

- - 前言
  - 示例
    - 示例一
    - 示例二
    - 示例三
  - 总结

## 前言

---

Author: 0x584A

总结下在测试工作中，Web系统因前端文件导致的安全问题，通过分析前端代码找到路由配置，在对 PATH 进行调用，往往会有意想不到的结果出现。

- 在页面源代码中找注释，可能会存在敏感配置信息或功能代码说明
- JS代码中搜索关键字（setPass,password,router,path等），分析前端代码使用框架类型，查找路由配置信息及硬编码配置

## 示例

---

### 示例一

某日在渗透内部系统，在搜索 JS 代码时发现含有可疑URL接口路径，"xxxProxy/passwordResetByMobileCode"，尝试调用后提示：请求类型错误需要POST提交、参数错误。

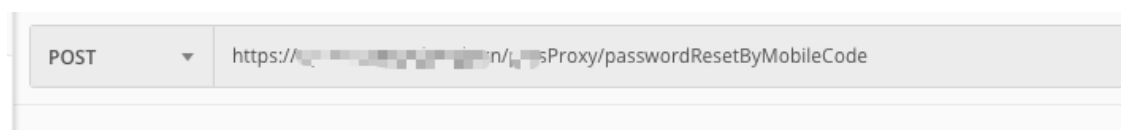
```
14093     params: e
14094   })
14095 },
14096 passwordResetByMobileCode: function (e) {
14097   return u/f
```

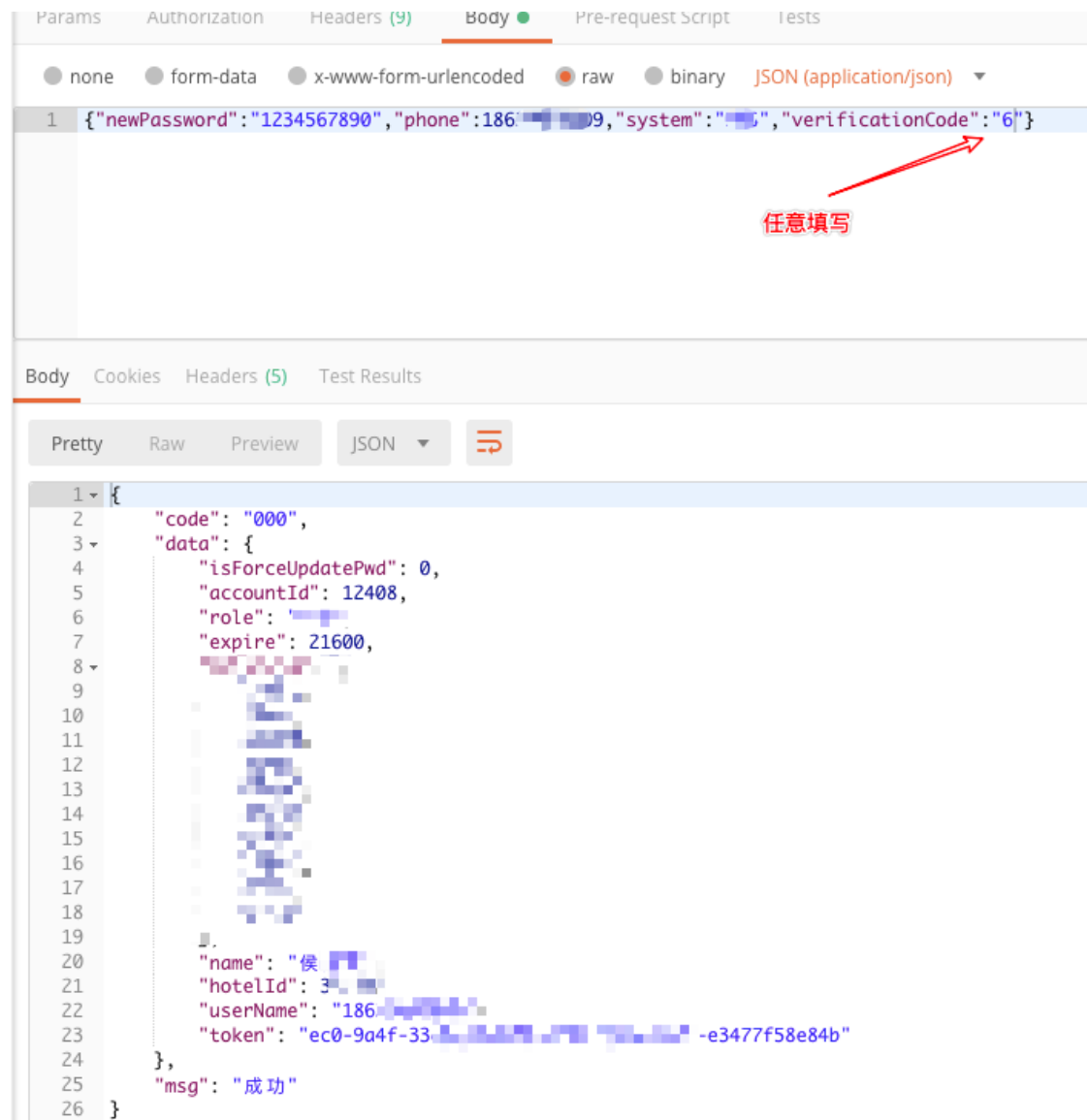
```
JS chunk-vendors.8dca531b.js 14097 return w({
JS um.js 14098 url: '/Proxy/passwordResetByMobileCode',
at.alicdn.com 14099 data: e
moz-extension:// 14100 })
efu.easemob.com 14101 },
resource:// 14102 loginByMobileCode: function (e) {
Webpack 14103 return w({
14104 url: '/Proxy/loginByMobileCode',
14105 data: e
14106 })
14107 }
```

随后在找回密码页面（xxx/forgetPassword）发现加载了一个 'chunk-5df4b15a.ea1375be.js' 文件，其中存在上面看到的关键字 'passwordResetByMobileCode'，并且代码中含有疑似接口所需要的参数。

```
var t, e, n;
return l.a.wrap(function(o) {
  for (; ; )
    switch (o.prev = o.next) {
      case 0:
        if (t = {
          phone: this.$store.state.user.userName,
          verificationCode: this.$store.state.user.verificationCode,
          newPassword: this.formCustom.newPassword,
          system: "PC"
        },
        e = this.$auth.login.getIsFirstLogin(),
        n = null,
        "1" === e)
          return o.next = 6,
            this.$service.user.passwordResetByToken(t);
        o.next = 9;
        break;
      case 6:
        n = o.sent,
        o.next = 12;
        break;
      case 9:
        return o.next = 11,
          this.$service.user.passwordResetByMobileCode(t);
      case 11:
        n = o.sent;
        ---- 12:
    }
```

尝试构建所需要的参数提交，发现存在任意密码重置漏洞，关键的 **verificationCode** 参数后端无校验。





## 示例二

这个与上面的差不多，同样在分析前端 JS 代码中，发现一个关键字

URL: `/user/setNewPass`，浏览器访问后是设置新密码页面，填写密码后提示参数错误。

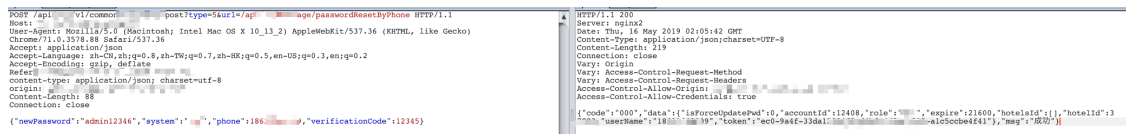
在重置密码页面 `/user/resetPass` 打开时，会加载了一个新的 JS 文件，分析代码并补全提交参数：phone、verificationCode。

```
if (!e) {
  var i = t.props,
  c = i.dispatch;
  i.login;
```

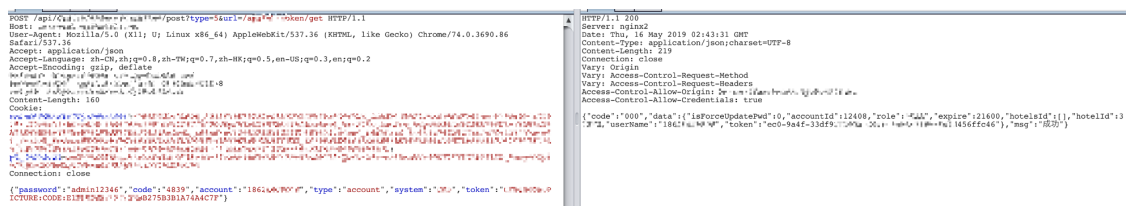
```
Object(E['g']) ({
  phone: a,
  verificationCode: o
}).then(function () {
  c({
    type: 'login/saveResetpassInfo',
    payload: {
      phone: a,
      verificationCode: o
    }
  }),
  v['a'].push({
    pathname: '/user/setNewPass',
    query: {
      type: 'forget'
    }
  })
}).catch(function (e) {
  console.log(e)
})
})
```

参数提交后发现并没有校验 **verificationCode** 的真实性，填写任意内容都可以重置成功。

账号原密 admin12345 重置成 admin12346：



随后用重制后的密码登录，提示成功：



## 示例三

某天在 GITHUB 上发现可疑的存在生产环境域名的文件，指向某分析平台：

未处理	<b>Impossible/coc-form - Login.vue</b> 更新时间：2019-07-26 08:50:39 发现时间：2019-07-27 13:15:43
290	let redirectURL = btoa(`\${location.protocol}://\${location.host}\${config.mainPath}`)
201	console.log(redirectURL)

```
291 console.log(redirectURL);
292 var url = encodeURIComponent(
293 // `http://192.168.1.100:8080/jingtalk/authCallBack?redirect_url=${redirectURL}`
```

查看 github lnmpossible/xxx 项目后，确认为某分析平台前端产线代码。

项目为 vue.js 开发，查看配置文件发现与生产环境一致，URL 含有 /pxxxform/ 路径：

```
},
publicPath: ['production'].includes(process.env.NODE_ENV) ? '/platform/' : '/',
outputDir: 'platform',
filenameHashing: true,
pwa: {
  'name': '数据',
  'iconPaths': {
    favicon32: 'img/icons/...',
    favicon16: 'img/icons/...',
    appleTouchIcon: 'img/icons/oyo-logo-152.png',
    maskIcon: 'img/icons/safari-pinned-tab.svg',
    msTileImage: 'img/ico!...'
  }
}
```

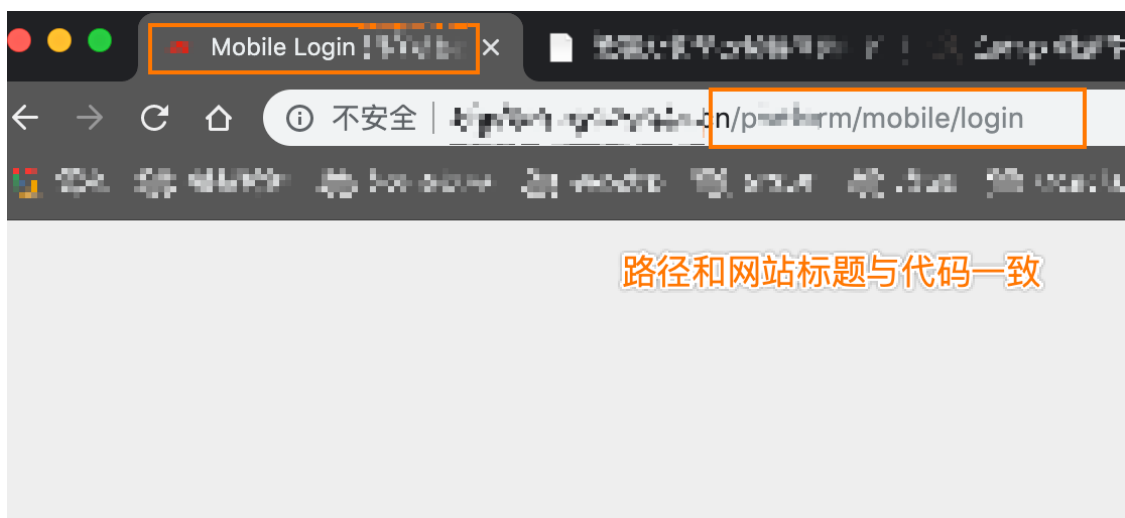
寻找前端路由验证代码是否一致：

```
4 export default new Router({
5   mode: 'history',
6   base: '/platform/mobile',
7   routes: [
```

```

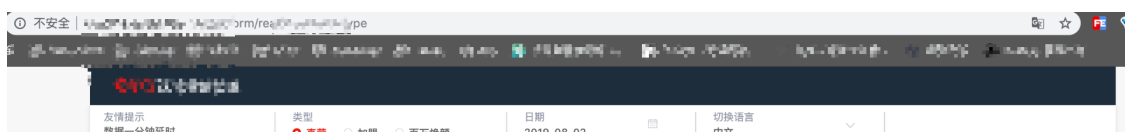
7 Routes: [
8   {
9     name: 'main',
10    path: '/',
11    // component: () => import(/* webpackChunkName: 'main'
12    component: MobileHome,
13    meta: {
14      title: 'Mobile Navigator | [REDACTED]',
15      keepAlive: true
16    }
17  },
18  {
19    name: 'login',
20    path: '/login',
21    component: () => import(/* webpackChunkName: 'login'
22    meta: {
23      title: 'Mobile Login | [REDACTED]',
24      keepAlive: true
25    }
26  },
27  {
28    name: 'nav',
29    path: '/nav',

```



账号 请输入手机号或姓名

通过分析泄漏代码，汇总前端路由路径、API调用路径发现存在多个未授权访问。多个敏感页面可以在外网直接浏览（含住和离时间、联系方式及身份证信息）。





用户信息

开始日期: 至: 结束日期: 15 11:11:17

注册时间	用户ID	注册电话	登录IP	是否员工	注册来源(一级)	注册来源(二级)	注册来源(三级)	注册二维码	注册场景	推荐注册活动	推荐人用户ID	推荐人手机号	风险等级
2019-11-08 08:05:37	CN_15241505566	15241505566		N	线上引流	今日注册	今日注册-摩力-安卓93	N	Android App	Android App			Android App

#### 标签信息

创建时间	用户ID	用户标签	是否有效	标签类型	备注
暂无数据					

< 1 >

#### 推荐信息

注册时间	用户ID	登录IP	是否员工	注册来源(一级)	注册二维码	手机	USER_TYPE	注册场景	推荐注册活动	推荐人用户ID	推荐人手机号	注册电话省份	注册电话城市
暂无数据													

< 1 >

#### 用户状态信息

用户ID	注册风控结果	注册时间	是否有预订记录	是否有入住且离店的记录	是否有预订且入住酒店	是否有券预订且入住酒店	是否有身份证读卡器入住且离店记录	第一次预订时间	第一次入住时间	第一次用券预订且入住并离店的入住时间	第一次用券预订且入住并离店的入住时间	第一次读卡器入住且离店的入住时间	预订单数	最近入住时间
CN_15241505566	Accept	2019-11-08 08:05:37	Y	Y	Y	Y	Y	2019-11-08 08:05:37	2019-11-08 08:05:37	2019-11-08 08:05:37	2019-11-08 08:05:37	2019-11-08 08:05:37	1	2019-11-08 08:05:37

#### 用户事件信息



#### 酒店信息

酒店ID	酒店名	街道	城市	邮箱	电话	简称	地址描述	纬度	经度	主要联系电话	地址电话	经理	当前佣金率	是否活跃
CN_15241505566	61 县里汉邦酒店	陕西省西安市西大街	西安	NA	155150556600	61 县里汉邦酒店	陕西省西安市西大街	34.2613	108.9421	+911111111111	+8629150556600	钟永刚	3	active

#### 酒店状态信息

1M预订单数	1M入住且离店单数	1M预订且入住并离店单数	1M用券预订且入住并离店单数	1M身份证入住且离店单数	预订单数	入住且离店单数	预订且入住并离店单数	用券预订且入住并离店单数	身份证入住且离店单数	1M预订用户数	1M入住且离店用户数	1M预订且入住并离店用户数	1M用券预订且入住并离店用户数	1M身份证入住且离店用户数
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

#### 酒店订单信息

酒店订单信息													
酒店订单信息													
入住时间	预定单号	房单单号	评分	会员ID	会员名	用户电话号码	酒店ID	操作人ID	销售ID	客户姓名	客户电话	是否预订	是否入住
2019-11-08 08:05:37	PB15241505566	1505566	0.00	0			45629	CN_15241505566	0			N	
2019-11-08 08:05:37	PB15241505566	1505566	0.00	0			45629	CN_15241505566	0			N	
2019-11-08 08:05:37	PB15241505566	1505566	0.00	0			45629	CN_15241505566	0			N	

联系到项目所属人为离职员工，已经联系其删除了该项目。

## 总结



在前后端分离开发时，常出现接口或页面的未授权访问、越权等问题。后端不能完全信任用户提交过来的参数，必须对参数及业务流程进行严格校验。