

任务一：实现三个主机

一、实验内容

```
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)
```

二、实验步骤

1. 编写 docker-compose.yml 完成容器配置
2. 向主机的 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
```

任务二：在 time.cybersecurity.seu.edu 上实现三个接口

一、实验内容

- 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。

二、实验步骤

进入实验目录

编写 host2 的 server.js 如下，实现接口 1/2/3

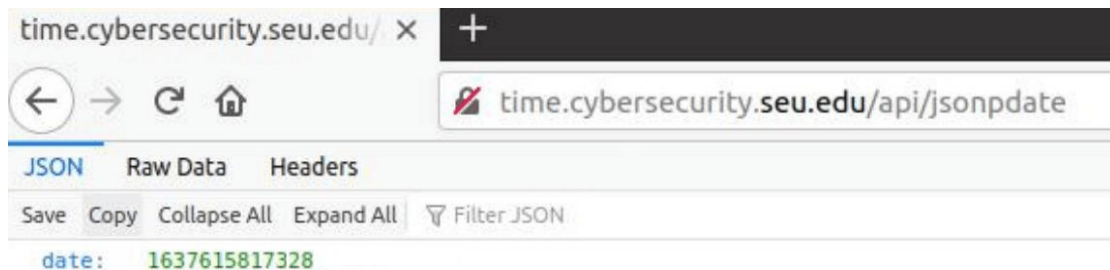
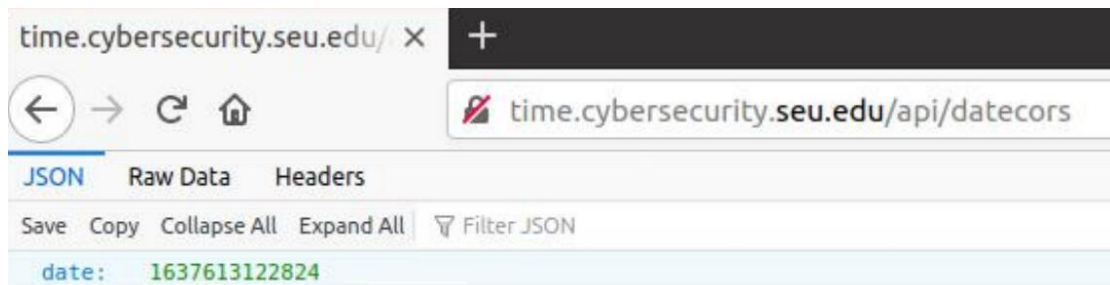
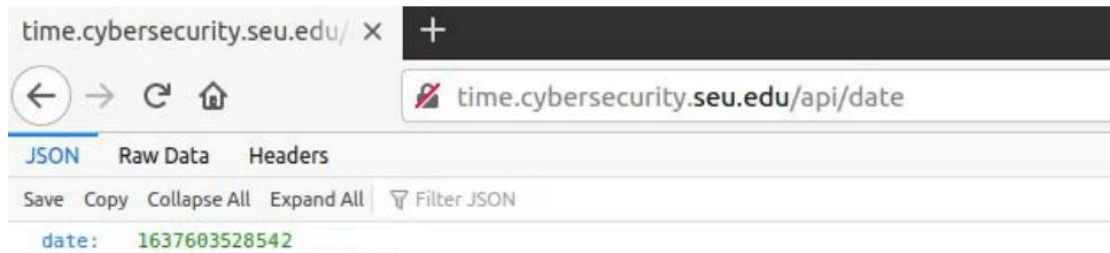
```
server.js
~/Desktop/websec_assignment1/host2

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 })
15
16 app.get('/api/datecors', (req, res) => {
17     res.set('Access-Control-Allow-Origin', '*')
18     res.send({ date: Date.now() })
19 })
20
21 app.get('/api/jsonpdate', (req, res) => {
22     res.jsonp({ date: Date.now() })
23 })
```

出现错误：要安装 express 模块

```
Host1 | Error: Cannot find module 'express'
Host1 | Require stack:
Host1 |   - /data/server.js
Host1 |     at Function.Module._resolveFilename (node:internal/modules/cjs/loader:933:15)
Host1 |     at Function.Module._load (node:internal/modules/cjs/loader:778:27)
Host1 |     at Module.require (node:internal/modules/cjs/loader:999:19)
Host1 |     at require (node:internal/modules/cjs/helpers:102:18)
```

安装之后。在浏览器搜索网址：可看到以下结果：



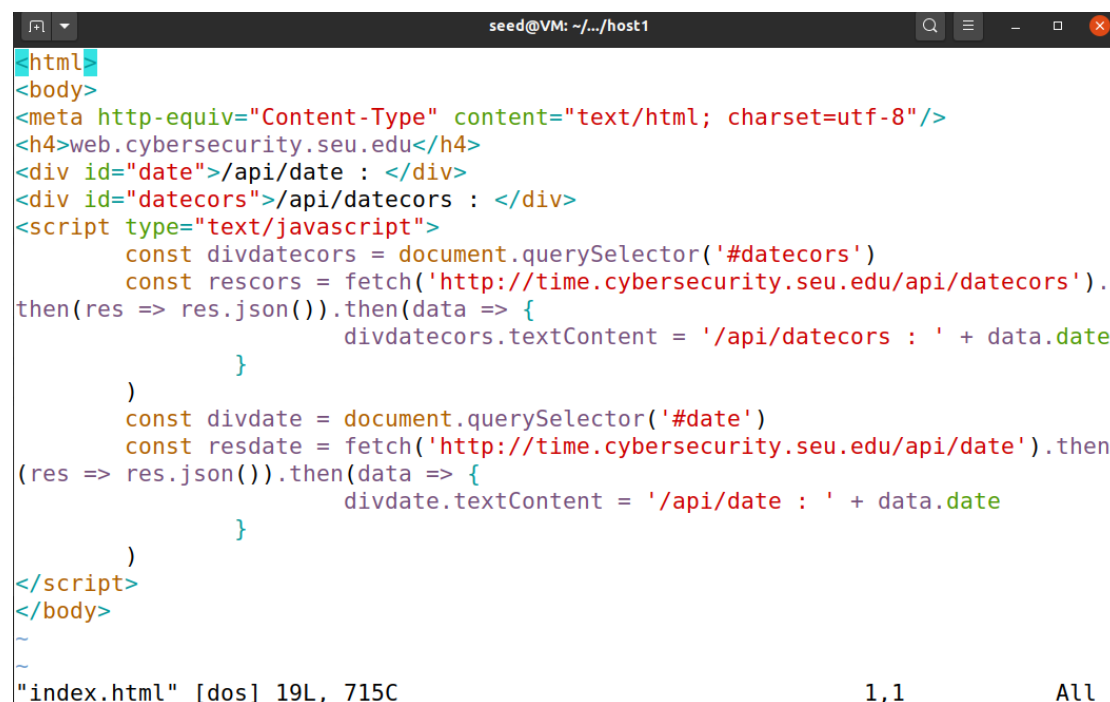
任务三：在 web.cybersecuiry.seu.edu 下实现一个页面

一、实验内容

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

二、实验步骤

编写 index.html

A screenshot of a code editor window titled "seed@VM: ~/.../host1". The editor displays the content of an index.html file. The code includes an HTML structure with two divs, #date and #datecors, and a JavaScript script using the fetch API to retrieve data from time.cybersecurity.seu.edu. The status bar at the bottom indicates the file is "index.html" with DOS line endings, 19 lines, and 715 characters. The cursor is at line 1, column 1.

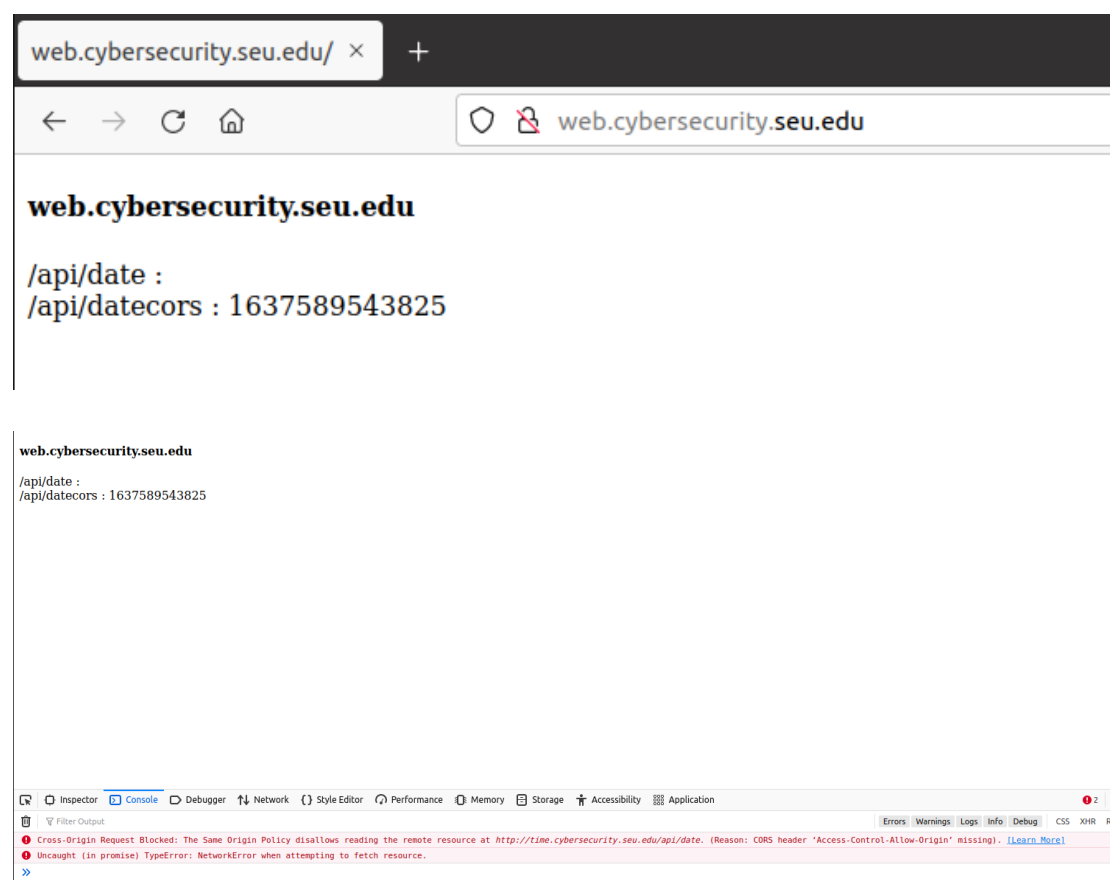
```
<html>
<body>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<h4>web.cybersecurity.seu.edu</h4>
<div id="date">/api/date : </div>
<div id="datecors">/api/datecors : </div>
<script type="text/javascript">
    const divdatecors = document.querySelector('#datecors')
    const rescors = fetch('http://time.cybersecurity.seu.edu/api/datecors').
then(res => res.json()).then(data => {
        divdatecors.textContent = '/api/datecors : ' + data.date
    })
    const divdate = document.querySelector('#date')
    const resdate = fetch('http://time.cybersecurity.seu.edu/api/date').then
(res => res.json()).then(data => {
        divdate.textContent = '/api/date : ' + data.date
    })
</script>
</body>
~
~
"index.html" [dos] 19L, 715C
1,1 All
```

编写对应的 server.js

```
server.js
~/Desktop/websec_assignment1/host1

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 })
```

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



可看到在未设置 cors 头时，浏览器会提示不满足同源策略要求；设置之后就能读取到 time.cybersecurity.seu.edu 的数据

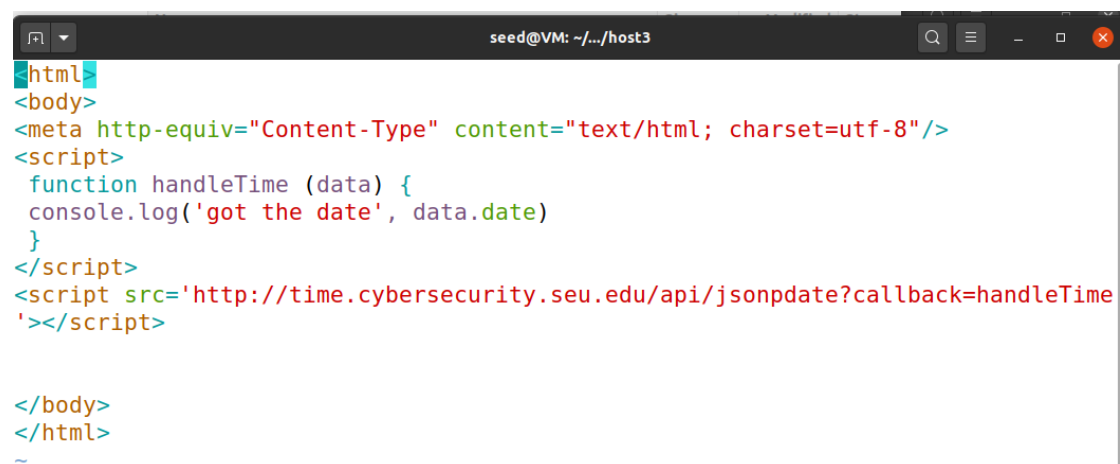
任务四：在 jsonp. cybersecuiry. seu. edu 下实现一个页面

一、实验内容

在 jsonp. cybersecuiry. 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

二、实验步骤

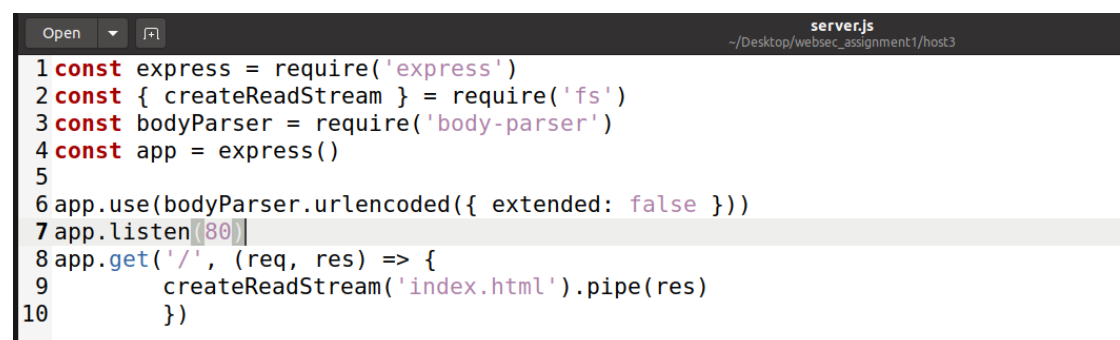
编写对应 index.html



```
<html>
<body>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<script>
  function handleTime (data) {
    console.log('got the date', data.date)
  }
</script>
<script src='http://time.cybersecurity.seu.edu/api/jsonpdate?callback=handleTime'></script>

</body>
</html>
~
```

编写对应 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 })
```

访问 <http://jsonp.cybersecurity.seu.edu/>

