

# JSONP 漏洞自动挖掘方法研究

杨传栋, 曲洋

(公安部第三研究所, 上海 201204)

摘要: JSONP 是一种支持浏览器内跨域信息交换的技术, 可用于不同域间的数据传递, 由于该技术灵活方便使其在 Web 领域得到了广泛应用, 但安全问题有待解决, 倘若网站在处理 JSONP 请求时没有严格检查来源会产生 JSONP 漏洞。JSONP 漏洞易导致敏感信息泄露, 危害极其严重。目前 JSONP 漏洞的挖掘方法非常有限, 主要靠手动方式完成, 目前该方法效率较低, 且挖掘不全面。针对 JSONP 漏洞手动挖掘方式的不足提出了一种基于 Chrome 插件的 JSONP 漏洞自动挖掘方法, 通过该方法可高效、自动、全面地挖掘网站中存在的 JSONP 漏洞。

关键词: JSON; JSONP; 同源策略; 回调函数; 漏洞挖掘

中图分类号: TP393.08 文献标识码: A 文章编号: 1009-3044(2016)35-0017-02

## 1 引言

随着信息技术的快速发展, 业务量的不断丰富, 互联网对于 Web 技术提出了更高的要求。JavaScript<sup>[1]</sup>的问世无疑更加丰富了用户体验, JSON 数据格式的广泛应用让前端技术的发展有了质的飞跃。所有支持 JavaScript 的浏览器都不允许页面访问非同源<sup>[2]</sup>(同源是指域名、协议、端口相同)信息, 然而在实际的业务中, 访问非同源信息的场景不可避免, 正因如此, JSONP<sup>[3]</sup>技术得以广泛应用, 成为一种非官方跨域数据交互协议。

JSONP 带来便利的同时产生了相应的安全问题, 如果没有合理利用 JSONP 易产生 JSONP 漏洞, 最终导致隐私泄露, 若被违法犯罪分子利用, 将会带来不可估量的损失。目前 JSONP 漏洞的挖掘主要是靠手工方式, 挖掘效率低且挖掘不够全面。针对 JSONP 漏洞手工挖掘方式的不足, 本文首次提出了基于 Chrome 插件形式的 JSONP 漏洞自动挖掘方法, 能够高效、自动、全面的挖掘网站的 JSONP 漏洞, 为 JSONP 漏洞的挖掘提供了新思路。

## 2 JSONP 漏洞介绍

### 2.1 JSON 与 JSONP

JSON<sup>[4]</sup>是 JavaScript Object Notation 的缩写, 是一种轻量、可读基于文本的数据交换开放标准。源于 JavaScript 编程语言中对简单数据结构和关联数组的展示功能, 它是仅含有数据对和简单括号结构的纯文本, 可通过多种途径进行 JSON 消息的传递。

JSONP 是 JSON with Padding 的缩写, 是一种非官方协议。在同源策略下, 某个服务器下的页面是无法获取该服务器以外数据的, 但 img、iframe、script 等标签是例外, 这些标签可以通过 src 属性请求到其他服务器上的数据。JSONP 的原理从本质上讲是利用 <script> 标签的 src 属性不受同源策略的限制, 访问跨域 URL。基于这一机制, 可以在源网页中注册一个回调函数, 然后在跨域服务器上调用这一回调函数<sup>[5]</sup>, 将服务器数据以参

数形式传递给回调函数, 并将服务器上这段代码以注入的方式添加到源网页 <script> 标签的 src 属性中, 从而实现服务器数据的跨域访问。

### 2.2 JSONP 漏洞

通过 JSONP 技术可以实现数据的跨域<sup>[6]</sup>访问, 必然会产生安全问题, 如果网站 B 对网站 A 的 JSONP 请求没有进行安全检查直接返回数据, 则网站 B 便存在 JSONP 漏洞, 网站 A 利用 JSONP 漏洞能够获取用户在网站 B 上的数据。

JSONP 漏洞利用过程如下:

- 1) 用户在网站 B 注册并登录, 网站 B 包含了用户的 id, name, email 等信息;
- 2) 用户通过浏览器向网站 A 发出 URL 请求;
- 3) 网站 A 向用户返回响应页面, 响应页面中注册了 JavaScript 的回调函数和向网站 B 请求的 script<sup>[7]</sup> 标签, 示例代码如下:

```
<script type="text/javascript">
function Callback(result)
{
    alert(result.name);
}
</script>
<script type="text/javascript" src="http://B.com/user?jsonp=Callback"></script>
```

- 4) 用户收到响应, 解析 JS 代码, 将回调函数作为参数向网站 B 发出请求;

5) 网站 B 接收到请求后, 解析请求的 URL, 以 JSON 格式生成请求需要的数据, 将封装的包含用户信息的 JSON 数据作为回调函数的参数返回给浏览器, 网站 B 返回的数据实例如下: Callback({"id":1,"name":"test","email":"test@test.com"}).

- 6) 网站 B 数据返回后, 浏览器则自动执行 Callback 函数对步骤 4 返回的 JSON 格式数据进行处理, 通过 alert 弹窗展示了用户在网站 B 的注册信息。另外也可将 JSON 数据回传到网站

收稿日期: 2016-11-10

基金项目: 2016 基本科研业务专项资金(C16356)

作者简介: 杨传栋(1987—), 男, 山东郯城人, 研究实习员, 硕士, 主要研究 Web 网络安全; 曲洋(1988—), 男, 吉林人, 研究实习员, 硕士, 主要研究方向为信息网络安全。

本栏目责任编辑: 代 影

A的服务器,这样网站A利用网站B的JSONP漏洞便获取到了用户在网站B注册的信息。

整个JSONP漏洞利用过程如图1所示:

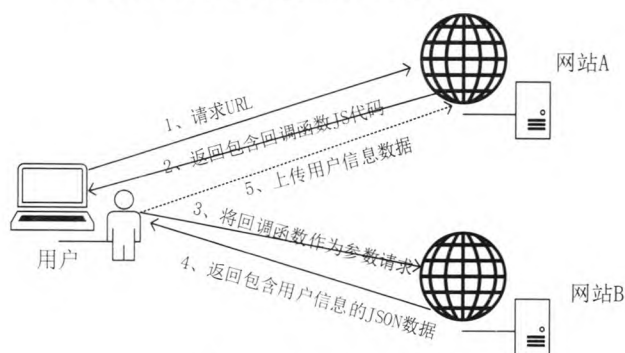


图1 JSONP漏洞利用过程

## 2.3 JSONP漏洞危害

JSONP是一种敏感信息泄露的漏洞,经过攻击者巧妙而持久地利用,会对企业和用户造成巨大的危害<sup>[8]</sup>。攻击者通过巧妙设计一个网站,网站中包含其他网站的JSONP漏洞利用代码,将链接通过邮件等形式推送给受害人,如果受害者点击了链接,则攻击者便可以获取受害者的个人的信息,如邮箱、姓名、手机等信息,这些信息可以被违法犯罪分子用作“精准诈骗”。对方掌握的个人信信息越多,越容易取得受害人的信任,诈骗活动越容易成功,给受害人带来的财产损失以及社会危害也就越大。

## 3 传统的JSONP漏洞挖掘方法

目前关于JSONP漏洞的挖掘分析大多以手动方式挖掘,效率较低。本文以手动方式挖掘baidu.com网站的JSONP漏洞为例来说明整个挖掘分析过程。

1)用Chrome浏览器打开baidu.com并登录账号;

2)打开Chrome浏览器调试窗口的“Network”选项,勾选“Preserve log”,以防止页面刷新跳转的时候访问记录被重置,重新刷新页面会看到网络数据,如图2所示。

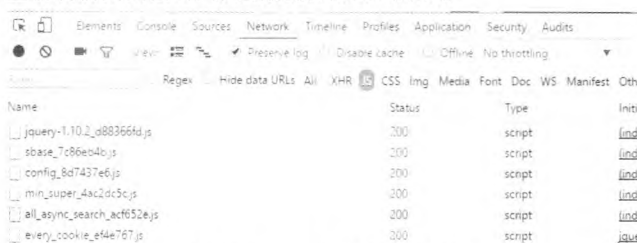


图2 Chrome浏览器调试窗口

3)JSONP漏洞返回的是JSON格式的数据,在漏洞挖掘的过程中主要查看JS文件,需要重点关注返回的含有回调函数的JSON格式数据,查看数据中是否包含登录的账户信息,如果含有则极可能是JSONP漏洞,需进一步验证,如图3所示是含有回调函数的JSON格式数据。



图3 含有回调函数的JSON格式数据

4)如果没有找到,则打开当前页面的其他链接进行尝试,在该域名的链接尝试完之后打开其他子域名(如:zhidao.baidu.com)链接重新进行尝试,直至所有的子域名链接全部查找完毕。

## 4 JSONP漏洞自动挖掘

### 4.1 JSONP漏洞自动挖掘

JSONP漏洞的手动挖掘方法耗时耗力,且能否成功与否依赖于挖掘经验。针对上述不足,本文提出了基于Chrome插件的JSONP漏洞自动挖掘方法,其核心思想是将手动挖掘过程以自动方式实现,具体通过开发一款Chrome插件实现JSONP漏洞的自动挖掘。

JSONP漏洞自动挖掘Chrome插件框架设计如图4所示,主要包括输入模块、爬虫模块、数据分析模块和输出模块。

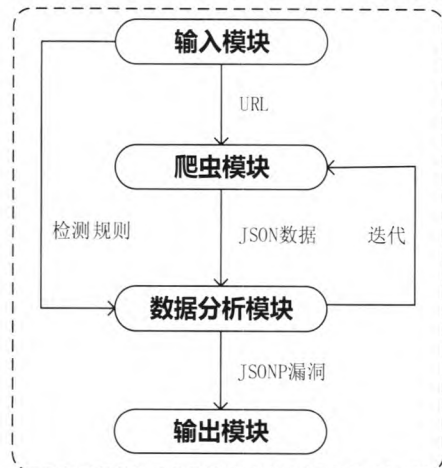


图4 JSONP漏洞自动化挖掘Chrome插件框架设计

JSONP漏洞分析挖掘过程中,各模块的功能简述如下:

1)输入模块:包含输入的URL和检测规则,其中检测规则是注册URL的账户标识信息,包括用户名、ID、email、手机号等信息。检测规则越多,JSONP漏洞挖掘会越全面。输入模块将URL传递给爬虫模块,将检测规则传递给数据分析模块。

2)爬虫模块:根据输入模块传递来的URL进行爬虫,该过程采用的是广度优先算法,只爬取URL域名及其子域名的链接。爬虫模块将获取到的JSON格式数据传递给分析模块,然后根据爬虫算法进行迭代,直至爬取URL域名及其子域名的所有链接。

3)数据分析模块:根据检测规则处理爬虫模块传递来的JSON格式数据,具体分为以下两步:

第一步,根据关键词比对,将包含关键词及回调函数的JSON数据链接筛选出来作为疑似JSONP漏洞。

第二步,通过在<script>标签的src引用疑似JSONP漏洞做进一步验证,若引用时能获取到关键词数据则说明是JSONP漏洞传递给输出模块,否则不是JSONP漏洞丢弃。

4)输出模块:输出模块接收数据分析模块传递来的JSONP漏洞,并以UI形式进行输出。

### 4.2 实验

通过开发的这款基于Chrome自动挖掘JSONP漏洞的插件对baidu.com进行了测试,登录百度账号,启动插件测试30分钟发现了3个百度JSONP漏洞,其中一个漏洞非常严重,该漏洞

(下转第21页)

参考文献:

[1] 刘志勇. 基于云计算的轻工行业公共云服务平台研究[D]. 湖南大学,2013.  
[2] 李英娟. 基于 SOA 的云计算企业资源计划模型研究[D]. 内蒙古大学,2014.  
[3] 温伟. 云计算环境下内存计算与移动无线开发技术在 ERP 系统中的研究与实现[D]. 北京交通大学,2013.

[4] 陶宏林. 基于云计算的库存管理系统设计与实现[D]. 吉林大学,2012.  
[5] 沈克勤,李焱炜,阮国军. 云计算环境下印刷业 ERP 的虚拟机安全策略[J]. 电脑知识与技术,2014,20:4697-4698.  
[6] 徐松. 基于云计算的电子商务 ERP 系统的设计与实现[D]. 南京邮电大学,2015.

(上接第18页)

包含了用户的百度 ID、用户名、邮箱、性别、邮箱活跃程度、手机、头像地址、注册时间、上次登录时间等,漏洞泄露的敏感信息如图 5 所示,其中关键信息部分打码处理。

```
3{"user_bname":"","nick":"","realname":"","sex":"2","email":"","0126.com","email_active":"1","st
"city_info":"","user_auth":"1","user_add_official":0,"user_telno_md5":"077818","artistid":"","
0126.com","securemobil":"","un":"","avatar_big":"http://v.hing.bding.com/sys/portrait/item/20160606
s://v.hing.bding.com/sys/portrait/item/20160606.jpg?
20160606","last_login_time":"16788888","incomplete_user":0,"user_source":0,"displayname":"","active":true,"cloud_li
a","cloud_upload":true))
```

图 5 挖掘的百度的一个 JSONP 漏洞

5 总结

本文针对 JSONP 漏洞手动挖掘耗时耗力、挖掘不全面的不足首次提出了基于 Chrome 插件的 JSONP 自动挖掘方法,该方法能高效、自动全面地挖掘网站中存在的 JSONP 漏洞,通过修复挖掘出的 JSONP 漏洞可减少用户信息的泄露,从而大幅降低由信息泄露造成的财产损失及社会危害。

参考文献:

[1] 陈腊梅,李为,程振林,等. AJAX 跨域访问的研究与应用[J]. 计算机工程与设计,2008(22):93-96.  
[2] RUDERMAN Jesse . Same Origin Policy for JavaScript [EB/OL]. 2011. [https://developer.mozilla.org/zh-TW/docs/Web/JavaScript/Same\\_origin\\_policy\\_for\\_JavaScript](https://developer.mozilla.org/zh-TW/docs/Web/JavaScript/Same_origin_policy_for_JavaScript)  
[3] Ozses S Ergul S. Cross-domain communications with JSONP [R]. 2009. <http://www.ibm.com/developerworks/web/library/wa-aj-jsonp1/>  
[4] 党寿江,王劲林,曾学问,等. JSONP 研究及其在 IPTV 门户系统中的应用[J]. 微计算机信息,2010,10 (3):183-185.  
[5] 刘耀钦. 基于 HTML5 跨域通信技术的客户端数据同步机制研究[J]. 现代计算机(专业版),2015(11):65-68  
[6] 何良,方勇,方昉,等. 浏览器跨域通信安全技术研究[J]. 信息安全与通信保密,2013(4):59-61.  
[7] 佟大柱. 网络环境下的个人信息安全与保护探讨[J]. 网络安全技术与应用,2012(8):18-20.