

某授权项目渗透测试

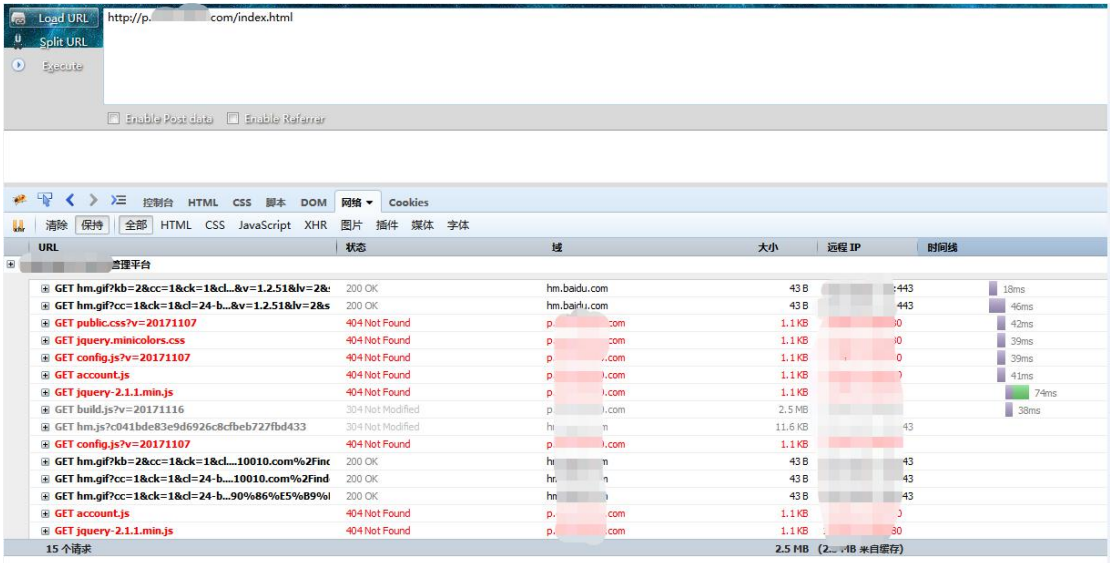
测试目标站点: https://xxx.****.com/#/

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

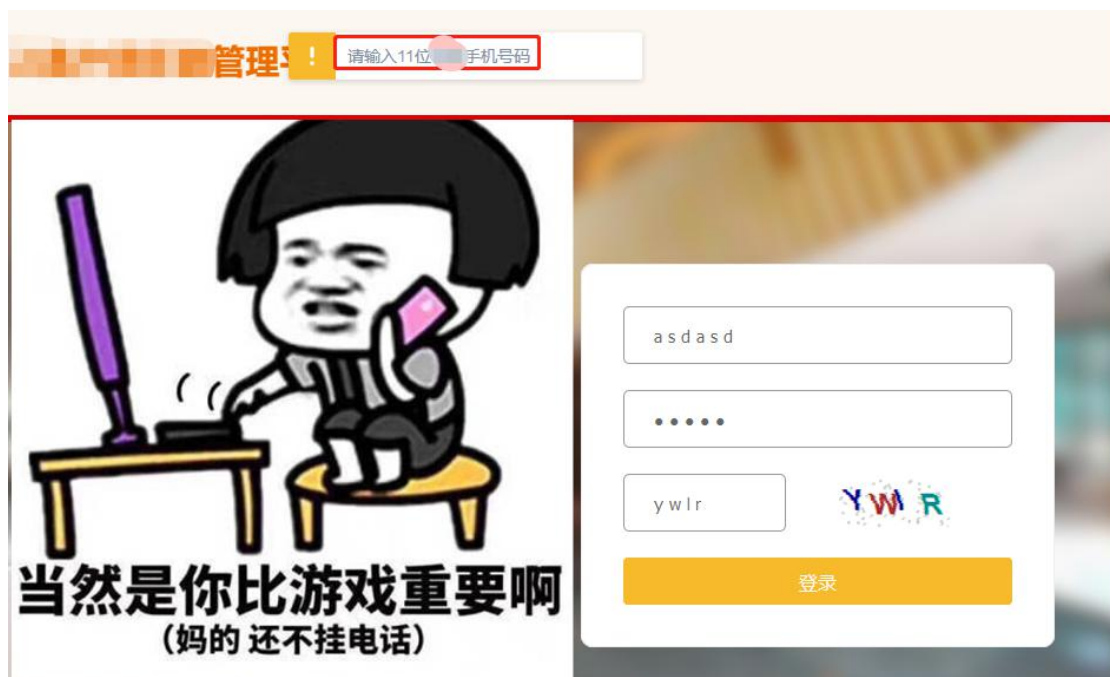


http://p.com/index.html	200
http://p.com/favicon.ico	200
http://p.com/admin/index.html	200
http://p.com/Index.html	200
http://p.com/map/	200
http://p.com/admin/admin.rar	200
http://p.com/admin/index.html	200
http://p.com/index.html	200
http://p.com/admin/index.html	200
http://p.com/index.html	200
http://p.com/index.html	200
http://p.com/index.html	200
http://p.com/admin/index.html	200

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 F12 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


```

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

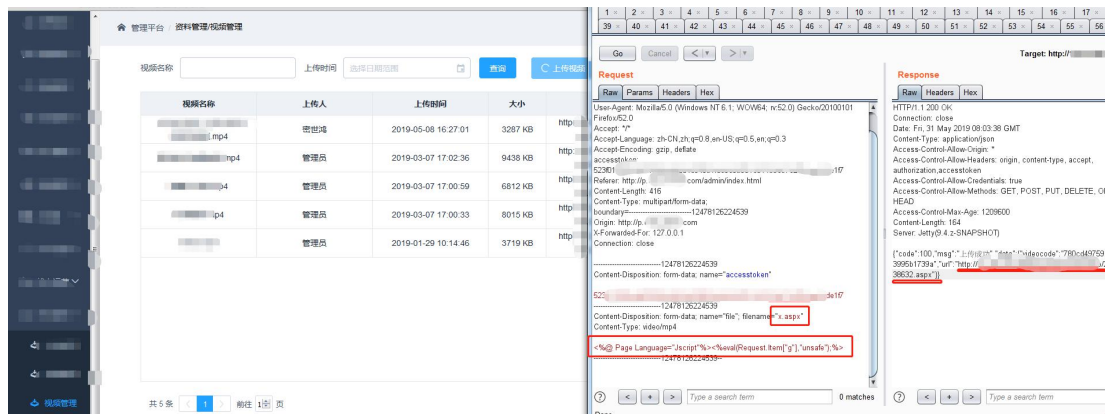
The screenshot shows a web application interface on the left and a Notepad++ editor on the right. The web interface displays a table for 'Topic Management' (话题管理) with columns for 'Title' (标题), 'Creator' (创建人), 'Type' (类型), and 'Creation Time' (创建时间). The Notepad++ editor shows a JavaScript file with a menu item configuration that includes a 'post' action, which is highlighted with a red box.

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

The screenshot shows a web application interface for 'Member Management' (会员管理). It features a search bar with '手机号码' (Mobile Number) and '积分' (Points) filters, and a table listing user accounts. The table has columns for 'Account Number' (账号), 'Name' (昵称), 'Region' (省份), 'Registration Time' (注册时间), 'Last Login' (上次登录), 'Status' (状态), and 'Action' (操作). The table shows a list of users with their respective details and actions.

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

5 commits

1 branch

0 releases

1 contributor

Branch: master

New pull request

Find File

Clone or download

r3start Update README.md

Latest commit 83Fee3e 3 days ago

CVE-2019-1003000 Jenkins-PreAuth-RCE 复现过程.pdf	Add files via upload	3 days ago
README.md	Update README.md	3 days ago
低危SSRF提权进内网.pdf	Add files via upload	3 days ago
某第三方支付边界机漏洞导致的内网渗透.pdf	Add files via upload	3 days ago
记一次有趣的命令执行.pdf	Add files via upload	3 days ago

README.md

Penetration_Testing_Case

建立此项目的目的,是希望自己能够在日常的渗透工作中,总结分享出一些个人的渗透经验和有趣的案例,此项目中所有的案例仅用于经验交流。

部分文章可能由于种种原因打码过于严重,如影响阅读请见谅。

若发现有打码不全的文章请务必联系我删除。

我会陆续的更新此项目。

<div>

Author:[R3start](#) 2019年7月11日