某授权项目渗透测试

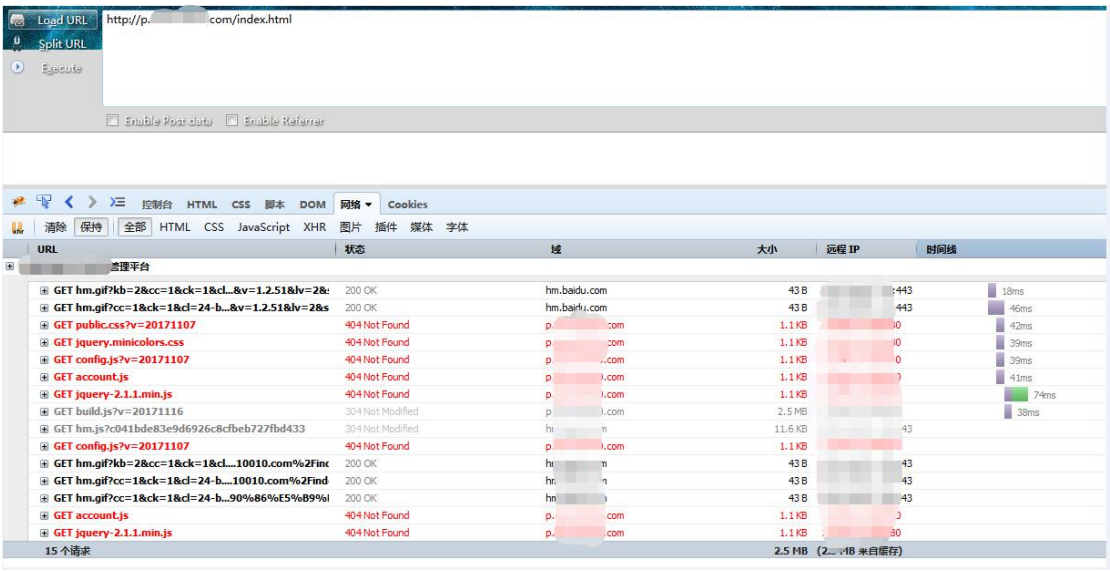
测试目标站点: https://xxx.*****.com/#/

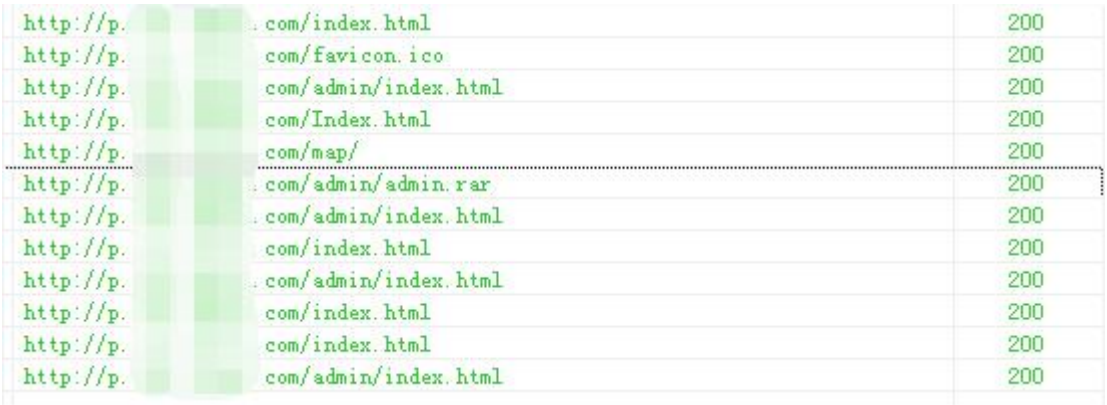
随便打开一个帖子 https://xxx.*****.com/#/***/***/**********22

发现图片路径存储在三级域名上

http://p.xxx.*****.com/upload/***/****/**4615.jpg

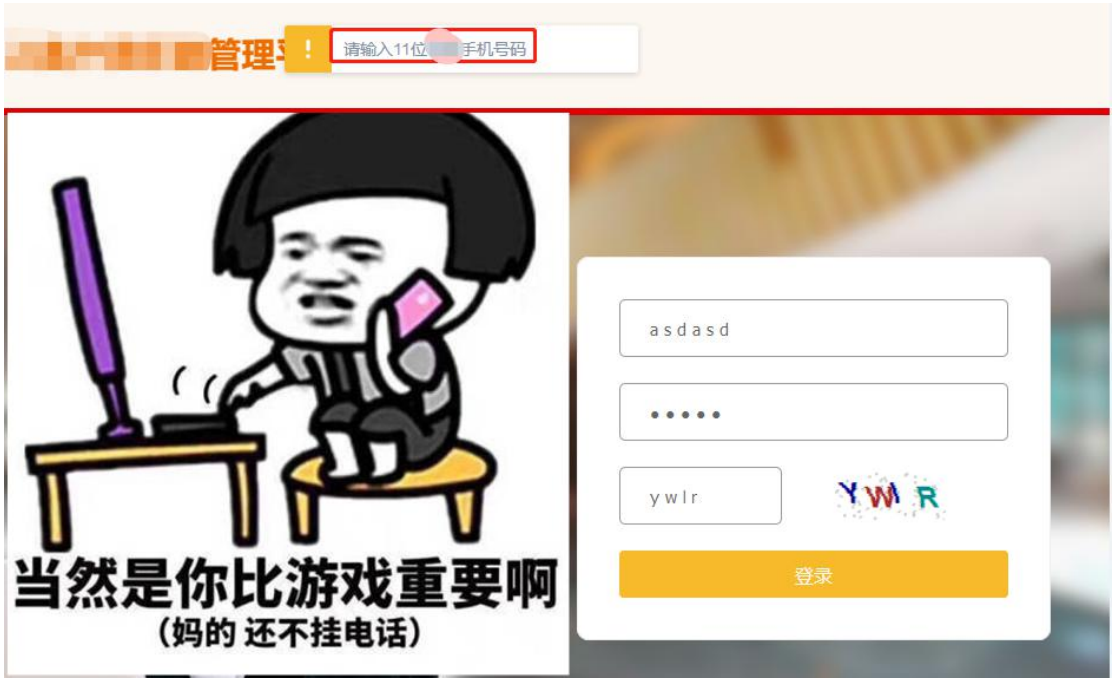
删掉 URI 直接访问根路径发现 title 是“某某管理平台”但是由于丢失 js 文件导致页面无法加载，故对目录进行扫描





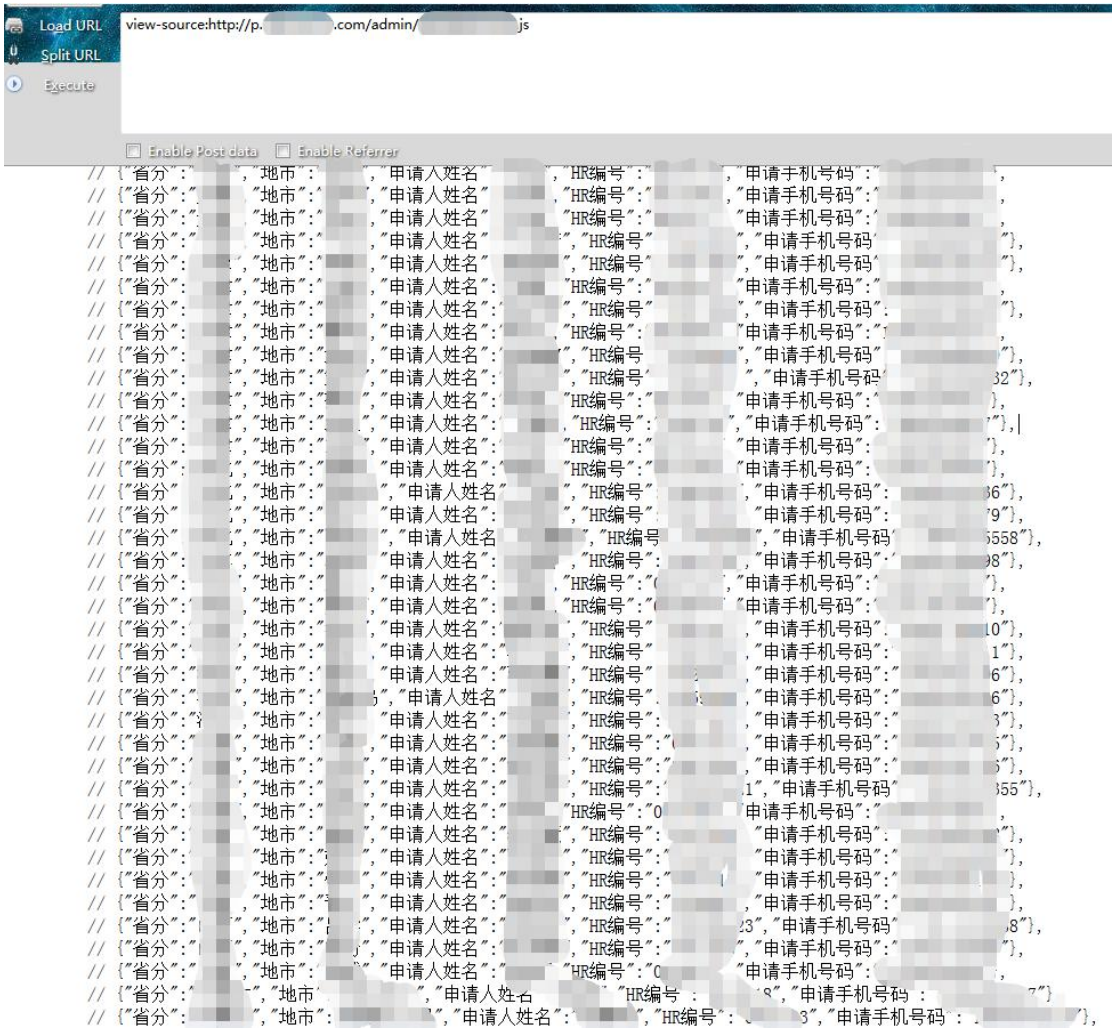
常规扫描后得到后台地址和一份后台 js 压缩包

尝试登陆发现账号为 11 位的某某手机号码，看到账号是 11 位手机号码，我直接放弃了识别验证码爆破的想法，动作太大，可能性太小。



只能通过别的思路进行下一步渗透，这种前端使用 webpack 打包的站点，每个功能基本都是以接口的形式调用，而且很多权限都控制不严，搞不好能找到后台接口，直接操作一些功能，于是先通过 F2 查看加载的所有 js 代码，果不其然在某处 JS 代码中发现泄露了 329 多个可登录的账号

http://p.xxx.*****.com/admin/****/****/***.js



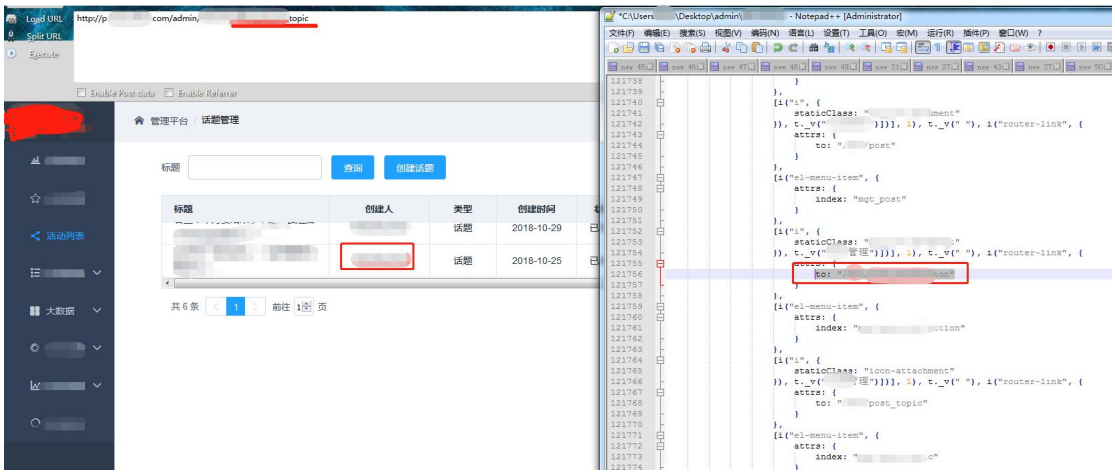
尝试多个弱口令无果，于是尝试从 js 压缩包查找密码规则

把每个 JS 源码格式化后 从 *******.js 文件中得到了用户默认密码的规则

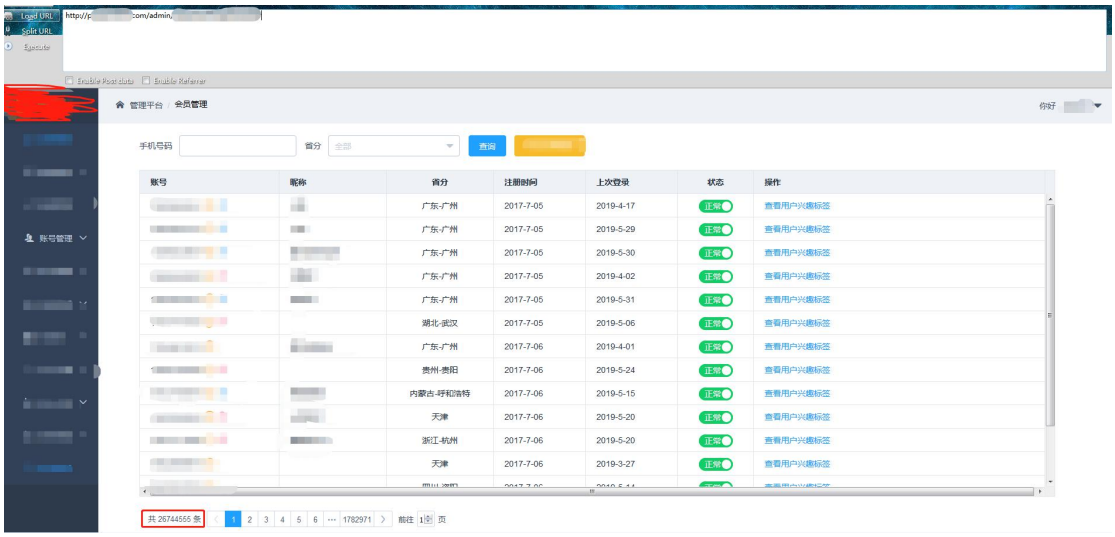
密码为账号后四位数字+ *******

```
60689                    })
60690                })
60691            },
60692            resetPwdFunc: function(t) {
60693                var e = this;
60694                e.$confirm("确定重置密码?密码恢复为账号后四位数字+[redacted]"提示", {
60695                    confirmButtonText: "确定",
60696                    cancelButtonText: "取消",
60697                    type: "warning"
60698                }).then(function() {
60699                    config.reqPost({
60700                        url: "/mgt/resetpwd",
60701                        params: {
60702                            code: t
60703                        },
60704                        success: function(t, i) {
60705                            e.$message({
60706                                type: "success",
60707                                message: i
60708                            })
60709                        }
```

使用账号 1******1234 登陆后发现权限并不大，然后通过 js 获取到了别的接口地址，发现存在越权漏洞，通过 JS 接口越权访问到活动管理页面获取到管理员的登陆账号

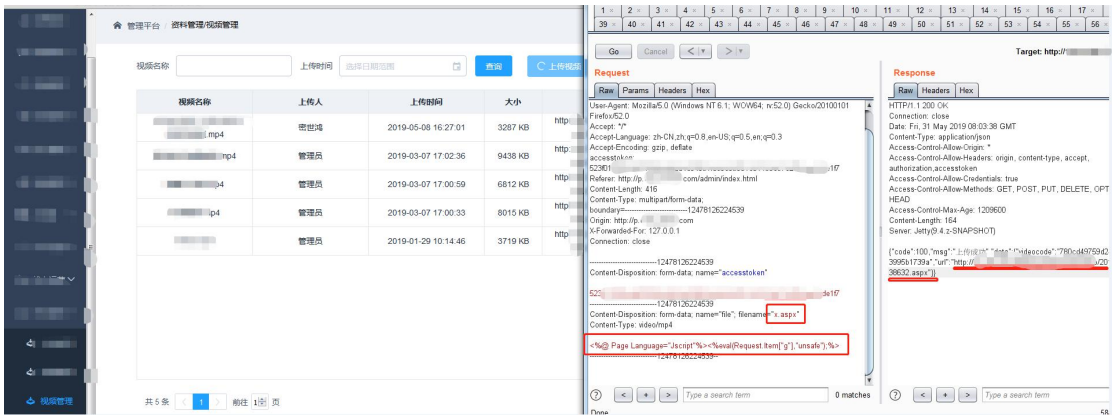


通过登陆管理员账号 1*******5678 发现可操作 2600 万+会员信息...



然后通过 资料管理 –> 管理 上传的地方 抓包修改文件后缀成功拿到网站 webshell

http://p.xxx.*****.com/admin/****/*****/***



shell 地址：http://**.***.**.**/********/*****/****/*******/******.aspx

任务完成，看了一下内网很大，数据海量、内网系统也很多，但是没有授权搞内网，只能点到为止。

其实登录后漏洞很简单，无限制任意上传，但是大多数都是被卡在了登录前，遇到这样的站，一般都是尝试绕过验证码爆破，无法绕过就尝试识别，无法识别，就各种扫目录扫端口，但是大多数的站点前端采用 VUE 开发，各种功能都是以接口的形式调用，你扫目录、扫文件没多少用处，还有可能触发报警，有时候右键查看 JS 源码，你可能会发现... 被注释的账号密码、接口、token、真实 IP、开发环境.... 永远不知道程序员在 JS 中给你留下了什么样的惊喜。

R3start

Github 地址：渗透测试案例

用于记录分享一些有趣的案例