打击违法犯罪

前言：

在渗透的时候经常能遇到任意文件上传，但大部分情况下都被上传到了阿里云的 oss 存储服务器，即使有任意文件上传也无法解析执行恶意木马，今天刚好遇到这样的情况，web 硬怼根本怼不下，发现目标的 app 存放在阿里云的 oss 服务器上，就一心寻找他们的 osskey 最终通过 oss 获取到目标权限。

正文：

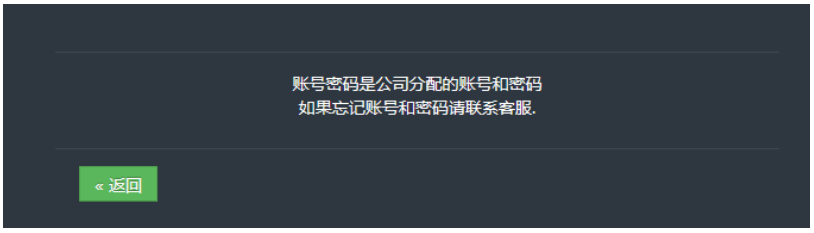
春节前，朋友突然丢了一个网站给我，让我帮忙看看，说是杀猪盘，核心人员已被羁押，但是证据链还差一点，需要进入他们的网站后台提取一些流水信息，各种授权均能开。

反正快放假了周末也没啥事干就帮个忙。首先明确目标，需求是什么？

1. 最低要求进入后台
2. 如果有数据库权限更好
3. 能拿服务器权限最好

目标明确，就开始踩点咯。网站服务器是阿里云的，IIS7.5、ASP.NET 首页打开空白，随手 admin 就是后台，验证码不刷新不过期，可以爆破。但是字典都爆破完了，还是没爆破进去....

扫目录得到一个普通用户登录地址，无验证码，继续爆破，还是没爆破进去....（后来发现大部分用户使用中文用户名，我没有加载我的中文字典....失算）



继续扫目录只有一个公告栏目，尝试注入也没有成功，从公告中也没有获取到有用的信息，于是转过来头来，看每一个 js 文件，希望能够找到一些接口或其他有用的信息，最终在一

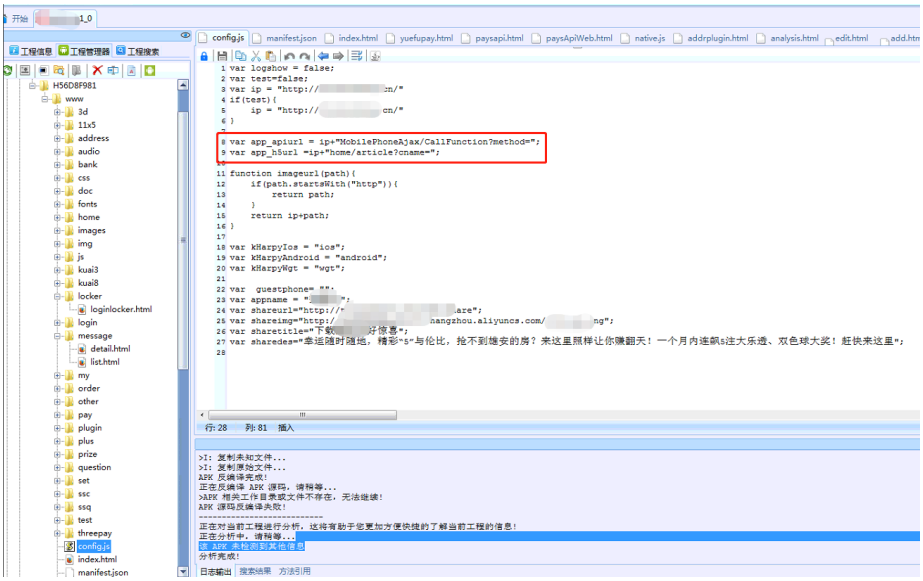
个 JS 中看到了 app 的下载地址。发现其 app 放在阿里云的存储服务器中，看到这里真是又激动又担心。

```
function iosStatistic() {

    location.href = "itms-services://?action=download-manifest&url=https://[redacted]plist";
}

function androidStatistic() {
    location.href = 'https://[redacted]-cn-hangzhou.aliyuncs.com/apk/[redacted]apk';
}

function iosGetOverlay() {
    if (!is_weixin()) {
        var type = "android";
        var stamp = "Ac2[redacted]NeJ1Y";
        if (!is_safari()) {
            alert("亲,请在苹果自带浏览器(safari)下载");
        }
        if (type != "null" && stamp != "null") {
            iosStatistic();
        } else {

            location.href = "itms-services://?action=download-manifest&url=https://[redacted]plist";
        }
        return;
    }
    var docHeight = $(document).height();
    $("#overlay").css({
        'display': 'block',
        'opacity': 0.7,
        'position': 'absolute',
```

激动是因为他们使用了阿里云的 oss 存储服务器存放东西，如果程序员没有什么安全意识的话，极有可能会泄露阿里云 oss 的 key，只要找到这个 key 就可以直接重置服务器密码，完成任务。

担心是因为如果没有找到这个 key，即使通过其他渠道进入后台，后台存在任意文件上传也有很大的可能上传到 oss 服务器上，导致无法解析。

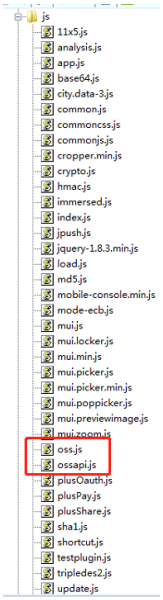
Web 已经没有啥思路可以搞的了，端口只开了 80，1433，3389 感觉弱口令的可能性比较低，就没有进行爆破，而且阿里云还有保护机制，mssql 爆破不了几次，至此陷入僵局。不过还好刚刚发现了 app 的下载地址，于是下载 app 本机反编译，寻找其他有用信息。发现此 app 的开发人员直接简单的对 app 进行打包了而已，并没有加密加壳，很轻松的获取到了反编译后的部分源码，并从中获取到了一些 html 和 js 源码，在 config.js 中发现了两个接口，其中一个存在注入，但是没有深入去看。



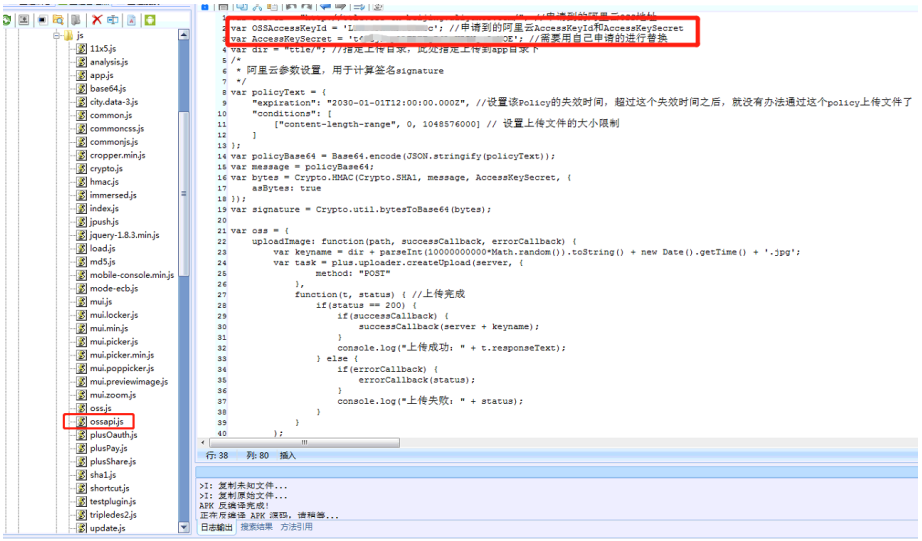
Sqlmap 识别出是 mssql 延时注入，但是获取不到信息，我估计是阿里云拦截了，先放着，如果最后没东西搞了就搞这个，大不了开个代理池。

```
GET parameter 'cname' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 61 HTTP(s) requests:
---
Parameter: cname (GET)
    Type: stacked queries
    Title: Microsoft SQL Server/Sybase stacked queries (comment)
    Payload: cname=admin';WAITFOR DELAY '0:0:5'--
---
[22:54:44] [INFO] testing Microsoft SQL Server
[22:54:44] [WARNING] it is very important to not stress the network connection during usage of time-based payl
oads to prevent potential disruptions
[22:54:44] [WARNING] the back-end DBMS is not Microsoft SQL Server
[22:54:44] [CRITICAL] sqlmap was not able to fingerprint the back-end database management system
[22:54:44] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 41 times
```

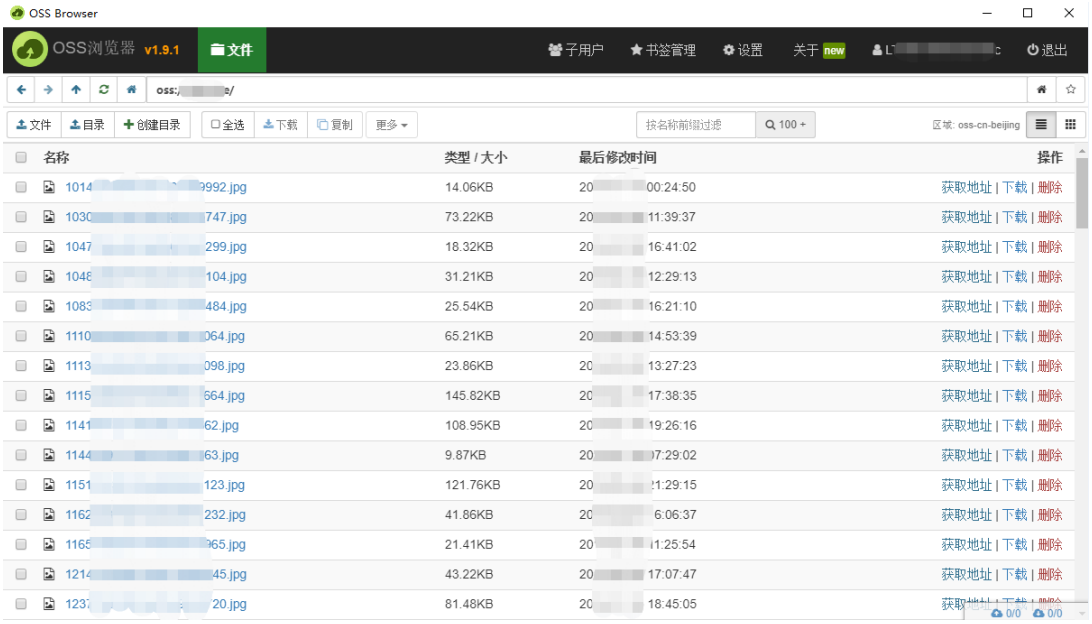
我对 js 情有独钟，直接点开 js 文件发现就发现 oss.js 和 ossapi.js 两个 js 文件



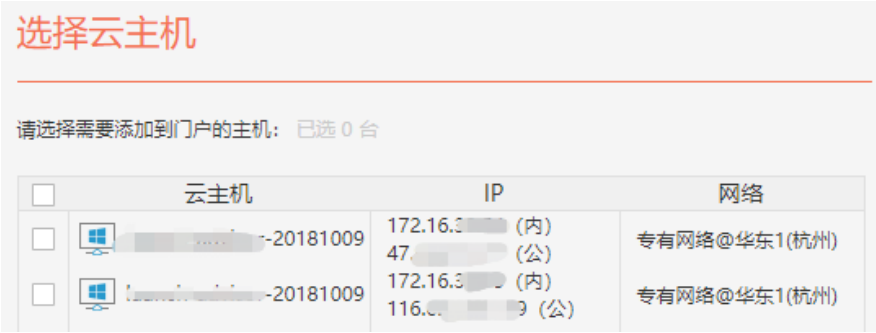
当打开 ossapi.js 后，证实了我的想法，即使有任意文件上传也只是上传到阿里云的 oss 存储服务器，不会解析的，还好他们的程序员足够沙雕，把 oss 的 key 写在 js 里面。



拿着 oss 的 key 使用 oss 浏览器链接上去后并没有发现如数据库备份、源码备份等有用信息，只是一些图片...

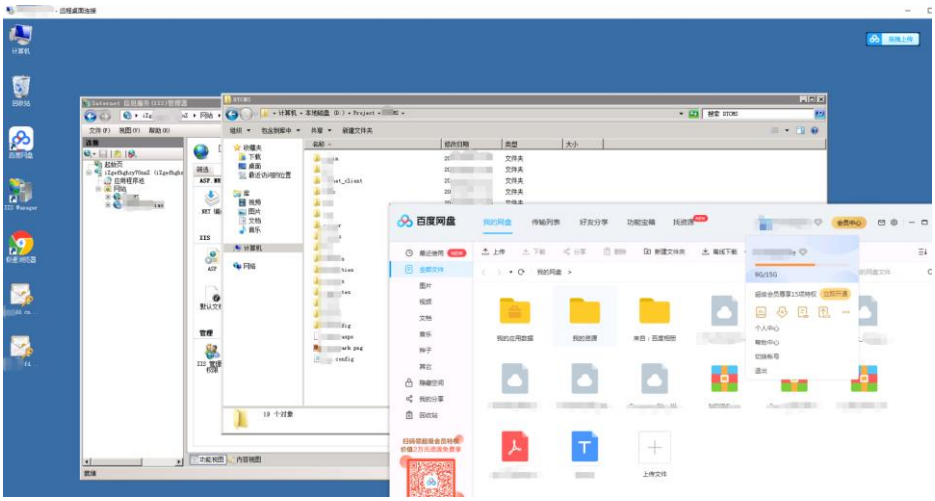


于是只能拿着他们的 key 使用第三方平台去链接他们的服务器，重置服务器密码了。



额。。。 只有两台服务器而且目标 IP 还不在里面，有点怀疑是否日偏了，有点小尴尬。后来通过 app 和 oss 里面的内容确认没有日偏，而这两台服务器 47 的 IP 也跑着和目标一样的程序，只不过名字不一样而已，最终确定是同一个开发团队管理的服务器，然后直接把 ip47 的服务器密码重置，果断登陆 3389（不到走投无路十分不建议这样干，简直找死。）

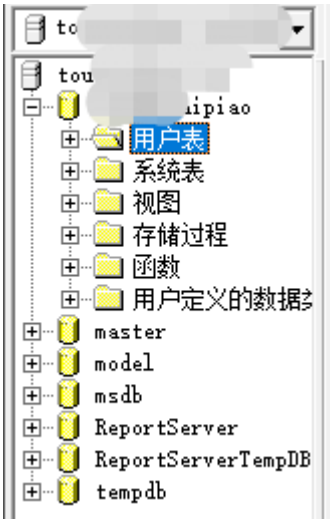
登陆以后快速收集一波信息、打包源码、留后门、种马、断开连接。（下面开启走狗屎运模式）



这里收集到了一份与目标相同的源码，还有一个公网数据库连接账号密码，还有管理员的百度盘账号，可以确定是一伙人。

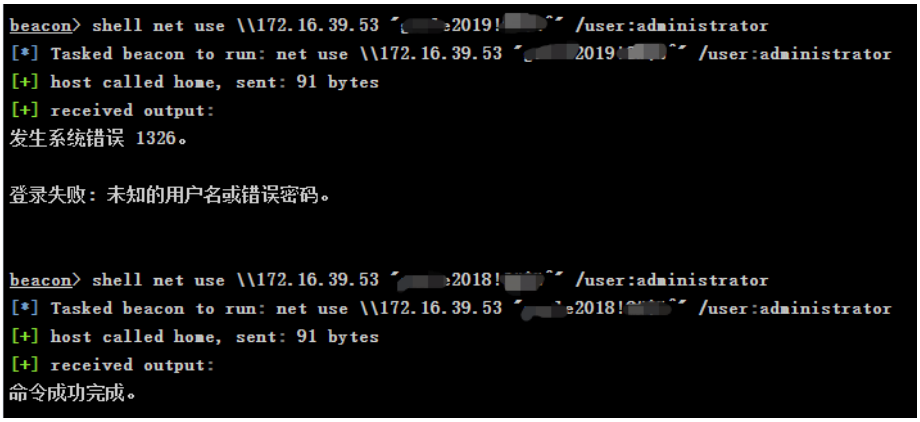
47.96[redacted]uid=[redacted]e;pwd=[redacted]e2018![redacted]

数据库里有价值的东西并不多，像是个废弃的。唯一最有价值的东西“服务器密码”被我重置了(一_一)，使用抓到的数据库密码尝试登陆目标数据库，但提示错误密码错误，看密码好像存在规则于是把密码中的 2018 换成了 2019，成功登录目标数据库.....mdzz

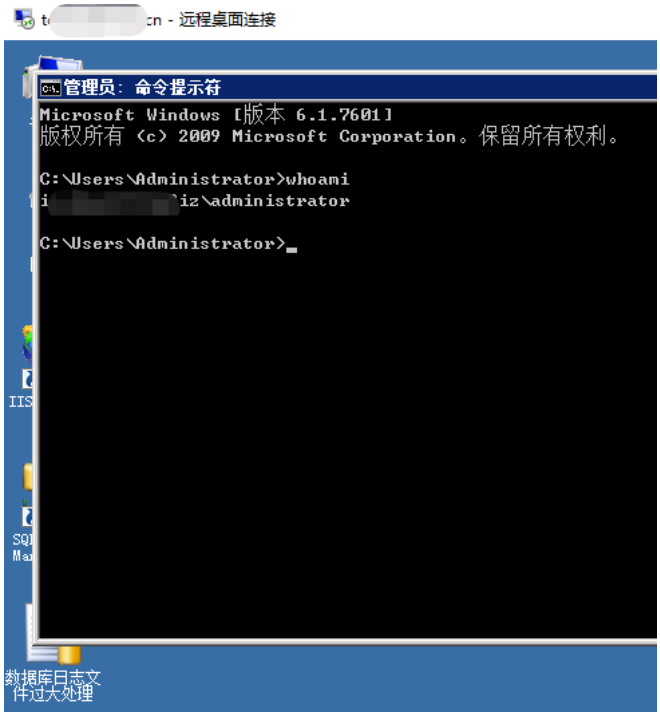


查出账号密码，解密进入后台，初级任务完成

后台可用功能少的可怜，唯一的上传点是 kineditor 编辑器，想要通过后台拿 shell 有点困难...但最终还是获取到了目标的服务器权限，本来想通过数据库执行命令拿服务器权限的，但因为管理员过于沙雕...数据库密码和服务器密码只是年份不一样而已...



导致猜到服务器密码直接连接 3389



任务完成。

R3start

2019 年 1 月 20 日 01:26:26