30
/
11
2018

84
期

KNOW IT THEN HACK IT

跨域资源共享(CORS)漏洞详解

难度系数：★★★☆☆

本期大咖

聂心明丨n0tr00t团队

个人简介：亚信安全软件工程师，n0tr00t团队成员
Kcon 2018《动态审计Python代码》演讲者
擅长php，python代码审计，渗透测试，代码审计，二进制入门级选手，在先知等知名安全论坛发表过多篇帖子，热爱技术和分享知识

个人博客：https://blog.csdn.net/niexinming

内容目录

大咖面对面

该信息安全技术公益讲座由漏洞银行方主办
每周五晚20:00，业内大咖与你零距离分享
答疑解惑 | 资源交换 | 剖析动态 | 认知升级

众多专家与你我共同扬帆，畅游知识海洋
加入我们的技术社群（Q群 598562771）

BUGBANK

漏洞银行 | BUGBANK

2018

大咖面对面

跨域资源共享(CORS)漏洞详解

n0tr00t团队

主讲：聂心明

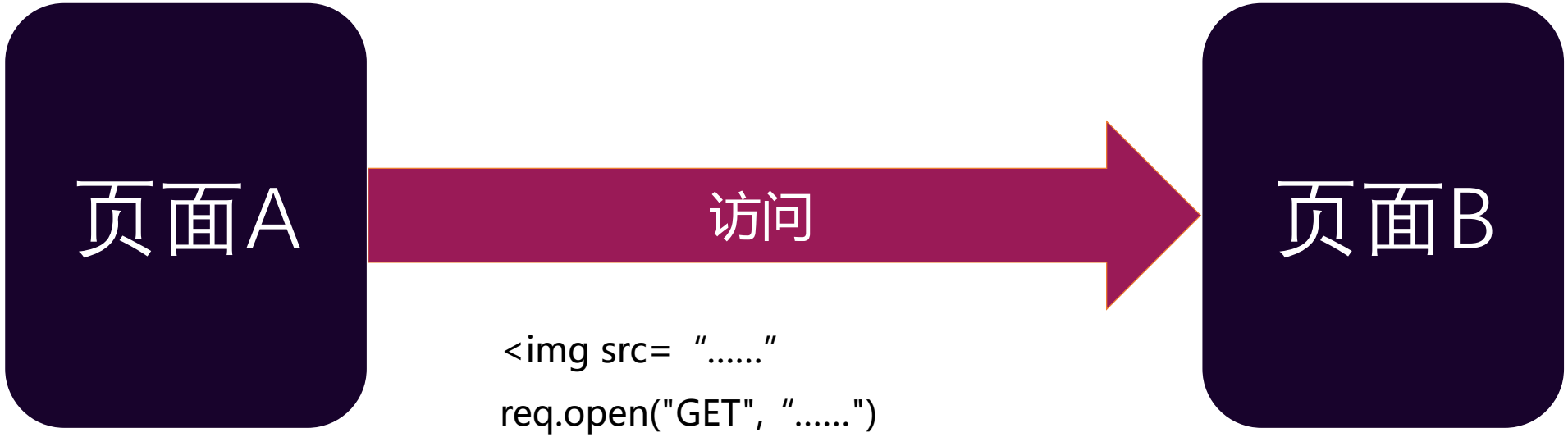
漏洞银行官网： www.bugbank.cn 大咖博客：https://blog.csdn.net/niexinming

参与讲座丨现场答疑丨后续交流 漏洞银行技术社群：598562771（Q群号）

漏洞银行 BUGBANK.CN | n0tr00t 安全团队

目录

浏览器同源策略



浏览器同源策略

下表给出了相对http://store.company.com/dir/page.html同源检测的示例:

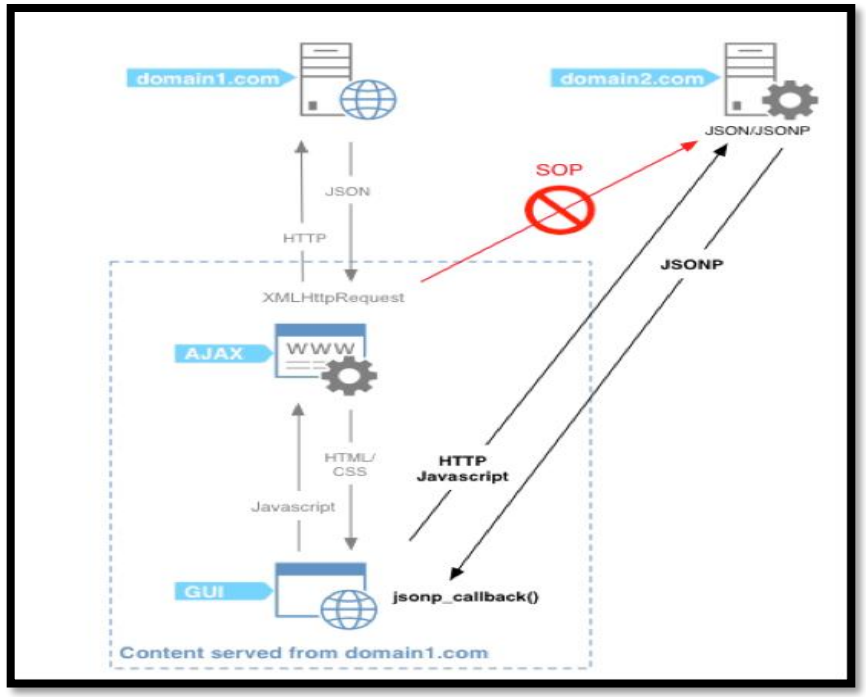
URL	结果	原因
http://store.company.com/dir2/other.html	成功	
http://store.company.com/dir/inner/another.html	成功	
https://store.company.com/secure.html	失败	不同协议（https和http）
http://store.company.com:81/dir/etc.html	失败	不同端口（81和80）
http://news.company.com/dir/other.html	失败	不同域名（news和store）

两种跨域的方法

- 1.JSONP跨域请求
- 2.CORS跨域请求

两种跨域的方法

1、利用JSONP可以跨域



两种跨域的方法

1、利用JSONP跨域

加载远程js，可以把远程js中数据带进来

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/x
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
    <script type="text/javascript">
    var localHandler = function(data){
        alert('我是本地函数，可以被跨域的remote.js文件调用，远程js带来的数据是: ' + data.result);
    };
    </script>
    <script type="text/javascript" src="http://remoteserver.com/remote.js"></script>
</head>
<body>

</body>
</html>
```

复制

两种跨域的方法

1、利用JSONP跨域

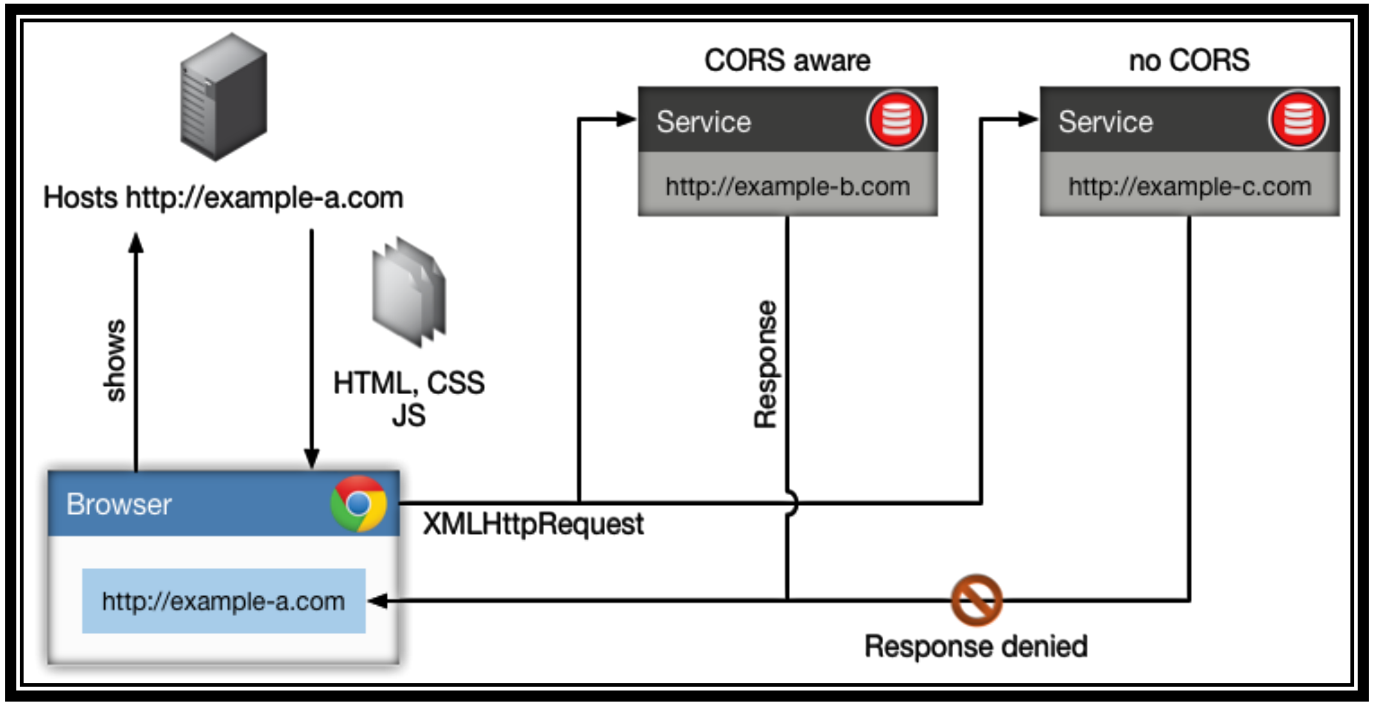
利用动态生成的方式

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/x
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
    <script type="text/javascript">
    // 得到航班信息查询结果后的回调函数
    var flightHandler = function(data){
        alert('你查询的航班结果是：票价 ' + data.price + ' 元，' + '余票 ' + data.tickets + ' 张。');
    };
    // 提供jsonp服务的url地址（不管是什么类型的地址，最终生成的返回值都是一段javascript代码）
    var url = "http://flightQuery.com/jsonp/flightResult.aspx?code=CA1998&callback=flightHandler";
    // 创建script标签，设置其属性
    var script = document.createElement('script');
    script.setAttribute('src', url);
    // 把script标签加入head，此时调用开始
    document.getElementsByTagName('head')[0].appendChild(script);
    </script>
</head>
<body>

</body>
</html>
```

两种跨域的方法

2、利用CORS跨域



两种跨域的方法

CORS安全问题

返回报文头部的Access-Control-Allow-Origin根据请求报文Origin生成

```
GET /userinfo/ HTTP/1.1
Host: 192.168.34.134:5000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://10.22.6.250:8081/cors-poc
Origin: http://10.22.6.250:8081
Cookie:
```

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 1368
Access-Control-Allow-Origin: http://10.22.6.250:8081
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET,POST,PUT,DELETE,OPTIONS
Access-Control-Max-Age: 21600
Vary: Cookie
Server: Werkzeug/0.14.1 Python/2.7.12
```

Ps:只要页面产生跨域请求，那浏览器就会在请求报文中自动带上

两种跨域的方法

CORS安全问题

返回报文头部的Access-Control-Allow-Credentials为true时

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 1368
Access-Control-Allow-Origin: http://10.22.6.250:8081
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET,POST,PUT,DELETE,OPTIONS
Access-Control-Max-Age: 21600
Vary: Cookie
```

这表明Cookie可以包含在请求中，一起发给服务器

两种跨域的方法

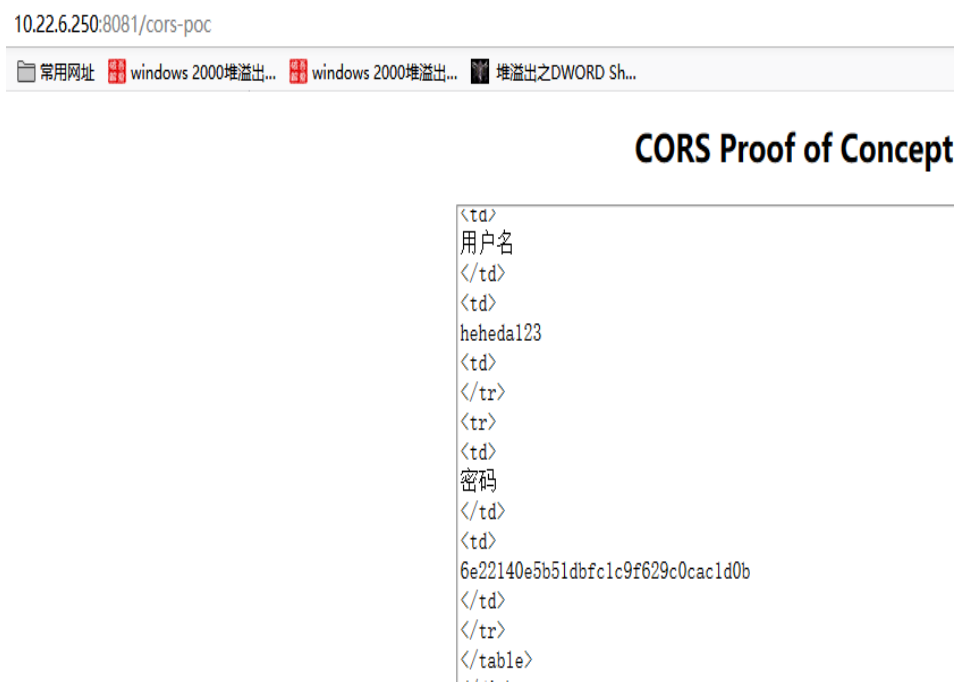
CORS安全问题

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 1368
Access-Control-Allow-Origin: http://10.22.6.250:8081
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET,POST,PUT,DELETE,OPTIONS
Access-Control-Max-Age: 21600
Vary: Cookie
Server: Werkzeug/0.14.1 Python/2.7.12
```

如果Access-Control-Allow-Origin可控，且Access-Control-Allow-Credentials为true，那么就可以利用一个可控的网站来窃取一个人的个人隐私信息

两种跨域的方法

CORS安全问题



盗取

两种跨域的方法

登陆后

收到

查看更多请点击 http://10.22.6.250:8081/cors-poc

点击后

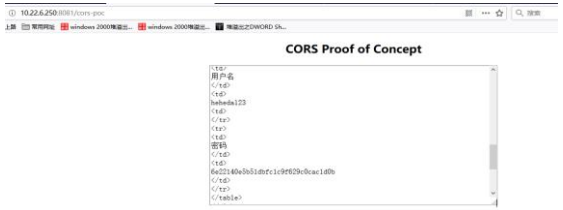
钓鱼邮件

```
GET /cors-poc HTTP/1.1
Host: 10.22.6.250:8081
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;
```

Ajax异步发出请求

```
GET /userinfo/ HTTP/1.1
Host: 192.168.34.134:5000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/6
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://10.22.6.250:8081/cors-poc
Origin: http://10.22.6.250:8081
Cookie:
session=.eJw9kE9rgzAYxr_KyLkHE-1F6GE1rWTwJnQkhjcXoatVo91AW2pT-t3netj1gef3_HmQ6jzWUOvyy3
```

拿到192上的数据



盗取过程

CORS漏洞与CSRF漏洞

相同点：

1.都要借助第三方网站
2.都要借助ajax的异步过程
3.一般都需要用户登陆

CORS漏洞与CSRF漏洞

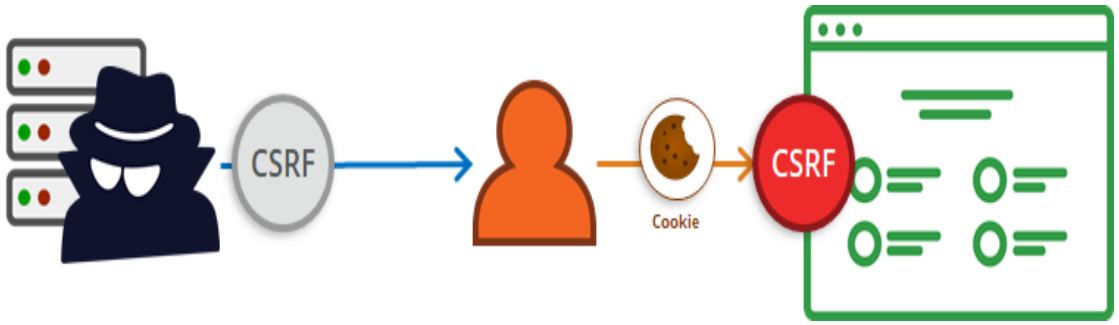
不同点：

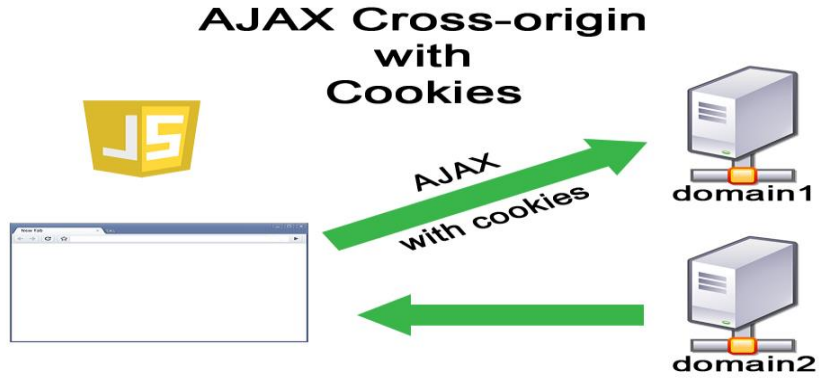
1.第三方网站可以利用CORS漏洞读取到受害者的敏感信息

2.第三方网站可以利用csrf漏洞可以替受害者完成诸如转账等敏感操作

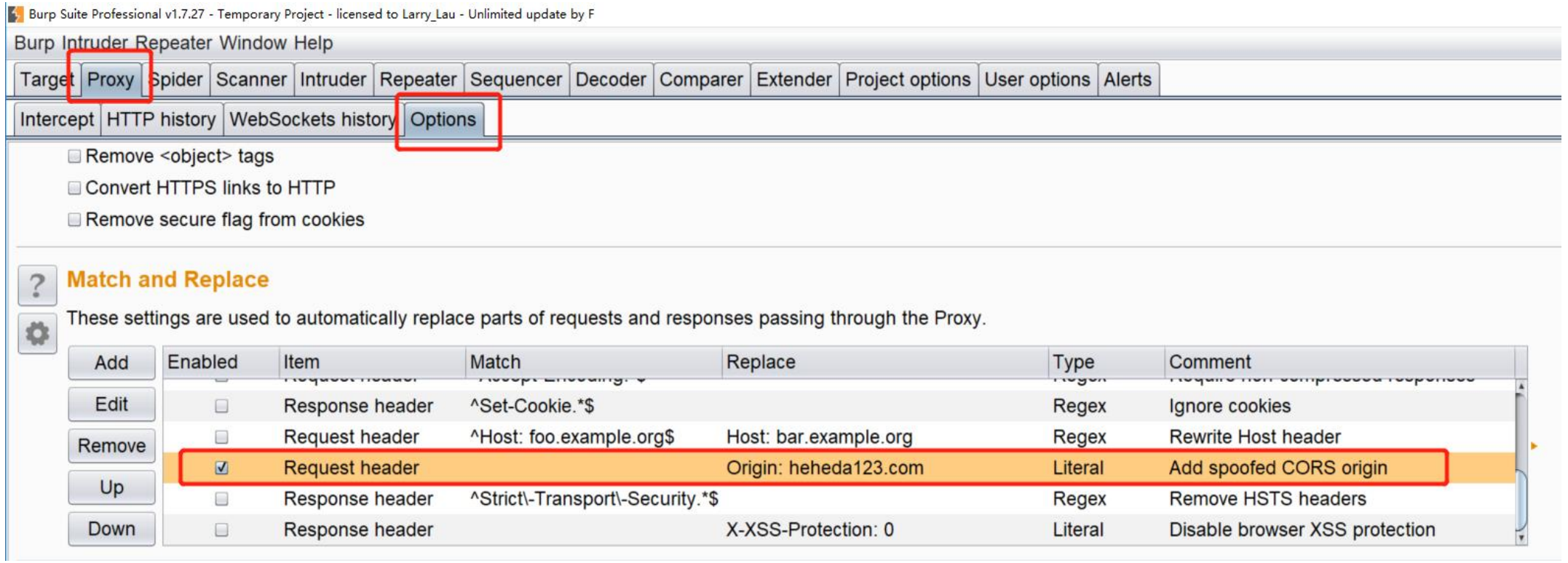
3.一般有CORS漏洞的地方都有csrf漏洞

CORS漏洞与CSRF漏洞

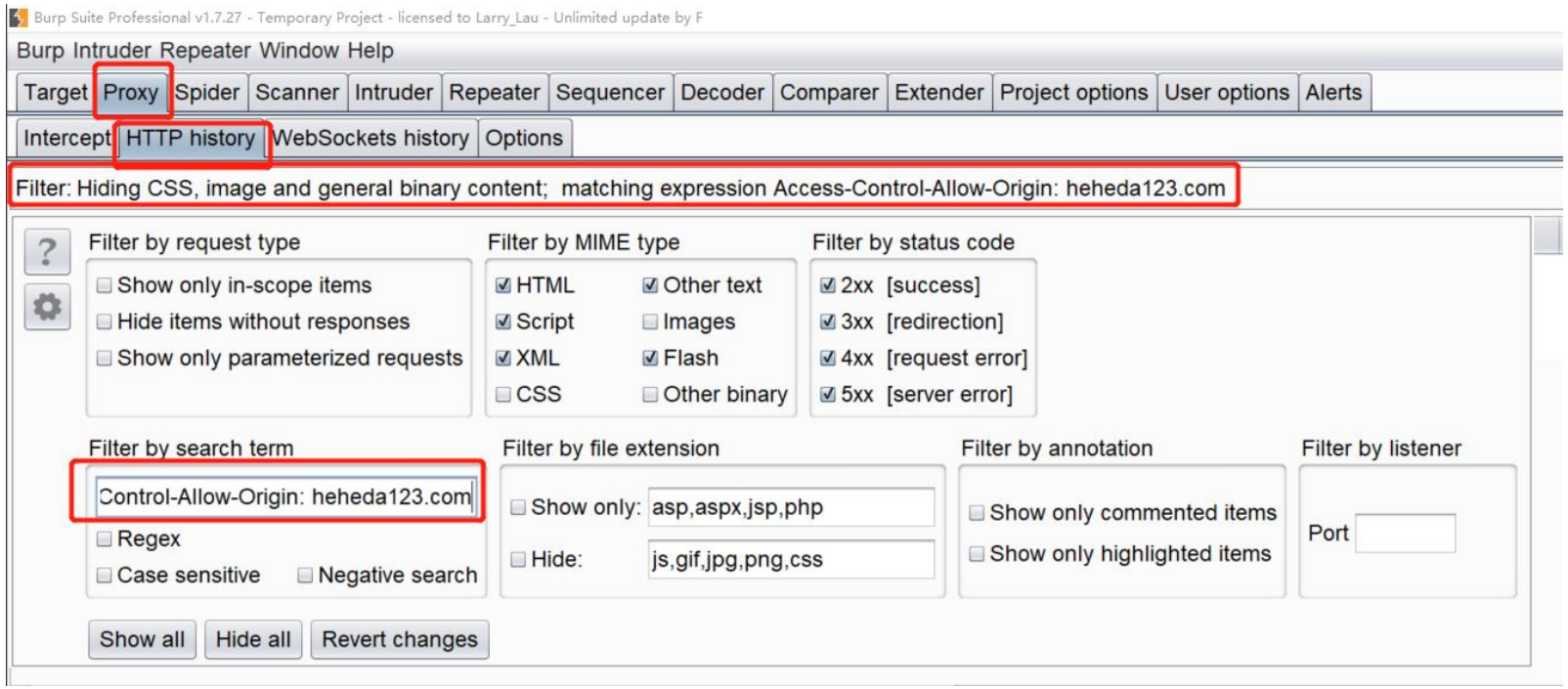




简单的检测漏洞方法


Burp Suite Professional v1.7.27 - Temporary Project - licensed to Larry_Lau - Unlimited update by F
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Remove <object> tags
Convert HTTPS links to HTTP
Remove secure flag from cookies
Match and Replace
These settings are used to automatically replace parts of requests and responses passing through the Proxy.
Add
Edit
Remove
Up
Down
Enabled Item Match Replace Type Comment
Response header ^Set-Cookie.*$ Regex Ignore cookies
Request header ^Host: foo.example.org$ Host: bar.example.org Regex Rewrite Host header
Request header Origin: heheda123.com Literal Add spoofed CORS origin
Response header ^Strict\-Transport\-Security.*$ Regex Remove HSTS headers
Response header X-XSS-Protection: 0 Literal Disable browser XSS protection

简单的检测漏洞方法


Burp Suite Professional v1.7.27 - Temporary Project - licensed to Larry_Lau - Unlimited update by F
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Filter: Hiding CSS, image and general binary content; matching expression Access-Control-Allow-Origin: heheda123.com
Filter by request type
Show only in-scope items
Hide items without responses
Show only parameterized requests
Filter by MIME type
HTML
Other text
Script
Images
XML
Flash
CSS
Other binary
Filter by status code
2xx [success]
3xx [redirection]
4xx [request error]
5xx [server error]
Filter by search term
Control-Allow-Origin: heheda123.com
Regex
Case sensitive
Negative search
Filter by file extension
Show only: asp,aspx,jsp,php
Hide: js,gif,jpg,png,css
Filter by annotation
Show only commented items
Show only highlighted items
Filter by listener
Port
Show all
Hide all
Revert changes

简单的检测漏洞方法

Host	Method	URL
http://192.168.34.134:5000	GET	/userinfo/
http://192.168.34.134:5000	GET	/
http://192.168.34.134:5000	GET	/userinfo/

```
GET /userinfo/ HTTP/1.1
Host: 192.168.34.134:5000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://10.22.6.250:8081/cors-poc
Origin: http://10.22.6.250:8081
Cookie:
session=.eJw9kE9rgzAYxr_KyLkHE-1F6GE1rWTwJnQkhjcXoatVo91AW2pT-t3netj1gef3_HmQ6jzWUOvyy3itV6TqTiR_kLcjyYnz2wFOQ4EbhlaslcXUcUyhgAx86dFjJrW5ARd3COa
```

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 1368
Access-Control-Allow-Origin: http://10.22.6.250:8081
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET,POST,PUT,DELETE,OPTIONS
Access-Control-Max-Age: 21600
Vary: Cookie
Server: Werkzeug/0.14.1 Python/2.7.12
Date: Thu, 11 Oct 2018 06:57:19 GMT
```

常见漏洞点

1.互联网厂商的api接口

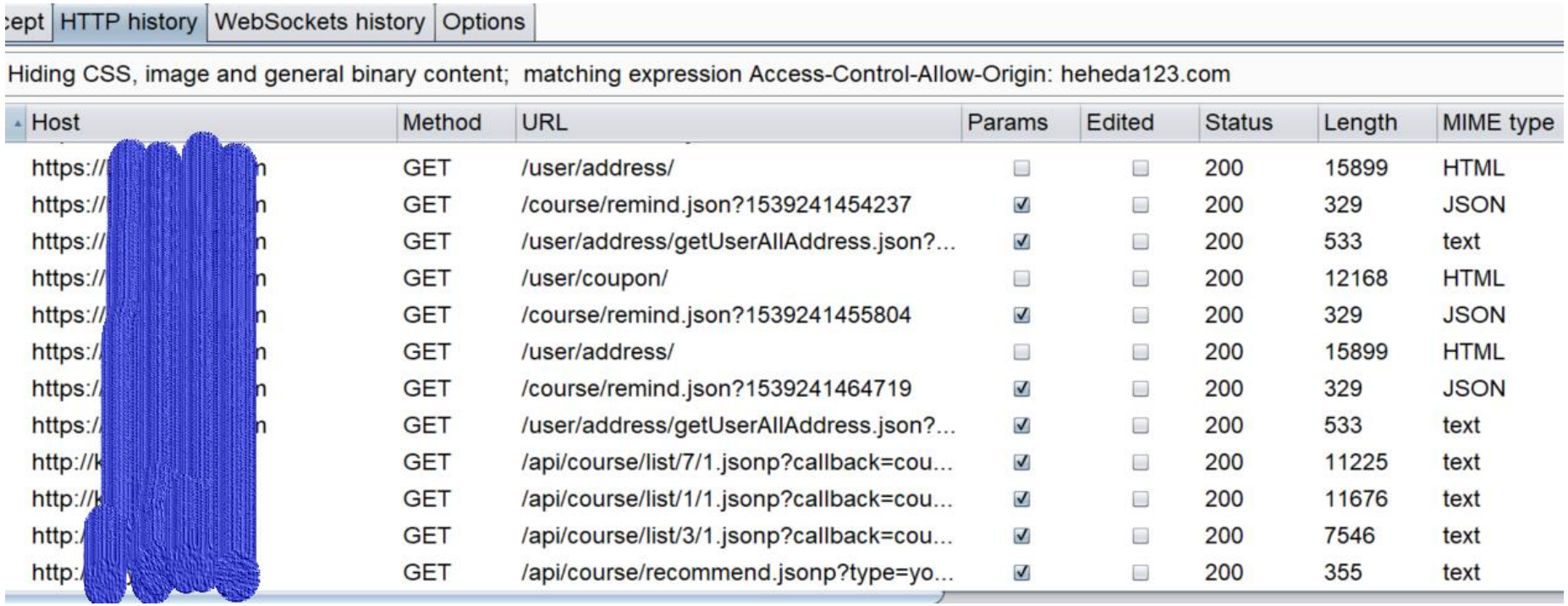
2.聊天的程序的api接口

3.app的api（不过有一些请求需要带有一些额外的请求头，利用起来比较困难）

4.区块链厂商

某互联网公司一个有趣的漏洞

当我访问某精品课的时候发现，某子站整站都有问题



构造poc获取用户收货地址

点我的链接获取你的敏感信息

Referer的检查

正常Referer

非youdao和163域Referer返回403页面

利用某子域的xss绕过Referer的检查

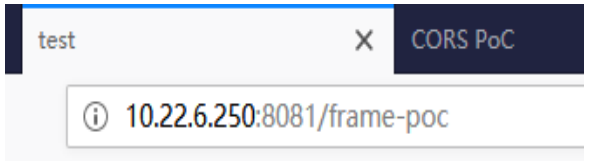
跨域资源共享(CORS)漏洞详解

于是

受害者

登陆

然后不小心访问到攻击者网站



隐藏的框架偷偷加载,触发xss

http://xxx.com/a.s?c=<xss>

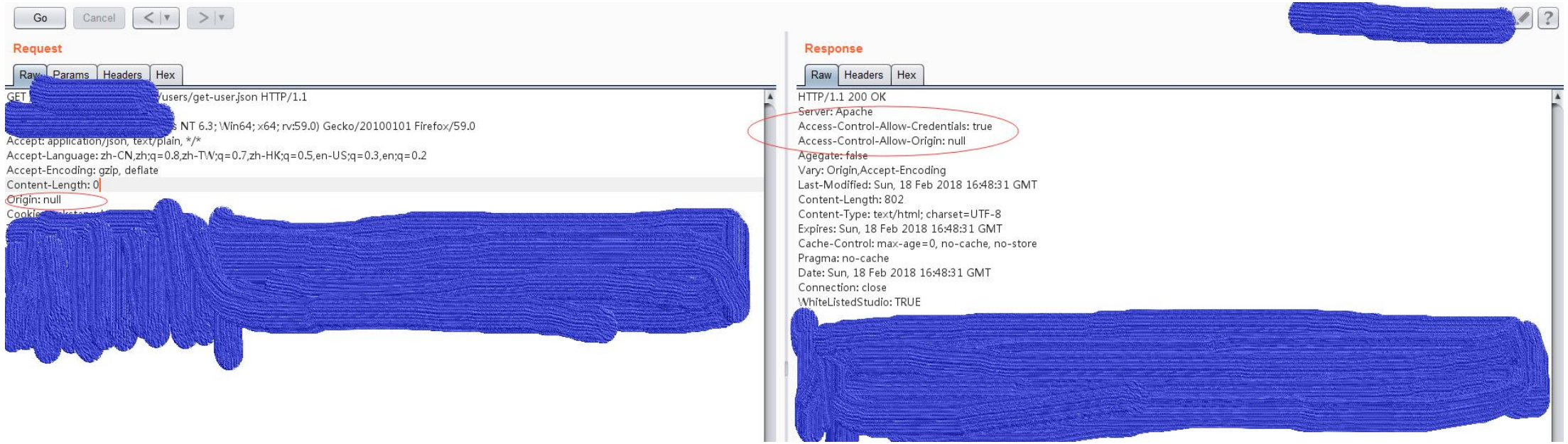
加载xss脚本

窃取受害者数据

结果


10.22.6.250:8081/frame-poc
搜索
火狐官方站点
新手上路
常用网址
windows 2000堆溢出...
windows 2000堆溢出...
堆溢出之DWORD Sh...
新闻
hao123
地图
来自 http://[redacted]的页面说：
{"result":[{"area":"东城区","zip":"1234","default":true,"address":"aaaaaaaabbbbbbbbb","province":"北京市","city":"北京市","mobile":"15011111111","name":"cpu0847723","id":1,"time":1538211816664}],"success":true}
确定

Hackerone一个价值500美金的漏洞



Hackerone一个价值500美金的漏洞

- CORS的规范中还提到了“NULL”源。触发这个源是为了网页跳转或者是来自本地HTML文件。
  目标应用可能会接收“null"源，并且这个可能被测试者（或者攻击者）利用，任何网站很容易使用沙盒iframe来获取”null“源

Hackerone一个价值500美金的漏洞

为什么服务器端会有这样的漏洞

- 1.开发人员开发，调试，测试代码一般都在本地
- 2.有时候他们会调用线上服务器数据
- 3.所以这样的问题很隐蔽也很常见

防御

- 1.不要配置“Access-Control-Allow-Origin”为通配符“*”，而且更重要的是，要严格效验来自请求数据包中的“Origin”的值。当收到跨域请求的时候，要检查“Origin”的值是否是一个可信的源，还要检查是否为null
- 2.避**免使用“Access-Control-Allow-Credentials: true”**
- **3.减少Access-Control-Allow-Methods所允许的方法**

还没看够？来了解更多技术干货

漏洞银行直播间：https://www.bugbank.cn/live/

直播资料丨社群伙伴丨听讲通知

QQ群号：327085041

也想当大咖？还不扫码报名

也可联系运营 QQ：2272924679

了解更多安全行业热点时事

行长叠报：BUG_BANK

漏洞银行 BUGBANK.CN | n0tr00t 安全团队