

Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心



跨域开发与安全

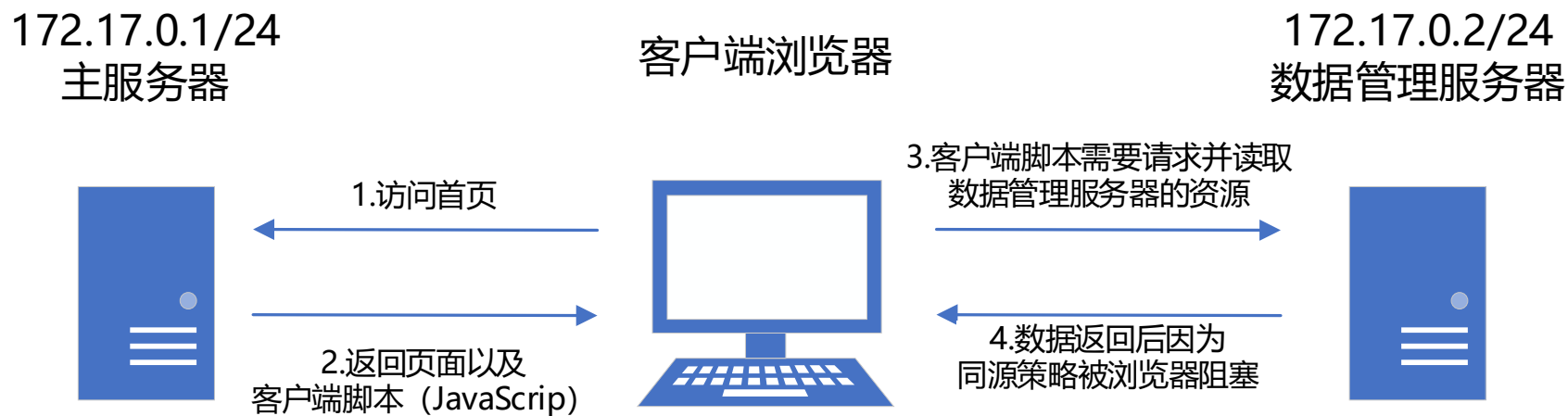
跨域开发与安全

硕士研究生 李橙

2020年09月13日

- 背景简介
- 预期收获
- 基础知识
- 跨域解决方案
- 跨域安全

- 项目中遇到的跨域开发问题



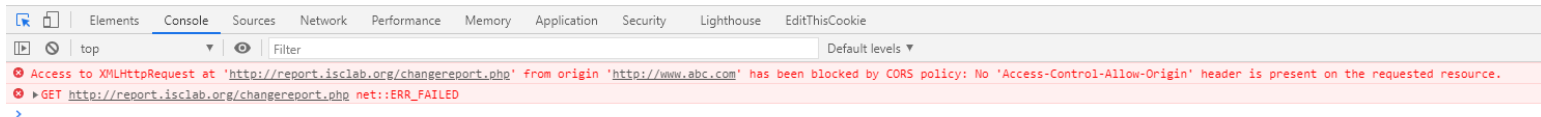
www.abc.com

客户端浏览器

report.isclab.org



```
<!DOCTYPE html>
<body>
  <script src="./jquery.js"></script>
  <script>
    $(document).ready(function() {
      $.get("http://report.isclab.org/queryweekly.php", function(data) {
        console.log(data);
      });
    })
  </script>
</body>
```



Access to XMLHttpRequest at 'http://report.isclab.org/changereport.php' from origin 'http://www.abc.com' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.

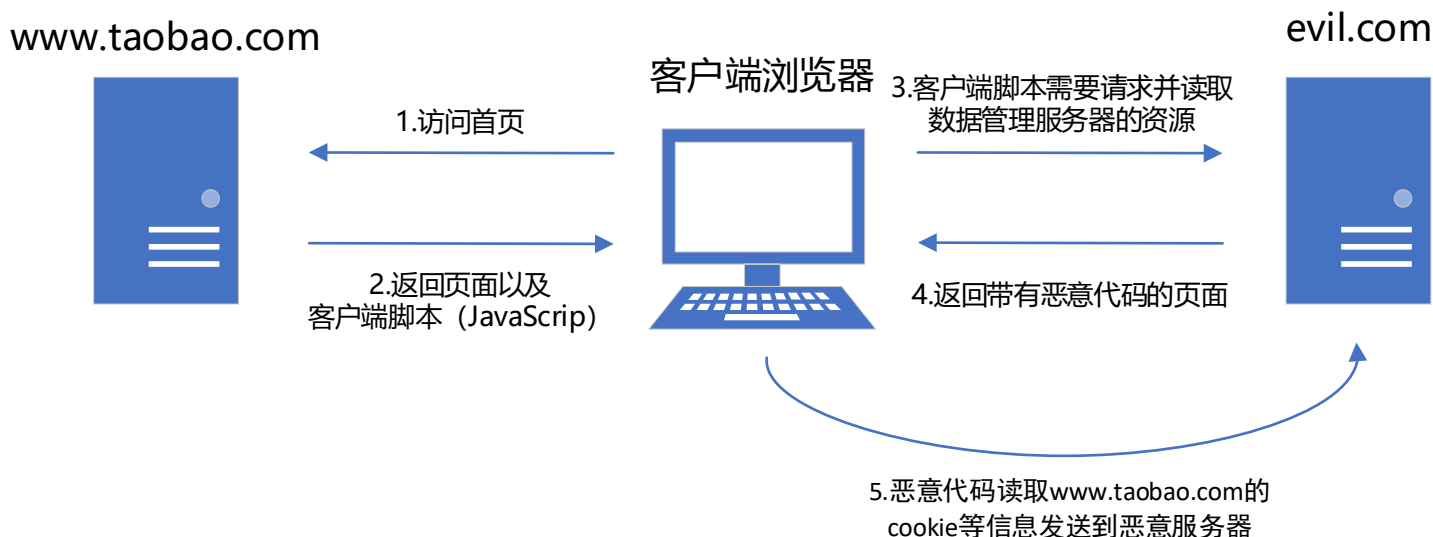
- 理解同源策略
- 了解跨域方案
- 了解跨域中的安全问题

- URI格式
- [协议名]://[用户名]:[密码]@[服务器地址]:[服务器端口号]/[路径]?[查询字符串]#[片段ID]
- 同源的定义
 - 如果两个页面的**协议**，**端口**和**域名**都相同，则两个页面具有相同的源。

URL	结果	原因
http://www.isclab.org.cn/dir1/other.html		
http://www.isclab.org.cn/dir2/index.php	同源	
http://www.isclab.org/dir1/other.html	不同源	域名不同
https://www.isclab.org.cn/dir1/other.html	不同源	协议不同
http://www.isclab.org.cn:8080/dir1/other.html	不同源	端口不同

- 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（Cookie、DOM树）。所以www.xyz.com下的js脚本采用ajax读取www.abc.com里面的文件数据是会被拒绝的。
- 同源政策的目的是，为了保证用户信息的安全，防止恶意的网站窃取数据。



- 同源策略的限制范围
 - Cookie、LocalStorage和IndexedDB无法读取
 - DOM无法获取
 - AJAX请求不能读取（ **请求可以成功发送！** ）
- 允许跨域加载的标签
 - 、<link>、<script>、<iframe>、<a>、<video>、<audio>

- Cookie

属性	描述
Name	设置要保存的 Key
Value	设置要保存的 Value
Domain	生成该 Cookie 的域名，如Domain="xyz.com"
Path	该 Cookie 是在当前的哪个路径下生成的，如Path=/
HttpOnly	如果Cookie中设置了HttpOnly属性，那么通过js脚本将无法读取到cookie信息
Secure	如果设置了这个属性，那么只会在 SSH 连接时才会回传该 Cookie
SameSite	如果链接来自外部站点，浏览器不会将cookie 添加到已通过身份验证的网站

- Cookie跨域方案

- 相同顶级域名，不同源的情况。

设置Cookie的Domain字段为顶级域名，则所有属于该顶级域名的其他域名都可以访问该Cookie。

例：当URL为www.xyz.com/index.php页面使用如下php代码设置Cookie时，*.xyz.com域下的所有页面都能够获取该Cookie信息。

1. `setcookie("secret", "heiheihei", 0, "/", "xyz.com");`
2. `header("Set-Cookie: secret=heiheihei; domain=xyz.com; path=/");`

- DOM跨域方案

比较典型的例子是iframe与不同源的父窗口之间无法通信。

- 当一级域名相同，二级域名不同时，可以通过同时设置document.domain为一级域名来进行通信。
- 使用location.hash和代理页面进行通信。



- DOM跨域方案
 - 使用window.name通信，当页面刷新时，该值依然不变（即使是非同域），并且支持较长的值（2MB）

book.xyz.com/index.html

- 1.先设置iframe的src为www.xyz.com/data.php
- 3.当iframe加载完后，改变iframe的src为book.xyz.com/proxy.html
- 4.删除iframe节点，否则会一直循环

www.xyz.com/data.php

2. window.name = data;

- 跨文档通信API `window.postMessage`
 - `otherWindow.postMessage(message, targetOrigin);`
 - 首先需要获取window对象，再调用postMessage方法，接受页面需要定义事件监听回调函数，用来处理数据。
 - 以iframe为例
 - 父窗口向子窗口通信，使用`iframeElement.contentWindow.postMessage`发送数据
 - 子窗口向父窗口通信，使用`parent.window.postMessage`发送数据

- JSONP

- JSONP是服务器与客户端跨源通信的常用方法。最大特点就是简单适用，老式浏览器全部支持，服务器改造非常小。

book.xyz.com

```
<script src='http:www.xyz.com/ip?callback=getIp'>
</script>

<script>
function foo(data) {
  console.log('Your public IP address is: ' + data.ip);
};
</script>
```

www.xyz.com/ip?callback=getIp

1. script调用外部接口



2.JSON格式数据
foo({'ip':'8.8.8.8'});



- WebSocket，一种全双工的通信协议，该协议不实行同源策略，只要服务器支持就可以进行跨源通信
- CORS（Cross-Origin Resource Sharing，跨源资源共享），是跨源AJAX的根本解决方法，因为JSONP只能发送GET请求，CORS这允许任何类型的请求，并且支持自定义配置。
 - 浏览器将CORS请求分为简单请求和非简单请求

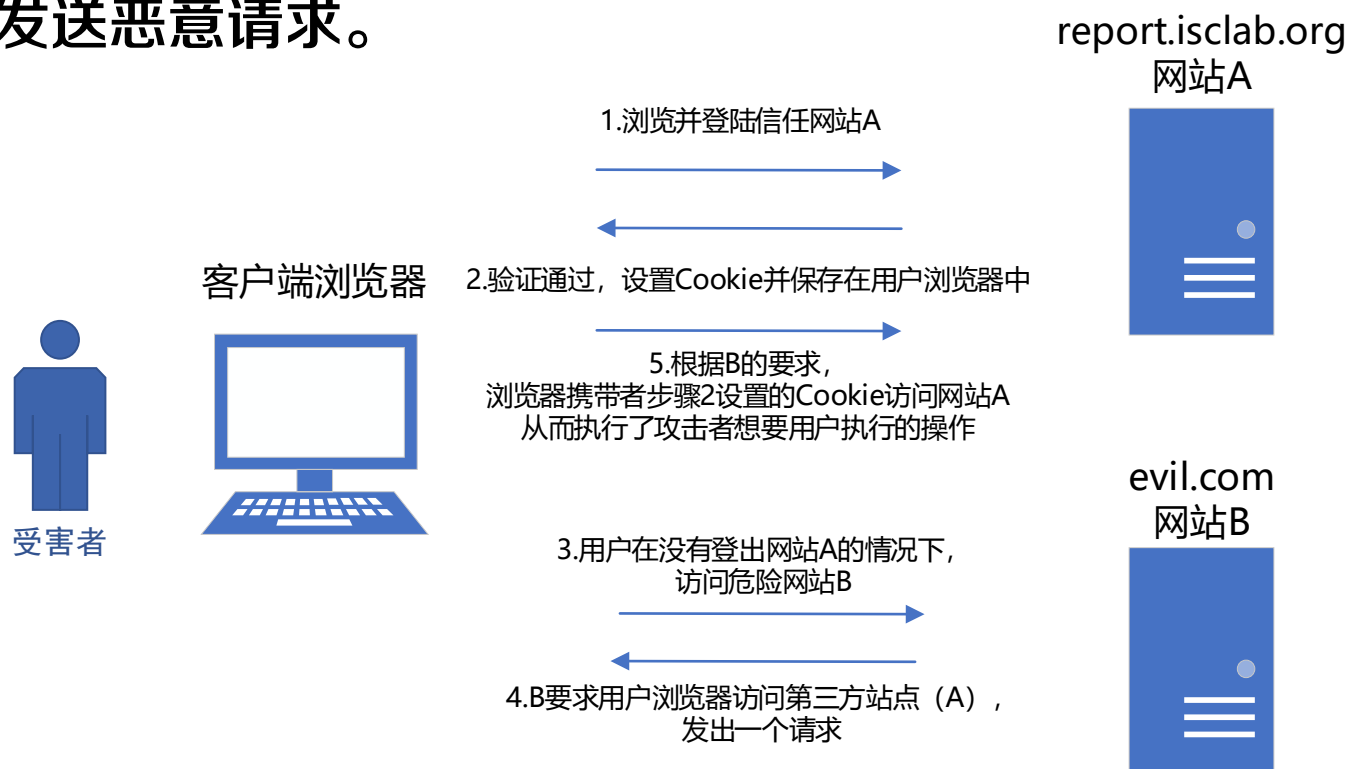
- CORS简单请求
 - 请求方法为HEAD、GET、POST其中之一
 - Content-Type只限于application/x-www-form-urlencoded、multipart/form-data、text/plain三个值
 - 对于简单请求，浏览器直接发出CORS请求，并且在HTTP头中加入一个Origin字段，用来表示该请求是从那个源发起的
 - 服务端通过设置Access-Control-Allow-Origin、Access-Control-Allow-Credentials、Access-Control-Expose-Headers等字段来判断请求是否合法

- 同源策略安全
 - 不同客户端浏览器对协议的实现不同，导致安全漏洞。
- 例如CVE-2015-7188火狐浏览器SOP绕过中，攻击者通过构造特殊的URL，并在攻击者自己控制的来自37.187.18.85的网页中发起跨域请求。
http://37.187.18.85\0B\uFF20translate.google.com/fx_sop_bypass/FlashTest.swf?url=http://translate.google.com/manager/website/
- 先通过\0B让Firefox认为这个请求是请求37.187.18.85本身的内容，再通过类似@字符的Unicode字符@(\uFF20)让浏览器认为@之前的字符都是translate.google.com的账号和密码，从而返回translate.google.com的网页内容，实现绕过SOP。

- 同源策略安全
 - 在Java6, 7中, 如果两个域名解析到相同的IP, 则会认为他们同源。假设我们有attacker.com和victim.com, 两者都共享主机123.123.123.123。攻击者attacker.com可以在自己控制的域名下上传一个jar文件来访问victim.com的内容。

- CSRF

- CSRF攻击的全称是跨站请求伪造 (cross site request forgery), 是一种对网站的恶意利用。简单点讲就是, 恶意网站(攻击者)盗用了你的身份, 以你的名义向信任网站发送恶意请求。



- CSRF实例
 - 1.使用手机浏览器登录report.isclab.org
 - 2.进入评分界面
 - 3.使用相同的浏览器，新建标签后访问
<http://www.isclab.org.cn/csrf.html>
- 防御方法
 - 读取referrer字段
 - Cookie设置SameSite字段
 - Strict 任何情况下，不跨域发送cookie
 - Lax 使用连接跳转且为GET请求时，发送cookie
 - 使用csrf token

- CORS安全
 - Access-Control-Allow-Origin配置不当
 - HTTP响应头修改

- Document.Domain安全
 - 现在很多网站把不同的之业务放在不同的子域下，比如：
www.isclab.org、iscc.isclab.org、ftp.isclab.org
并且通常为了业务之间方便通信，使用一个proxy.html或者直接
直接在页面中设置document.Domain为一级域名。
 - 此时如果在某个子域中发现XSS漏洞，则将会危害到其他子域的安全。



谢谢！

大成若缺，其用不弊。大盈
若冲，其用不穷。大直若屈。
大巧若拙。大辩若讷。静胜
躁，寒胜热。清静为天下正。

