

# Web 作业报告 JSONP

57118229 袁超然

## 1. 主机配置:

使用 Docker 配置三个主机:

```
docker-compose.yml
~/Desktop/hwk1
Save

1 version: '3.5'
2 services:
3   host1:
4     image: node
5     container_name: Host1
6     tty: true
7     command:
8       - /bin/bash
9       - -c
10      - |
11        cd data
12        node server.js
13     networks:
14       WSnetwork:
15         ipv4_address: 10.0.0.2
16     volumes:
17       - ./host1:/data
18       - ./node_modules:/data/node_modules
19
20   host2:
21     image: node
22     container_name: Host2
23     tty: true
24     command:
25       - /bin/bash
26       - -c
27       - |
28        cd /data
29        node server.js
30     networks:
31       WSnetwork:
32         ipv4_address: 10.0.0.3
33     volumes:
34       - ./host2:/data
35       - ./node_modules:/data/node_modules
36
37   host3:
38     image: node
39     container_name: Host3
40     tty: true
41     command:
42       - /bin/bash
43       - -c
44       - |
45        cd /data
46        node server.js
47     networks:
48       WSnetwork:
49         ipv4_address: 10.0.0.4
50     volumes:
51       - ./host3:/data
52       - ./node_modules:/data/node_modules
53
54   networks:
55     WSnetwork:
56       driver: bridge
57       ipam:
58         config:
59           - subnet: "10.0.0.0/24"
60
```

Hosts 文件配置:

```
10.0.0.2      web.cybersecurity.seu.edu
10.0.0.3      time.cybersecurity.seu.edu
10.0.0.4      jsonp.cybersecurity.seu.edu
```

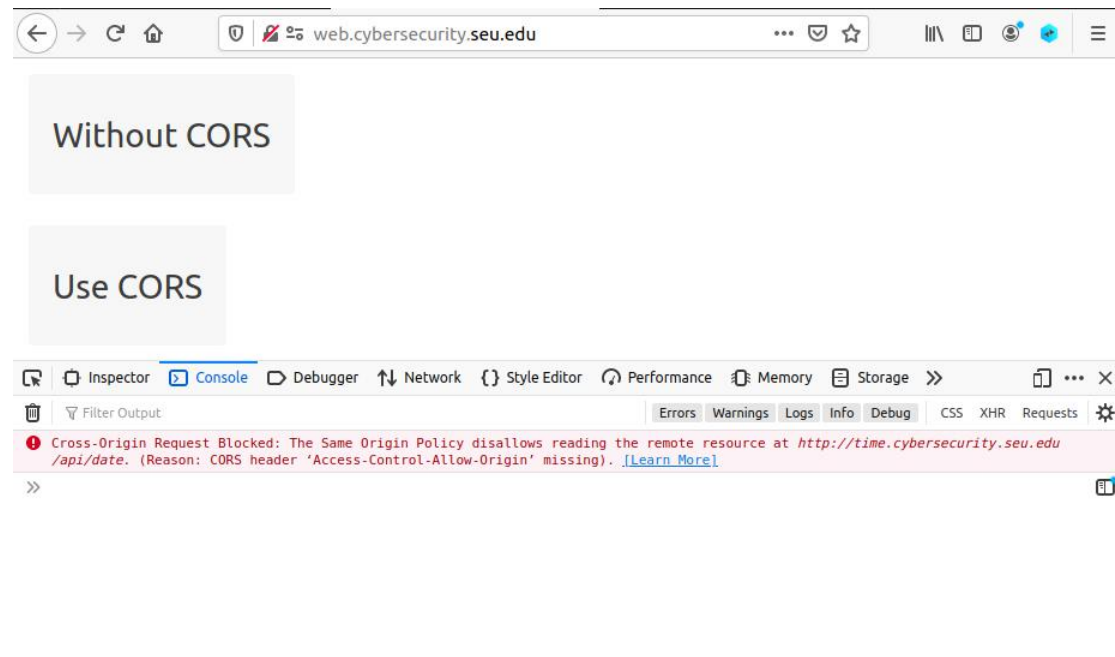
2. 在 time.cybersecurity.seu.edu 分别实现三个接口:  
其中, datecors 接口设置了 CORS 头部字段:

```
E: > websec_assignment1-main > hwk1 > host2 > JS server.js > ...
1  const express = require('express')
2  const { createReadStream } = require('fs')
3  const bodyParser = require('body-parser')
4  const app = express()
5
6  app.use(bodyParser.urlencoded({ extended: false }))
7  app.listen(80)
8  app.get('/', (req, res) => {
9    |   createReadStream('index.html').pipe(res)
10 | })
11 app.get('/api/date', (req, res) => {
12 |   res.send({ date: Date.now() })
13 | })
14 app.get('/api/datecors', (req, res) => {
15 |   res.set('Access-Control-Allow-Origin', '*')
16 |   res.send({ date: Date.now() })
17 | })
18 app.get('/api/jsonpdate', (req, res) => {
19 |   res.send(req.query.callback + '({ "date": ' + Date.now() + ' })')
20 | })
21
```

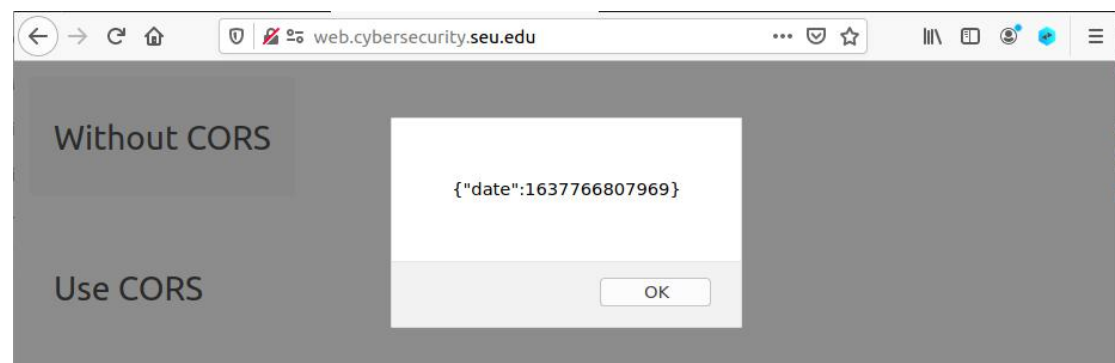
3. 在 web.cybersecurity.seu.edu 中读取 time.cybersecurity.seu.edu 的接口数据:  
当点击不同按钮时, 执行不同的读取操作:

```
25 <script>
26   function onClickBtn(url) {
27     let req = new XMLHttpRequest()
28     req.open("GET", url, true)
29     req.onreadystatechange = () => {
30       if (req.readyState === 4 && req.status === 200) {
31         let data = req.responseText
32         alert(data)
33       }
34     }
35     req.send()
36   }
37 </script>
38 <button class="button" style="vertical-align:middle"
39   onclick="onClickBtn('http://time.cybersecurity.seu.edu/api/-
date')"><span>Without CORS</span></button>
40 <p>
41 </p>
42 <button class="button" style="vertical-align:middle"
43   onclick="onClickBtn('http://time.cybersecurity.seu.edu/api/-
datecors')"><span>Use CORS</span></button>
44 </body>
```

4. 点击第一个按钮，显示报错信息，无法跨域通信：



5. 点击第二个按钮，成功跨域读取信息：



6. 在 jsonp.cybersecurity.seu.edu 中通过回调读取信息：

```
1 <html>
2 <body>
3
4 <h4>jsonp.cybersecurity.seu.edu</h4>
5 <div id="jsonpdate">/api/jsonpdate : </div>
6 <script>
7     function handleTime (data) {
8         const divjsonpdate =
9         document.querySelector('#jsonpdate')
10        divjsonpdate.textContent = '/api/jsonpdate : ' +
11        data.date
12    }
13 </script>
14 <script src="http://time.cybersecurity.seu.edu/api/jsonpdate?-
15     callback=handleTime"></script>
```

7. 在未设置 CORS 头部字段的情况下，实现了接口数据的读取：

