# 某APP 渗透测试

比较详细,大牛轻喷。

0x001.

下载app 安装了之后，挂个代理。分析之后，这个app是 浏览器+ html 模式。说白了wap 网站，采用全站api，所以的请求貌似都通过用户访问的服务器中转请求api服务器，拿到数据再返回给中转服务器，中转服务器再返回给客户端。其中采取了token校验数据是否来源于中转服务器，比较无语。。。。

1. 上个bugscan 开始扫。
2. 然后我就去收集一些whois 信息什么的。

运气比较好。在主站扫到个源码包泄露，但是并没有发现什么配置文件。

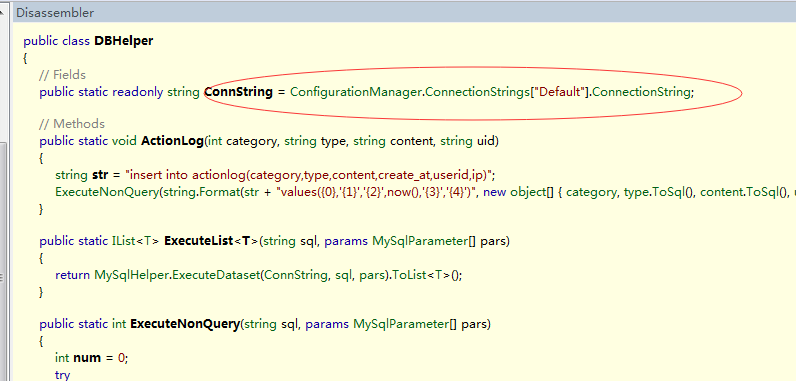
看到bin目录中存在dll文件，接下来请出了 reflector 进行反编译。

初步最好通过名字找到一些数据库的配置信息（有可能直接写在dll里面）

看到一个Common 文件，

发现4个比较有用的信息点：

第一个：是个dbhelper类

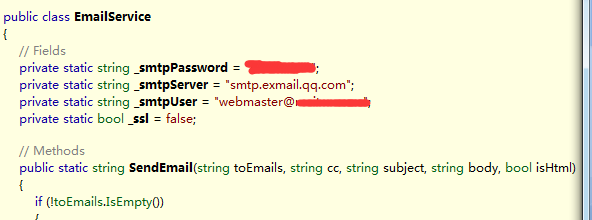


可以看到是从配置文件读取过来的。但是泄露的源码包里面并没有配置文件。

这条路走不通

第二个：

发现一个emailservice 类。 通常邮件服务器 能发现很多有用的信息。



是个腾讯的企业邮箱。

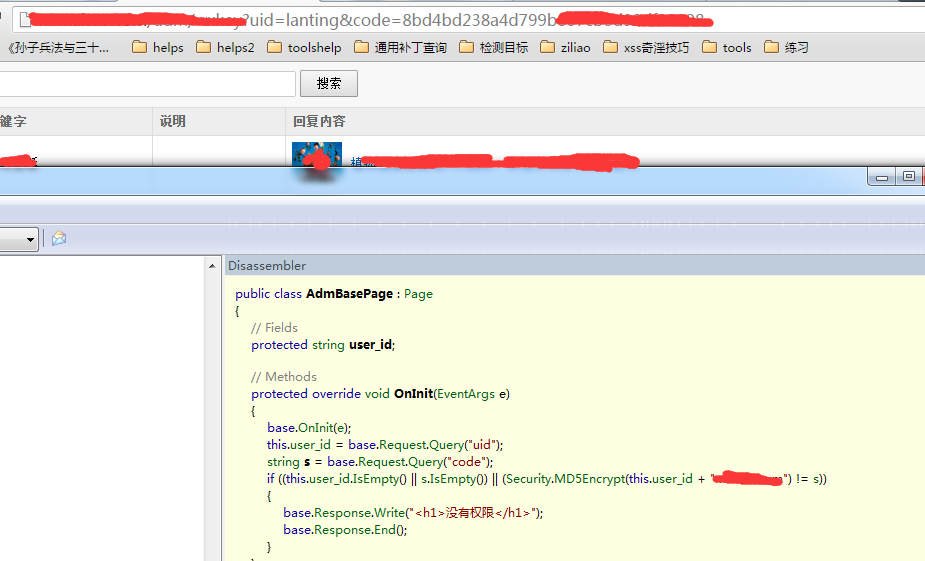
但是里面也没有发现太多有用的信息

第四个信息：

后台某个地方token加密过于简单

采用uid+“string” md5加密形式

绕过了后台，后台有一些搜索框框什么的，但是发现这些功能都用不了，



0x002.采取新思路，既然是后台某个模块的一个功能，不如打入xss，简单拿下管理cookies并且后台管理路径，晚上6点左右打的，吃完饭，收到信息，



这个后台是个独立的域名，这个时候我们获得一个信息，就是说我们绕过的这个页面 可以理解为一个“组件“ ，也许同时嵌套在app 管理员 的页面， 和pc 端的后台 功能页面。

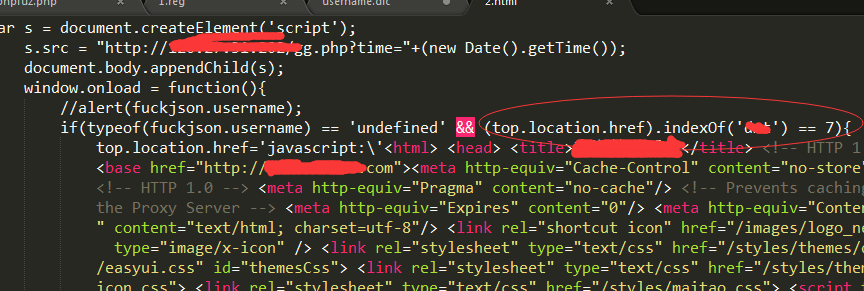
进入后台系统，无语的发现了httponly，而且不能暴力破解，错五次就锁定账户，都不是锁定ip。又让我无语了一次。。。。



看样子应该是DWZ 后台的框架。

能简单就简单，于是乎可以尝试 钓鱼了。

注意2个点，第一个：这个是不同域，前面已经提到了这个功能是个“组件“ 所以我们要判断是否来自于后台那个域名过来的，在图片画圈处。然后设置一个base标签。因为应用的资源来自于另外的域名。



最后在自己服务器，用php写个接收代码 利用对面暴露的短信接口。打到账户密码就发送短信到我手机上面。