

JS敏感信息泄露

对于JS泄露敏感信息问题，攻击者不仅可以轻松收集用户手机号，姓名等隐私信息，更可以借此攻入企业后台甚至是getshell。本文将通过一些公开和未公开的漏洞详细阐述此类漏洞。一个微小的漏洞，经过攻击者的巧妙而持久的利用，也会对企业 and 用户造成巨大的危害。

1、漏洞成因

JavaScript作为一种相当简单但功能强大的客户端脚本语言，本质是一种解释型语言。所以，其执行原理是边解释边运行。上述特性就决定了JavaScript与一些服务器脚本语言（如ASP、PHP）以及编译型语言（如C、C++）不同，其源代码可以轻松被任何人获取到。一些粗心的开发者将各式敏感信息存储在JavaScript脚本中，由于JS的特性，攻击者可以对这些信息一览无余，从而导致对WEB服务和用户隐私造成不同程度的威胁。

2、漏洞分类及利用

根据泄露的内容、利用方式以及带来的危害不同，大致可以将JS敏感信息泄露分为以下三类：

1. JS文件泄露后台管理敏感路径及API

此类问题主要存在于后台登陆页面以及类似网页内引入的JS文件中。在企业渗透测试时如果遇到后台，在SQL注入或者是路径爆破都试过，但是仍然无法进入后台时。根据此类漏洞，说不定登陆页面下引入的js文件暴露的后台路径会成为突破口。如果某台的某一个页面没有对是否登陆状态做验证，攻击者就可以一次未授权访问这些暴露的API，实现篡改前台内容甚至是getshell。下面这个某大型互联网服务提供商的房产后台页面引入的js文件泄露后台接口信息，就是非常典型的例子：



drops.wooyun.org

```
<html>
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
    <meta charset="utf-8" />
    <title>房产-用户登录</title>
    <link rel="stylesheet" href="http://img1.cache.wooyun.com/f2e/house/members/css/common.733798.css" />
    <link rel="stylesheet" href="http://img2.cache.wooyun.com/f2e/house/members/css/members.931038.css" />
    <script src="http://img2.cache.wooyun.com/f2e/libs/underscore.js"></script>
    <script src="http://img2.cache.wooyun.com/f2e/libs/jquery.js"></script>
    <script src="http://img2.cache.wooyun.com/f2e/libs/backbone.js"></script>
    <script src="http://img1.cache.wooyun.com/f2e/house/members/js/common.833008.min.js"></script>
```

drops.wooyun.org

(信息泄露源是房产管理登陆后台页面下的common.js)

(js文件中的urlMap对象内容完整的泄露了后台所有功能实现的API，因此攻击者无需登陆就可以进行未授权操作)

经过以往测试经验的归纳，此类漏洞常见该类型的泄露常见于网站评论功能实现处。由于程序员疏忽直接在页面的js以及AJAX请求返回的内容中输出参与评论用户的敏感信息，导致攻击者可以轻松获取用户的手机号，真实姓名，注册邮箱，住址甚至有可能获取用户手机的IMEI，和抽奖和抽中的游戏礼包号。从个人的测试经验来看，涉及的厂商范围广泛，从某搜索服务提供商下的订票网站、在线挂号网站到某公司旗下的大型游戏网站都发现过此类漏洞，其他厂商肯定也存在该类问题。不需要SQL注射脱库，黑产哥通过编写爬虫就能大批量的获取用户的敏感信息，利用难度小，危害却很大。

(某FM类应用评论处泄露匿名评论用户邮箱, 手机号)

(某大型游戏网站抽奖页面下泄露获奖用户礼包兑换号码)

此类漏洞修复相对容易，在明白了JavaScript的特性以后，不把此类敏感信息直接存储进页面内的js和ajax请求响应内容中就可以解决这类问题。

js信息收集

JSFinder

项目地址: <https://github.com/Threezh1/JSFinder>

一、查看帮助信息

```
python3 JSFinderPlus.py -h
```

帮助

```
usage: JSFinder.py [-h] [-u URL] [-c COOKIE] [-f FILE] [-ou OUTPUTURL] [-os
OUTPUTSUBDOMAIN] [-j] [-d]
```

optional arguments:

```
-h, --help            show this help message and exit
-u URL, --url URL      The website
-c COOKIE, --cookie COOKIE
                        The website cookie
-f FILE, --file FILE  The file contains url or js
-ou OUTPUTURL, --outputurl OUTPUTURL
                        Output file name.
-os OUTPUTSUBDOMAIN, --outputsubdomain OUTPUTSUBDOMAIN
                        Output file name.
-j, --js              Find in js file
-d, --deep            Deep find
```

二、简单爬取得

```
python3 JSFinderPlus.py -u http://172.16.71.140:8088
```

三、深度爬取

```
python3 JSFinderPlus.py -u http://172.16.71.140:8088 -d
```

-d 对发现的URL进行查找，默认只查找输入的URL

四、结果保存

1、-ou 指定文件名保存URL链接

```
python JSFinder.py -u http://www.test.com -ou url.txt
```

2、os 指定文件名保存子域名

```
python JSFinder.py -u http://www.test.com -os subdomain.txt
```

3、-c 指定cookie来爬取页面 例:

```
python JSFinder.py -u http://www.test.com -c "session=xxx"
```

4、案例

```
python3 JSFinder.py -u http://www.test.com -ou url.txt -os domain.txt
```

-ou 指定文件名保存URL链接

-os 指定 指定文件名保存子域名

注意

url 不用加引号

url 需要http:// 或 https://

指定JS文件爬取时，返回的URL为相对URL

指定URL文件爬取时，返回的相对URL都会以指定的第一个链接的域名作为其域名来转化为绝对URL。

URLFinder

项目地址: <https://github.com/ping0y/URLFinder>

一、查看帮助信息

URLFinder.exe -h

二、简单爬取得

单url

显示全部状态码

```
URLFinder.exe -u http://www.cisp.com -s all -m 3
```

显示200和403状态码

```
URLFinder.exe -u http://www.cisp.com -s 200,403 -m 3
```

批量url

结果分开保存

导出全部

```
URLFinder.exe -s all -m 3 -f url.txt -o .
```

只导出html

```
URLFinder.exe -s all -m 3 -f url.txt -o res.html
```

结果统一保存

```
URLFinder.exe -s all -m 3 -ff url.txt -o .
```












参数 (更多参数使用 -i 配置) :

```
-a 自定义user-agent请求头
-b 自定义baseurl路径
-c 请求添加cookie
-d 指定获取的域名,支持正则表达式
-f 批量url抓取,需指定url文本路径
-ff 与-f区别: 全部抓取的数据,视为同一个url的结果来处理 (只打印一份结果 | 只会输出一份结果)
-h 帮助信息
-i 加载yaml配置文件,可自定义请求头、抓取规则等 (不存在时,会在当前目录创建一个默认yaml配置文件)
-m 抓取模式:
    1 正常抓取 (默认)
    2 深入抓取 (URL深入一层 JS深入三层 防止抓偏)
    3 安全深入抓取 (过滤delete,remove等敏感路由)
-max 最大抓取数
-o 结果导出到csv、json、html文件,需指定导出文件目录 (.代表当前目录)
-s 显示指定状态码,all为显示全部
-t 设置线程数 (默认50)
```

```
-time 设置超时时间（默认5,单位秒）
-u 目标URL
-x 设置代理,格式: http://username:password@127.0.0.1:8877
-z 提取所有目录对404链接进行fuzz(只对主域名下的链接生效,需要与 -s 一起使用)
    1 目录递减fuzz
    2 2级目录组合fuzz
    3 3级目录组合fuzz（适合少量链接使用）
```

浏览器插件FindSomething

FindSomething用于快速在网页的html源码或js代码中提取一些有趣的信息，包括可能请求的资源、接口的url，可能请求的ip和域名，泄漏的证件号、手机号、邮箱等信息。

主页配置			
处理中..0/0			
IP	复制	Path	复制URL复制
			
IP_PORT	复制	IncompletePath	复制
			
域名	复制	Url	复制
			
身份证	复制	StaticUrl	复制
			
手机号	复制		
			
邮箱	复制		
			
JWT	复制		
			
算法	复制		
			
Secret	复制		

处理完成: 51/51

IP

复制



IP_PORT

复制



域名

复制

< .og.csdn.net
< .n.net
< .gle.cn
< .
< .com
< .uaweicloud.com
< .fame.googlesyndication.com
< .p
< //bj.baidu.com
< //idmg.cn
< //eli.baidu.com
< //foogoogleapis.com
< //gimg.cn
< //naming.csdn.net
< //nyon.net
< //cdu.com
< //bdu.com/
< //dnet
< //sdn.net
< //statisticdn.net/

Path

复制URL 复制

< /..
< /1
< 88115/example/
< z]*
< .
< .cle/details
< aacle/details/
< brnk-policy.htm
< cli/bai
< /o/barers?b=
< /o-in-policy.htm
< /eent
< /erCorrectLevel:
< /eensions
< .
< =
< .pd/as?
< /
< /ntner.html
< .e
< .
< .
< .
< .s/
< .k.w/api/timesWin
< /outn