

实验一 JSONP

实验内容：

a)实现三个主机

web.cybersecuiry.seu.edu(10. 0. 0.2, host1),

time.cybersecurity.seu.edu (10. 0. 0. 3, host2),

jsonp.cybersecurity.seu.edu(10. 0. 0. 4,host3)

b)在 time. cybersecurity. seu. edu 上实现：

i. /api/date 接口，返回形式为{date: 124566334}这样的 json 数据。

ii. /api/datecors 接口， 返回形式为{date: 124566334}这样的 json 数据,并设置 CORS 头部字段。

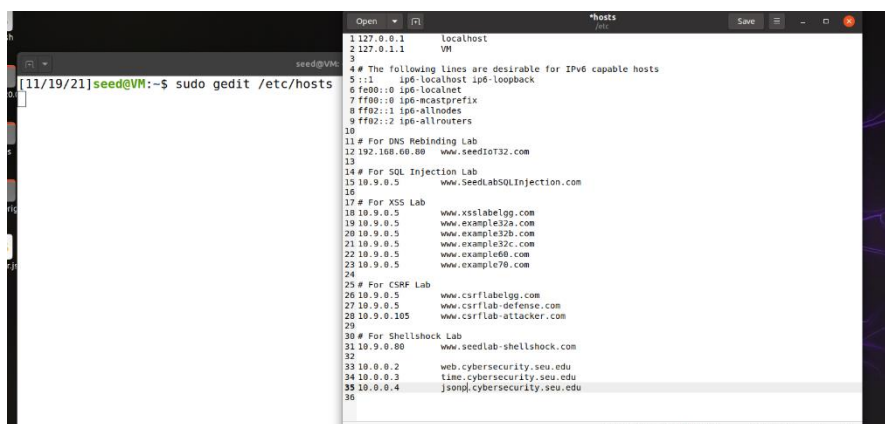
iii. /api/ jsonpdate 接口，返回形式为 jsonp 格式的数据。

c)在 web.cybersecuiry.seu.edu 下实现一个页面，在页面中通过 js 代码读取 time.cybersecurity.seu.edu 的接口数据，分别测试在 time.cybersecurity.seu.edu 中设置和未设置 CORS 头的情况下，web.cybersecuiry.seu.edu 读取接口数据的情况，提供读取成功和未读取成功模式下的截图。

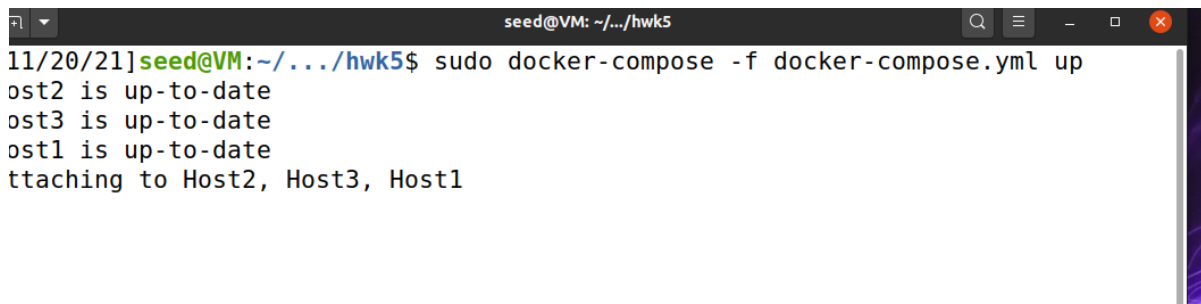
d)在 jsonp.cybersecurity.seu.edu 下实现一个页面，测试在 time.cybersecurity.seu.edu 中未设置 CORS 头的情况下，在页面中通过回调 js 代码读取 time.cybersecurity.seu.edu 的接口数据的方法，提供读取成功模式下的截图。

实验步骤：

1. 向主机(运行 docket 的机器)的 hosts 文件中添加以下条目，以设置对应的 ip 地址和域名。



编写 host 主机中的服务器代码，运行命令 “sudo docker-compose -f docker-compose.yml up” 启动容器，实现三个主机。



2.设置 /api/date 接口，相关代码如下，返回 json 格式的时间数据。

```

11
12 app.get('/api/date', (req, res) => {
13   res.send({ date: Date.now() })
14 })
15

```

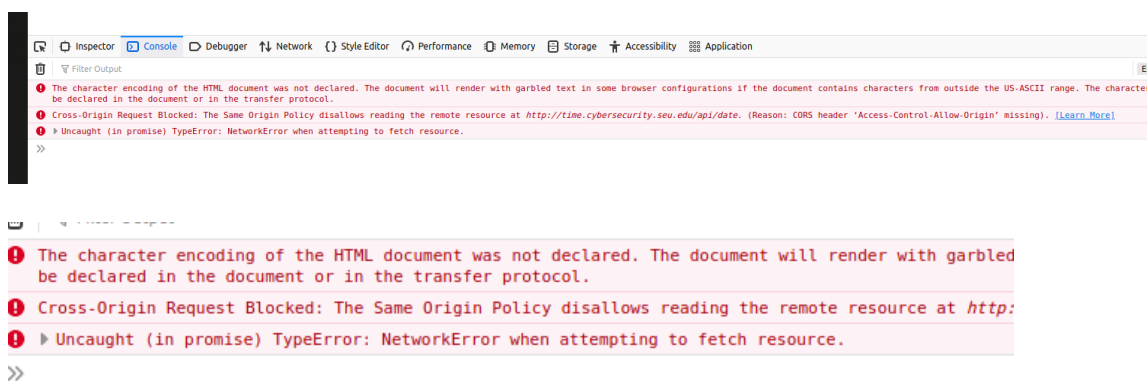
在 host1 主机的 html 页面中添加如下代码，通过 js 代码读取 time.cybersecurity.seu.edu 的接口数据，此时未设置 CORS 头。

```

15 async function get_date() {
16   const res = await fetch('http://time.cybersecurity.seu.edu/api/date')
17   const data = await res.json()
18   console.log(data)
19 }
20 get_date();

```

运行打开页面，由于存在跨域请求，发现不能获取到相关数据。



3. 设置 /api/datecors 接口，相关代码如下，返回 json 格式的时间数据。

```

16 app.get('/api/datecors', (req, res) => {
17   res.set('Access-Control-Allow-Origin', '*')
18   res.send({ date: Date.now() })
19 })

```

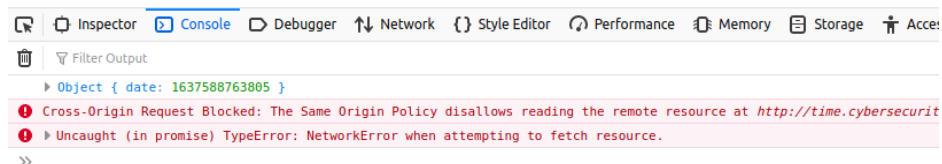
在 host1 主机的 html 页面中添加如下代码，通过 js 代码读取 time.cybersecurity.seu.edu 的接口数据，此时设置了 CORS 头。

```

9 async function get_corsdate() {
10   const res = await fetch('http://time.cybersecurity.seu.edu/api/datecors')
11   const data = await res.json()
12   console.log(data)
13 }
14 get_corsdate();

```

运行打开页面，发现获得了返回值。



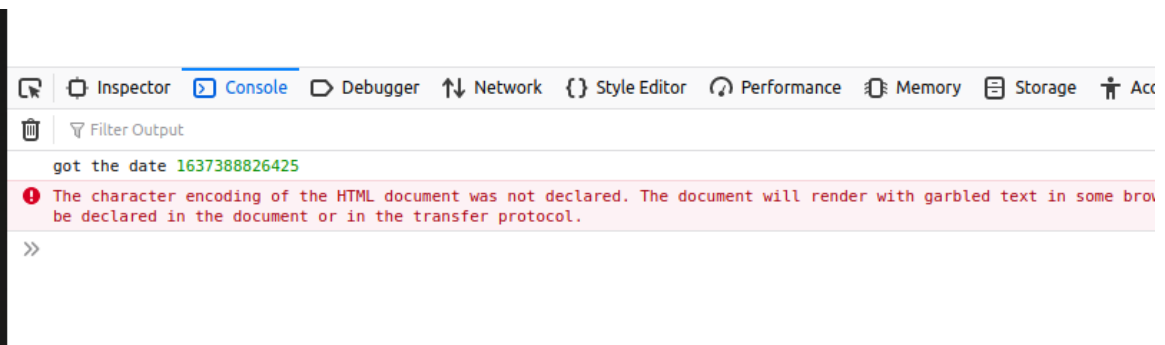
4. 设置 /api/jsonpdate 接口，相关代码如下，返回形式为 jsonp 格式的数据。

```
11
12 app.get('/api/jsonpdate', (req, res) => {
13   res.jsonp({ date: Date.now() })
14 })
15
```

在 host1 主机的 html 页面中添加如下代码，通过 js 代码读取 time.cybersecurity.seu.edu 的接口数据，且在未设置 CORS 头的情况下。

```
1 <script>
2   function handleTime (data) {
3     console.log('got the date', data.date)
4   }
5 </script>
6 <script src='http://jsonp.cybersecurity.seu.edu/api/jsonpdate?callback=handleTime'></script>
```

运行打开页面，发现获得了返回值。



实验结论：

当网页存在跨域请求时，会由于同源策略不能访问，设置 CORS 头和通过 JSONP 可以产生例外情况，实现了跨域请求并且获取到了数据。