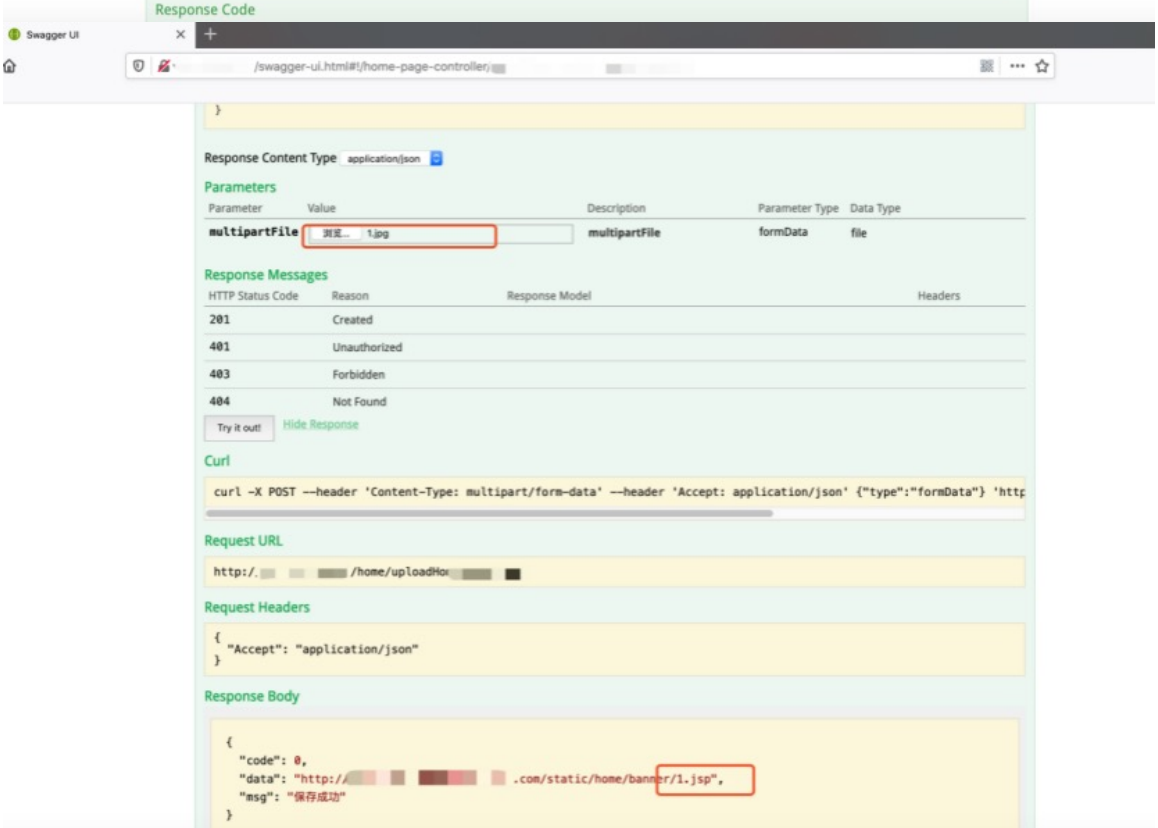


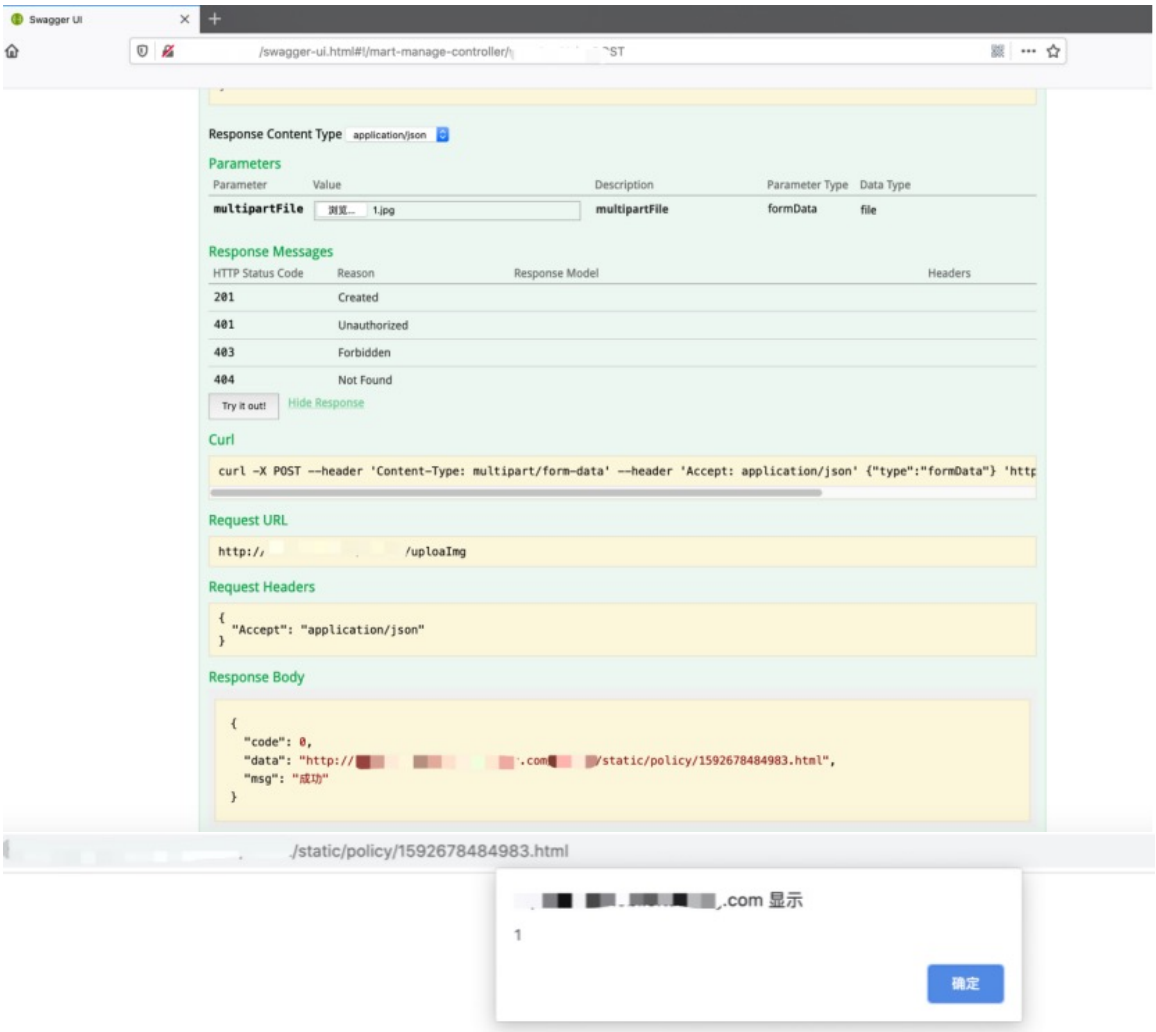


4. 文件上传

文件上传接口大部分均为纯接口形式上传文件，不存在前端校验，可直接上传相应测试脚本文件进行安全测试。

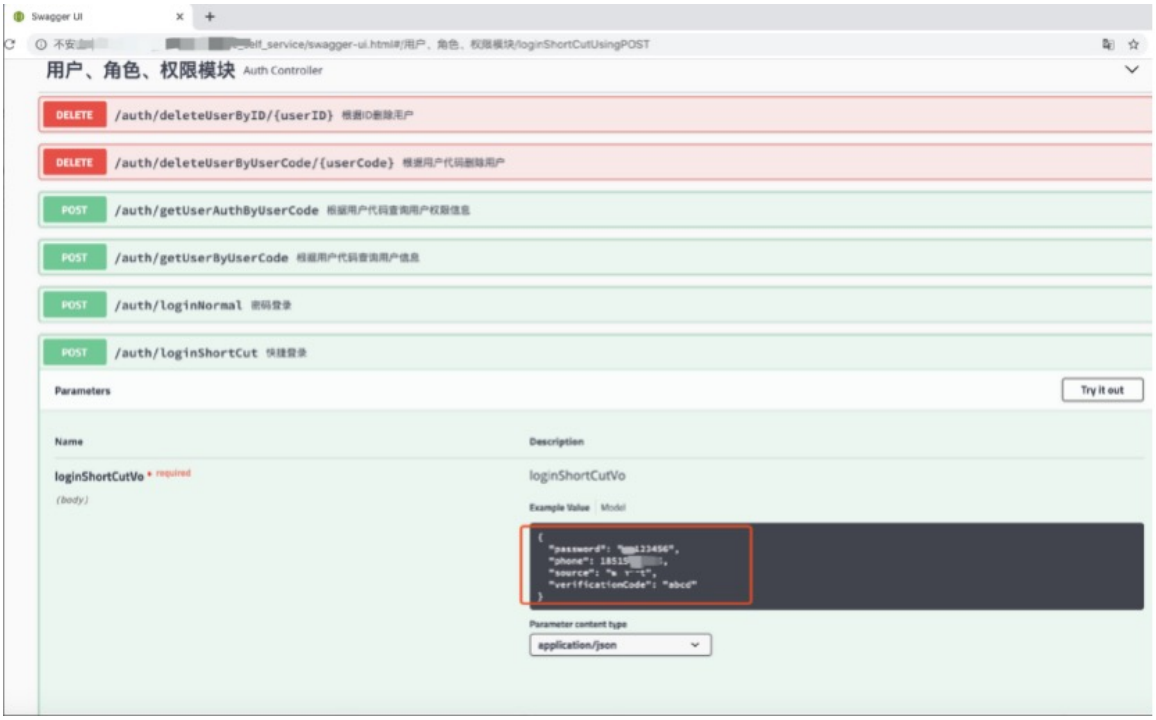

/swagger-ui.html#!/home-page-controller/uploadHomePageImageUsingPOST
Response Content Type application/json
Parameters
Parameter Value Description Parameter Type Data Type
multipartFile 选择文件 1.jpg multipartFile formData file
Response Messages
HTTP Status Code Reason Response Model Headers
201 Created
401 Unauthorized
403 Forbidden
404 Not Found
Try it out! Hide Response
Curl
curl -X POST --header 'Content-Type: multipart/form-data' --header 'Accept: application/json' {"type":"formData"} 'http
Request URL
http:/ /home/uploadHomePageImage
Request Headers
{
"Accept": "application/json"
}
Response Body
{
"code": 0,
"data": "http:/, /static/home/banner/1.jpg",
"msg": "保存成功"
}
Response Code


Swagger UI
/swagger-ui.html#!/home-page-controller/
}
Response Content Type application/json
Parameters
Parameter Value Description Parameter Type Data Type
multipartFile 浏览... 1.jpg multipartFile formData file
Response Messages
HTTP Status Code Reason Response Model Headers
201 Created
401 Unauthorized
403 Forbidden
404 Not Found
Try it out! Hide Response
Curl
curl -X POST --header 'Content-Type: multipart/form-data' --header 'Accept: application/json' {"type":"formData"} 'http
Request URL
http:/. /home/uploadHo
Request Headers
{
"Accept": "application/json"
}
Response Body
{
"code": 0,
"data": "http:/ .com/static/home/banner/1.jsp",
"msg": "保存成功"
}



5.敏感信息泄露

因为swagger ui页面泄露本身就属于最大的敏感信息泄露，而相对于接口中的敏感信息泄露，大部分为在模块中的测试数据泄露，而有些测试账号也会有很大的权限。

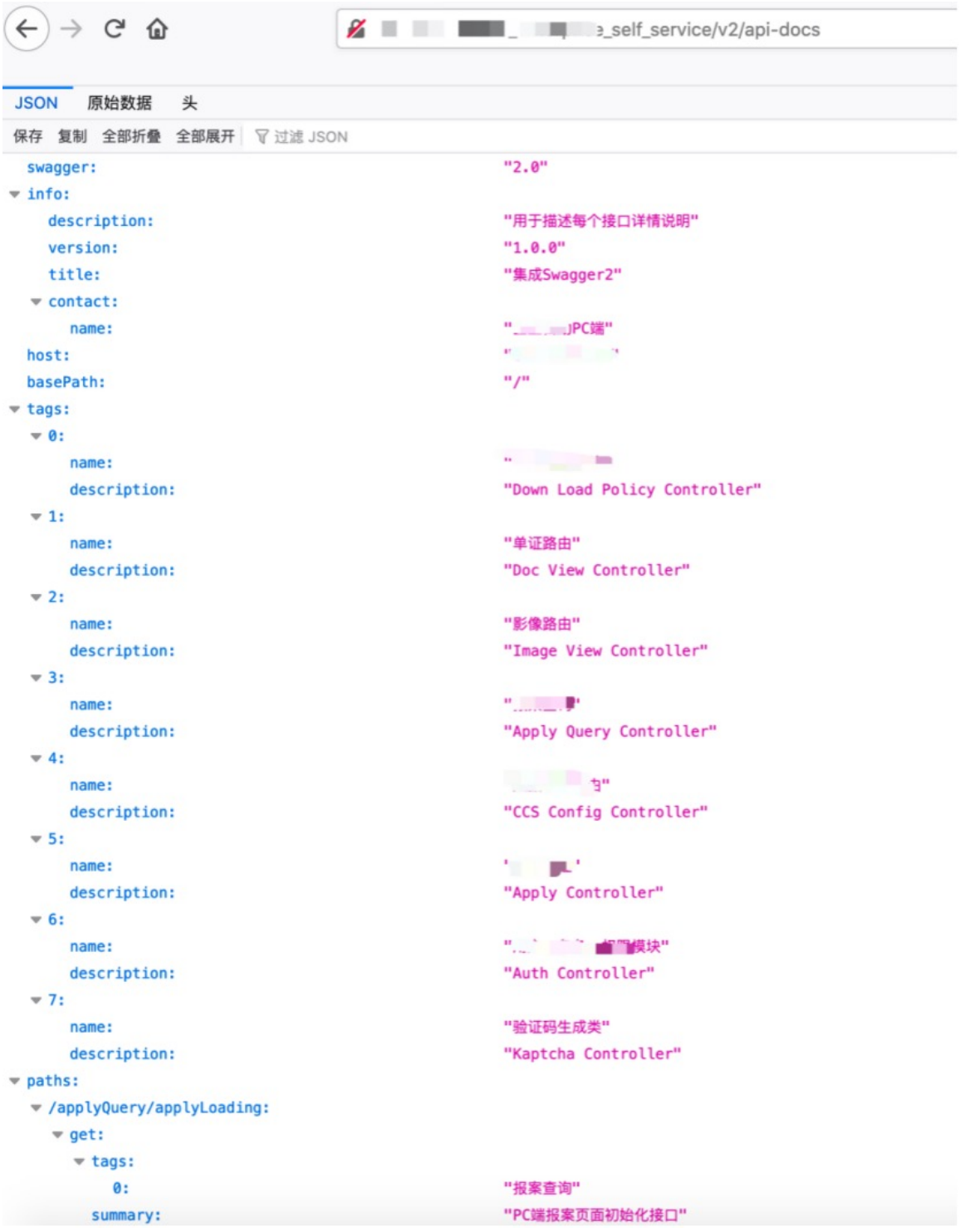


认证限制突破思路

1.Swagger开启了页面访问限制

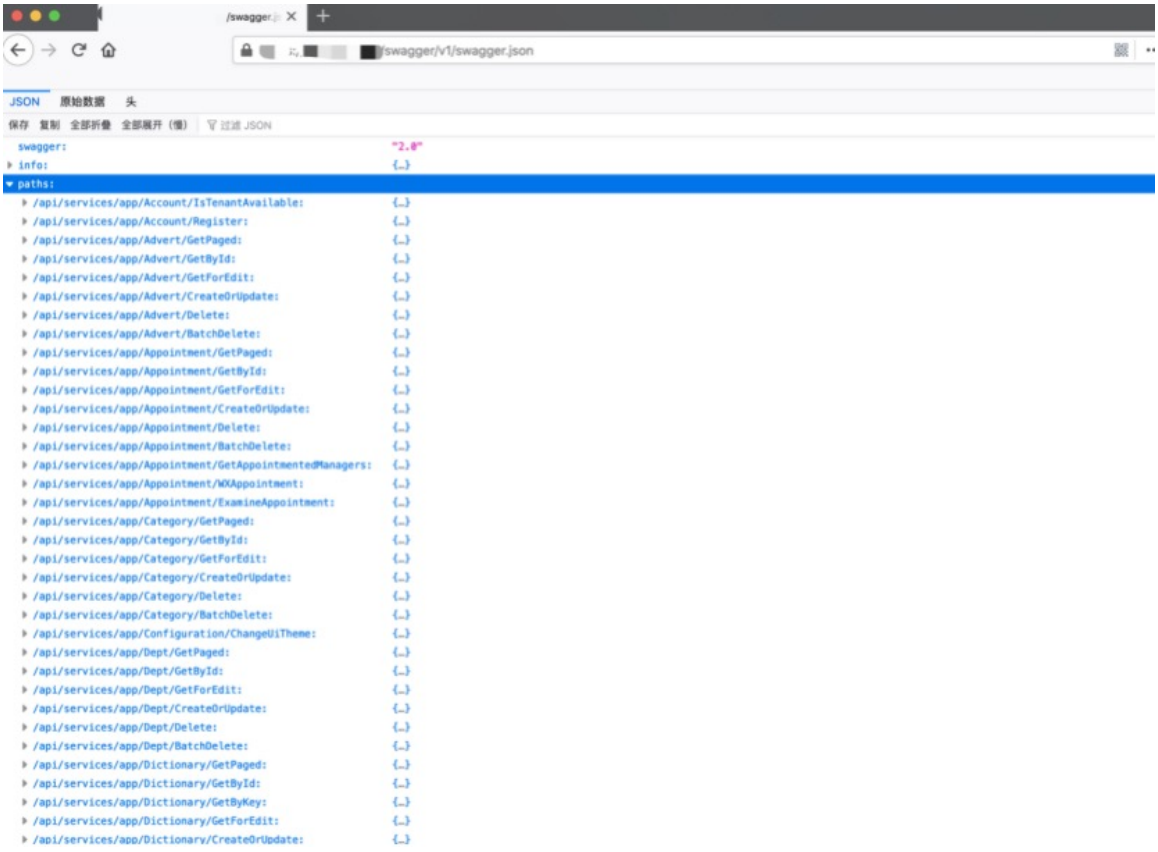
如图所示，某个swagger-ui.html页面添加了登录认证。

除了进行弱口令尝试，还可以直接在swagger-ui.html前一层路径后添加/v1/api-docs即可访问接口(v1代表接口迭代版本，可以尝试v2、v3等可能会有惊喜等着你~)



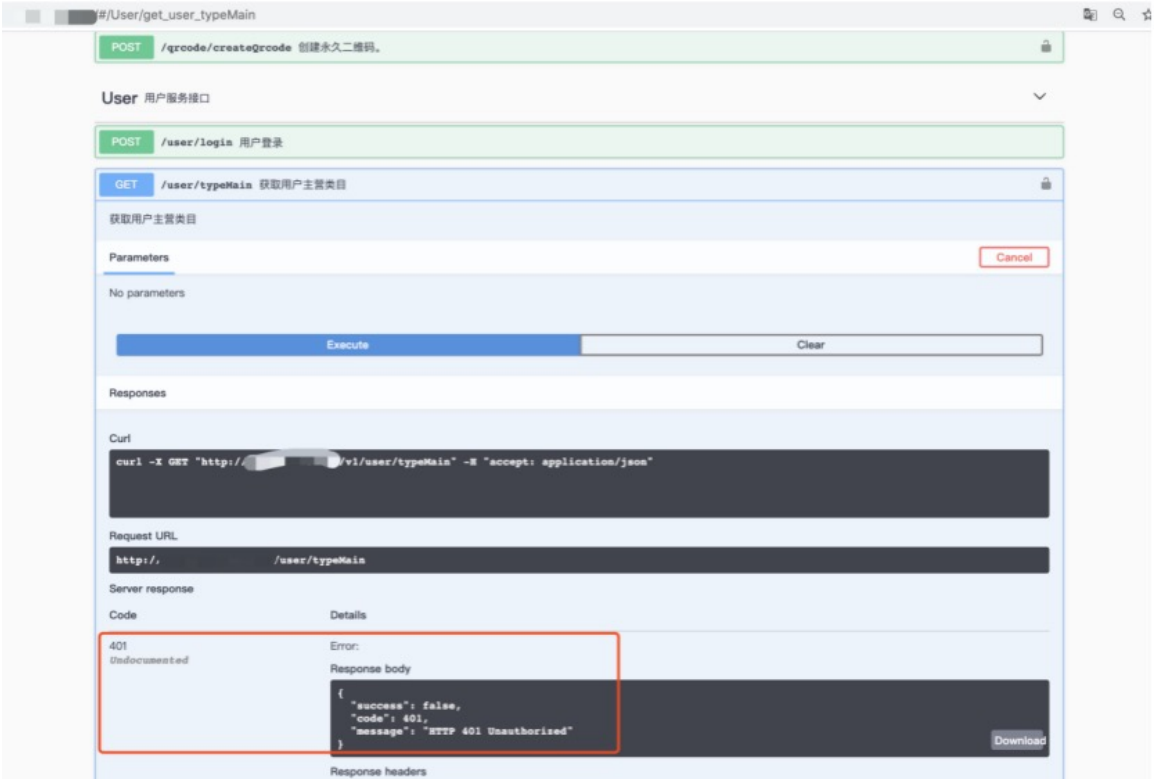
以json格式查看：

/swagger/v1/swagger.json



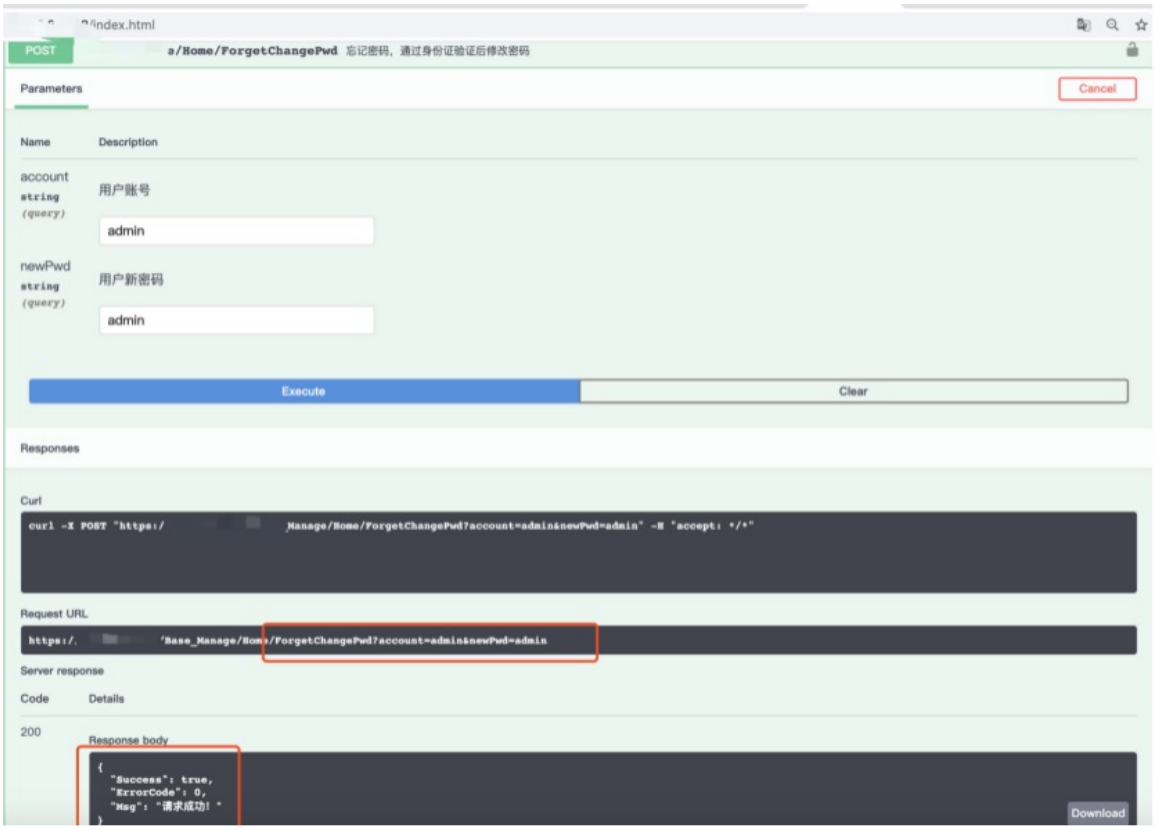
2.Swagger开启了Authorize认证

若Swagger在每个接口请求中开启了严格的Authorize认证，即使我们可以获取所有的接口路径，但因为没有身份会话，导致接口也无法执行成功。

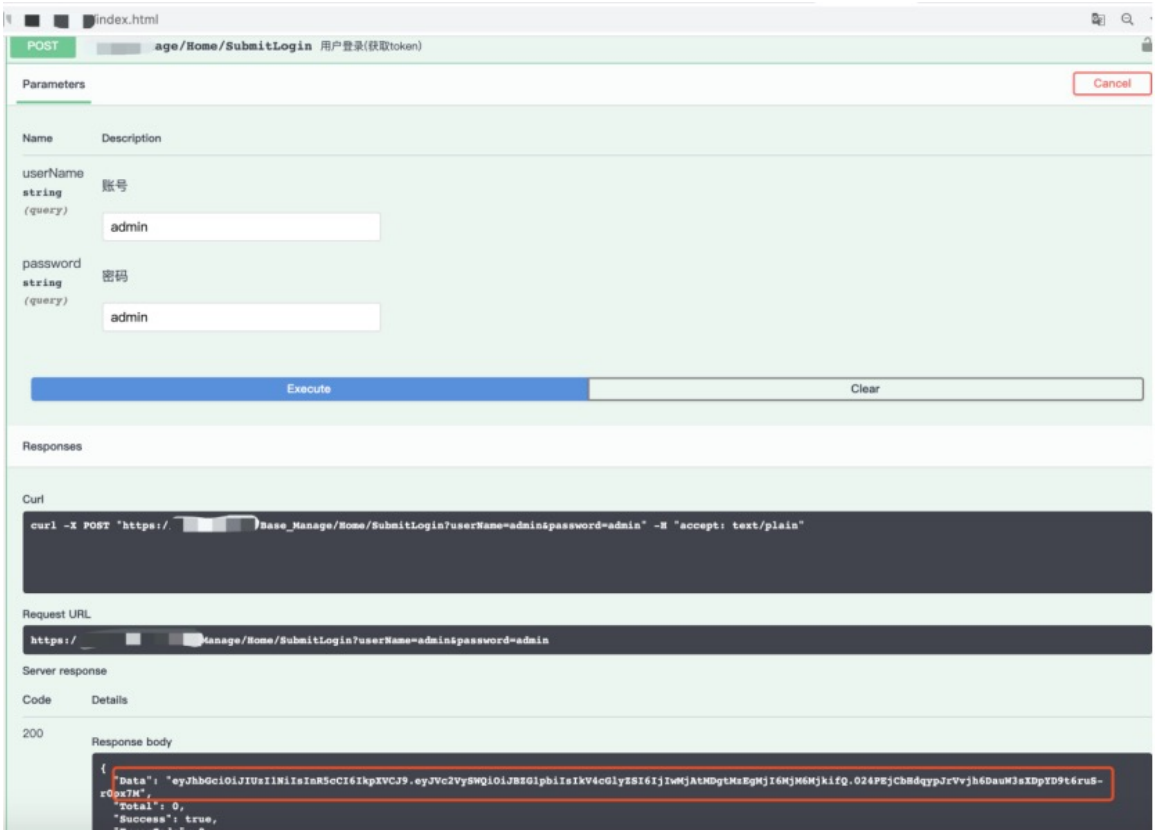


针对于接口开启Authorize认证也不要放弃，我们不要放弃每一个接口，此时更多可以去尝试上传/下载文件、修改密码、登录等模块接口，这些接口也是不需要或者身份认证最容易出现遗漏的地方，当出现一个漏网之鱼时，便可以点溃面，拿下接口权限。

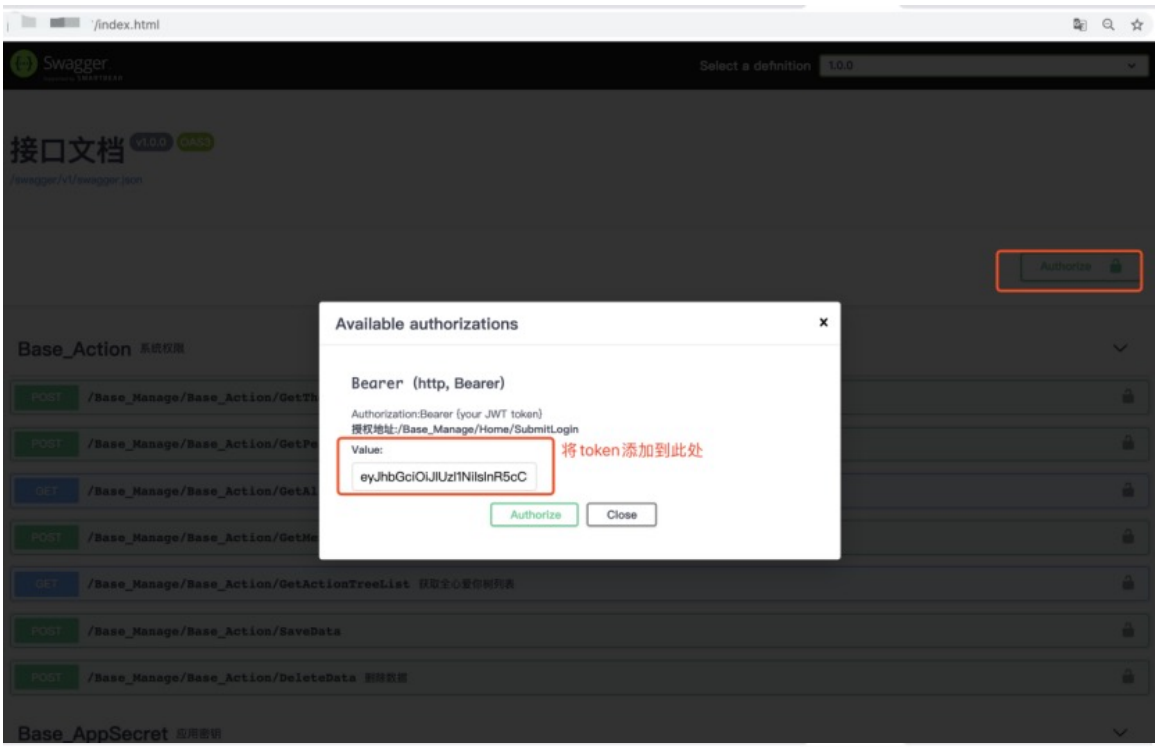
如下在某个swagger页面中，开启了Authorize认证，但通过查找发现一处逻辑缺陷，修改用户账号密码时直接根据用户账号便可修改，可通过此缺陷直接重置管理员密码。

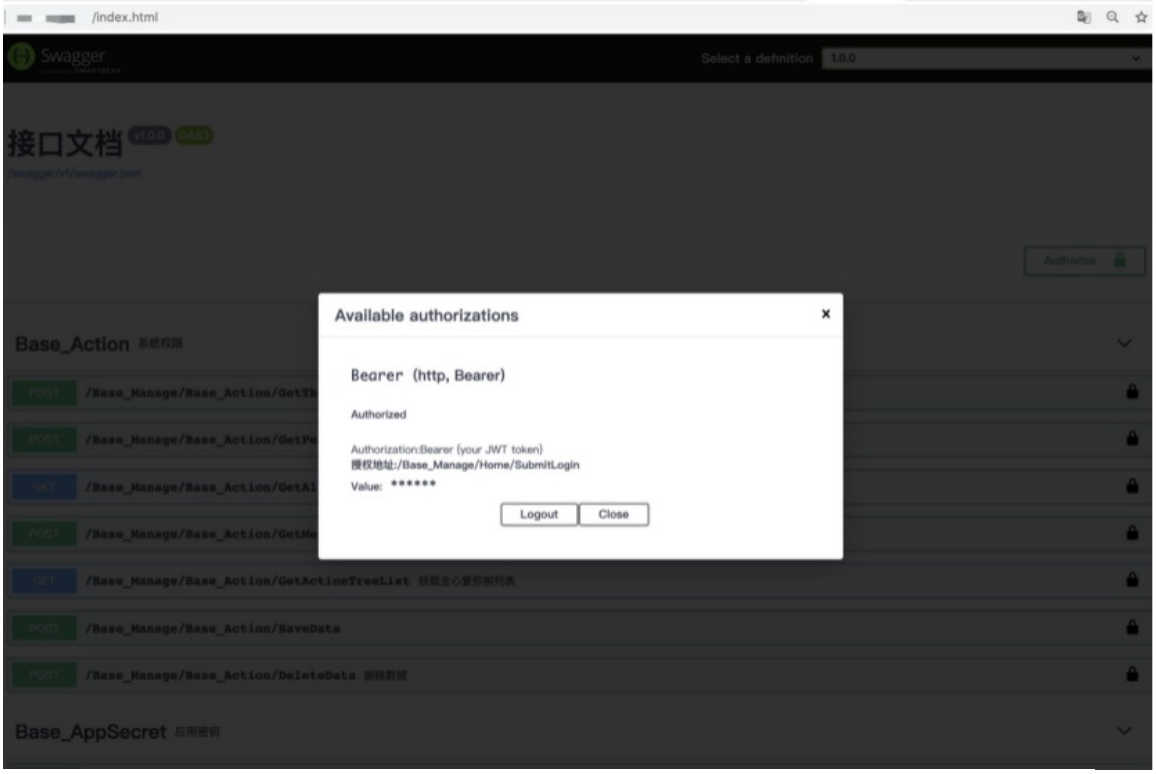


在登录接口尝试登录，发现可以成功登录并获取token。



因为当前swagger接口文档开启了Authorize认证，大部分接口无法直接调试，此时获得了管理员的token，便可对当前swagger文档中的所有接口进行操作。


/index.html
Swagger
Select a definition
1.0.0
接口文档
Authorize
Base_Action
Available authorizations
Bearer (http, Bearer)
Authorization:Bearer (your JWT token)
授权地址:/Base_Manage/Home/SubmitLogin
Value:
eyJhbGciOiJIUzI1NiIsInR5cC
将token添加到此处
Authorize
Close
/Base_Manage/Base_Action/GetActionTreeList
/Base_Manage/Base_Action/SaveData
/Base_Manage/Base_Action/DeleteData
Base_AppSecret


/index.html
Swagger
Select a definition
1.0.0
接口文档
Authorize
Base_Action
Available authorizations
Bearer (http, Bearer)
Authorized
Authorization:Bearer (your JWT token)
授权地址:/Base_Manage/Home/SubmitLogin
Value: ******
Logout
Close
/Base_Manage/Base_Action/GetActionTreeList
/Base_Manage/Base_Action/SaveData
/Base_Manage/Base_Action/DeleteData
Base_AppSecret

总结

不管是挖掘SRC还是日常的渗透测试中，发掘泄露的接口文档可以辅助我们更好的进行漏洞挖掘，而Swagger UI页面中一般会包含大量的测试接口，在进行上述漏洞总结点的安全测试时，还可以尝试组合漏洞的利用，只要心（dan）够（zi）细（da），从接口测试到getshell、内网漫游也未尝不可。

精选留言

用户设置不下载评论