

## 任务 1：实现三个主机

### （一）实验内容：

web. cybersecurity. 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)

### （二）实验步骤：

#### 1. 编写 docker-compose.yml 完成容器配置

```
host1: ip = 10.0.0.2, domain = web.cybersecurity.seu.edu  
host2: ip = 10.0.0.3, domain = time.cybersecurity.seu.edu  
host3: ip = 10.0.0.4, domain = jsonp.cybersecurity.seu.edu
```

#### 2. 向主机(运行 docket 的机器)的 hosts 文件中添加以下条目

```
sudo gedit /etc/hosts
```

```
33 10.0.0.2 web.cybersecurity.seu.edu  
34 10.0.0.3 time.cybersecurity.seu.edu  
35 10.0.0.4 jsonp.cybersecurity.seu.edu
```

## 任务 2：在 time. cybersecurity. seu. edu 上实现三个接口

### （一）实验内容

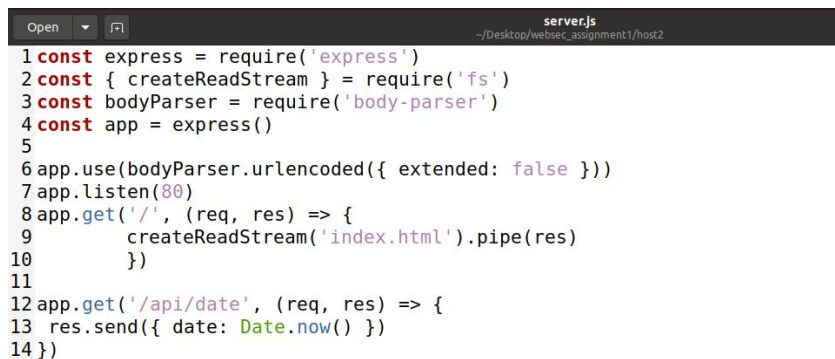
1. /api/date 接口，返回形式为 {date: 124566334} 这样的 json 数据。
2. /api/datecors 接口，返回形式为 {date: 124566334} 这样的 json 数据,并设置 CORS 头部字段。
3. /api/jsonpdate 接口，返回形式为 jsonp 格式的数据。

## (二) 实验步骤

### 1. 进入实验目录

```
cd websec_assignment1
```

### 2. 编写 host2 的 server.js 如下，并且实现接口 1



```
server.js
~/Desktop/websec_assignment1/host2

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

### 3. 继续在 host2 的 server.js 编写实现接口 2 的代码

```
15
16 app.get('/api/datecors', (req, res) => {
17   res.set('Access-Control-Allow-Origin', '*')
18   res.send({ date: Date.now() })
19 })
```

### 4. 继续在 host2 的 server.js 编写实现接口 3 的代码

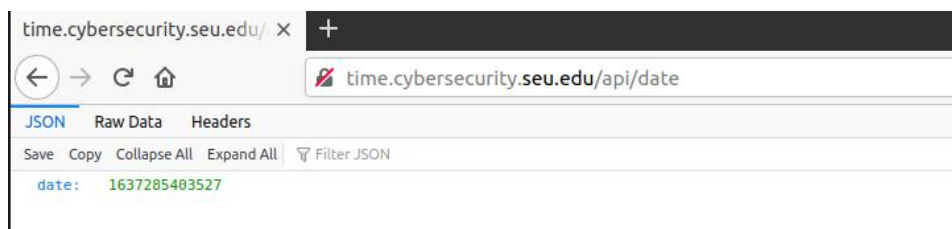
```
21 app.get('/api/jsonpdate', (req, res) => {
22   res.jsonp({ date: Date.now() })
23 })
```

### 5. 启动容器

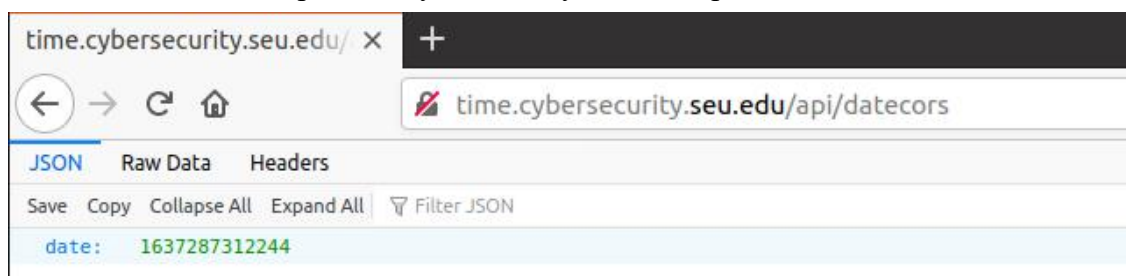
```
sudo docker-compose -f docker-compose.yml up
```

### 6. 验证接口实现效果

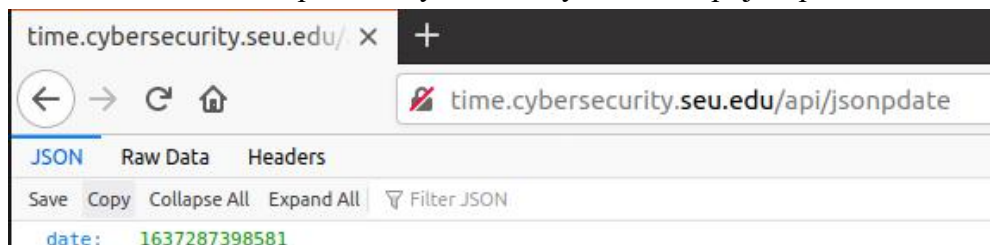
- 在浏览器访问 <http://time.cybersecurity.seu.edu/api/date>



- 在浏览器访问 <http://time.cybersecurity.seu.edu/api/datecors>



- 在浏览器访问 <http://time.cybersecurity.seu.edu/api/jsonpdate>



### 任务 3：在 web. cybersecurity. seu. edu 下实现一个页面

#### （一）实验内容：

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. cybersecurity. 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。

## (二) 实验步骤

### 1. 编写对应 index.html

```
Open  ~index.html  ~/Desktop/websec_assignment1/host1  Save  ⋮  -  ⌵  ✖

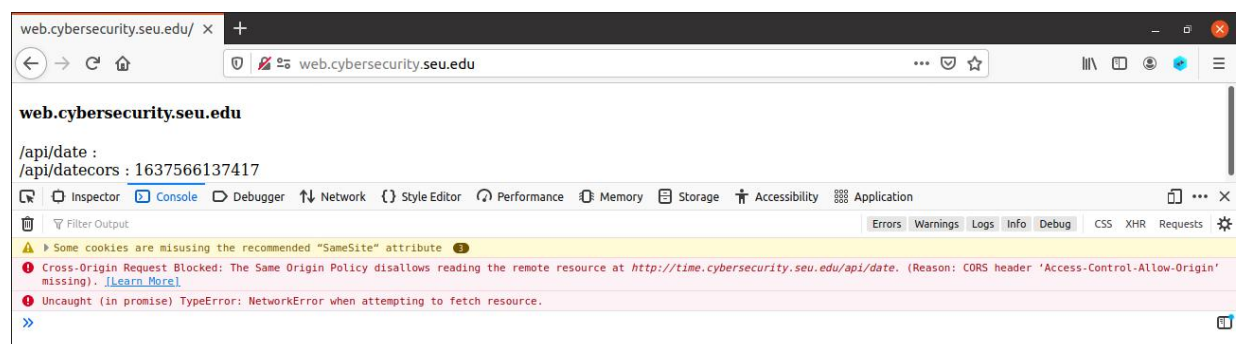
1 <html>
2 <body>
3 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4 <h4>web.cybersecurity.seu.edu</h4>
5 <div id="date">/api/date : </div>
6 <div id="datecors">/api/datecors : </div>
7 <script type="text/javascript">
8     const divdatecors = document.querySelector('#datecors')
9     const rescors = fetch('http://time.cybersecurity.seu.edu/api/datecors').then(res =>
10     res.json()).then(data => {
11         divdatecors.textContent = '/api/datecors : ' + data.date
12     })
13     const divdate = document.querySelector('#date')
14     const resdate = fetch('http://time.cybersecurity.seu.edu/api/date').then(res =>
15     res.json()).then(data => {
16         divdate.textContent = '/api/date : ' + data.date
17     })
18 </script>
19 </body>
```

### 2. 编写对应的 server.js

```
Open  server.js  ~/Desktop/websec_assignment1/host3

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 })
```

### 3. 在浏览器访问 <http://web.cybersecurity.seu.edu/>



可以看到在未设置 cors 头时，浏览器会报错提示不满足同源策略要求；在设置 cors 头后能读取到 time.cybersecurity.seu.edu 的数据。

## 任务 4：在 jsonp.cybersecuiry.seu.edu 下实现一个页面

### （一）实验内容：

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，提供读取成功的截图。

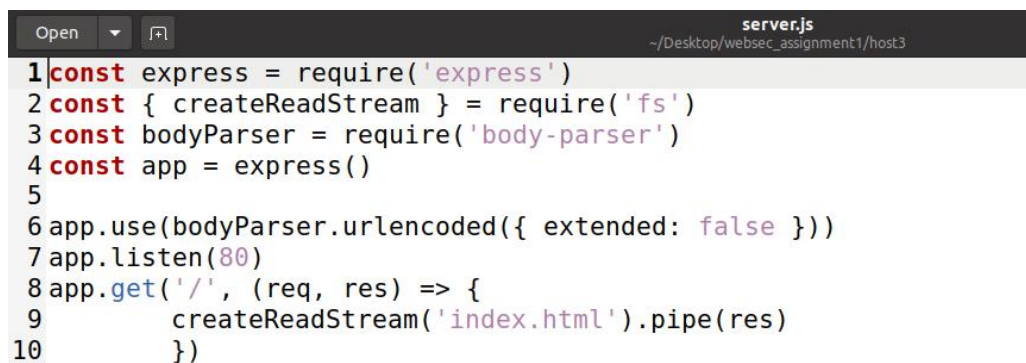
### （二）实验步骤：

#### 1. 编写对应的 index.html 调用 jsonpdate 接口



```
1<html>
2<body>
3<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
4<script>
5  function handleTime (data) {
6    console.log('got the date', data.date)
7  }
8</script>
9<script src='http://time.cybersecurity.seu.edu/api/jsonpdate?callback=handleTime'></script>
10
11
12</body>
13</html>
```

#### 2. 编写对应的 server.js



```
1const express = require('express')
2const { createReadStream } = require('fs')
3const bodyParser = require('body-parser')
4const app = express()
5
6app.use(bodyParser.urlencoded({ extended: false }))
7app.listen(80)
8app.get('/', (req, res) => {
9  createReadStream('index.html').pipe(res)
10})
```

3. 启动容器，在浏览器访问，<http://jsonp.cybersecurity.seu.edu/>，  
结果如下图，表明读取成功。

