

# Web 安全实验一

## A.实现三个 docker 主机的本地 DNS

### 1.配置宿主机 hosts

```
[11/24/21]seed@VM:~$ sudo vi /etc/hosts

# For Web security
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.导入三个主机的配置文件

```
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
19   host2:
20     image: node
```

### 3.在每个主机的文件夹中运行 npm install express

```
[11/24/21]seed@VM:~/.../host1$ npm install express
added 50 packages, and audited 51 packages in 11s
found 0 vulnerabilities
```

## B. 实现网络 api

在 time.cybersicurity.seu.edu 上实现三个：

i./api/date 接口

li./api/datecors 接口，并设置 CORS 头部字段

lii./api/jsonpdate 接口

### 1. 编写 host2 目录下的 server.js 文件

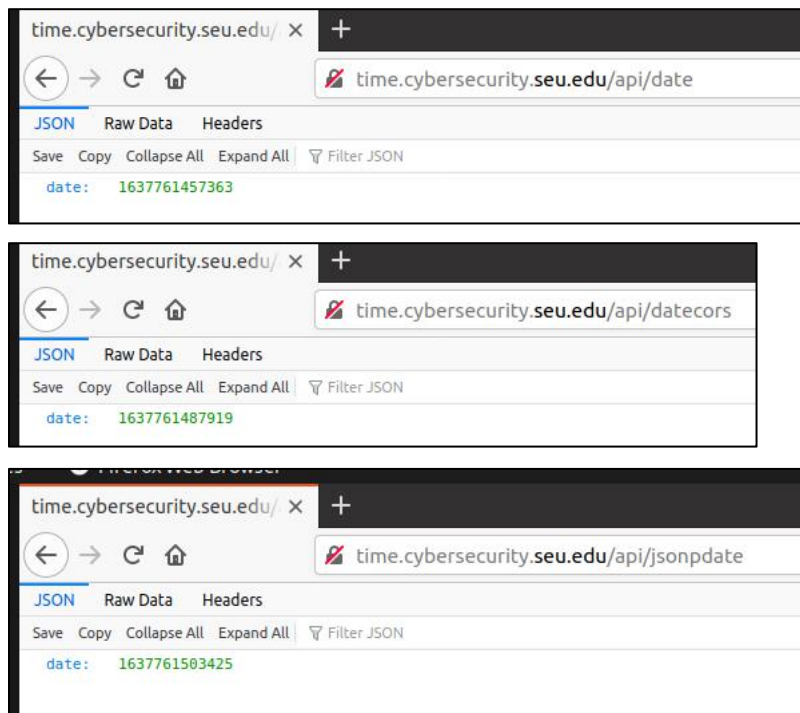
```
index.html × *server.js
1 const express = require('express')
2 const { createReadStream } = require('fs')
3 const bodyParser = require('body-parser')
4 const app = express()
5
6 app.use(bodyParser.urlencoded({ extended: false }))
7 app.listen(80)
8
9 app.get('/', (req, res) => {
10     createReadStream('index.html').pipe(res)
11 })
12
13 app.get('/api/date', (req, res) => {
14     res.send({ date: Date.now() })
15 })
16
17 app.get('/api/datecors', (req, res) => {
18     res.set('Access-Control-Allow-Origin', '*')
19     res.send({ date: Date.now() })
20 })
21
22 app.get('/api/jsonpdate', (req, res) => {
23     res.jsonp({ date: Date.now() })
24 })
```

### 2. 进入作业文件夹，运行容器配置

```
[11/24/21]seed@VM:~$ cd Desktop/assignment1/
```

```
[11/24/21]seed@VM:~/../assignment1$ sudo docker-compose -f docker-compose.yml up
Starting Host1 ... done
Starting Host2 ... done
Starting Host3 ... done
Attaching to Host1, Host3, Host2
```

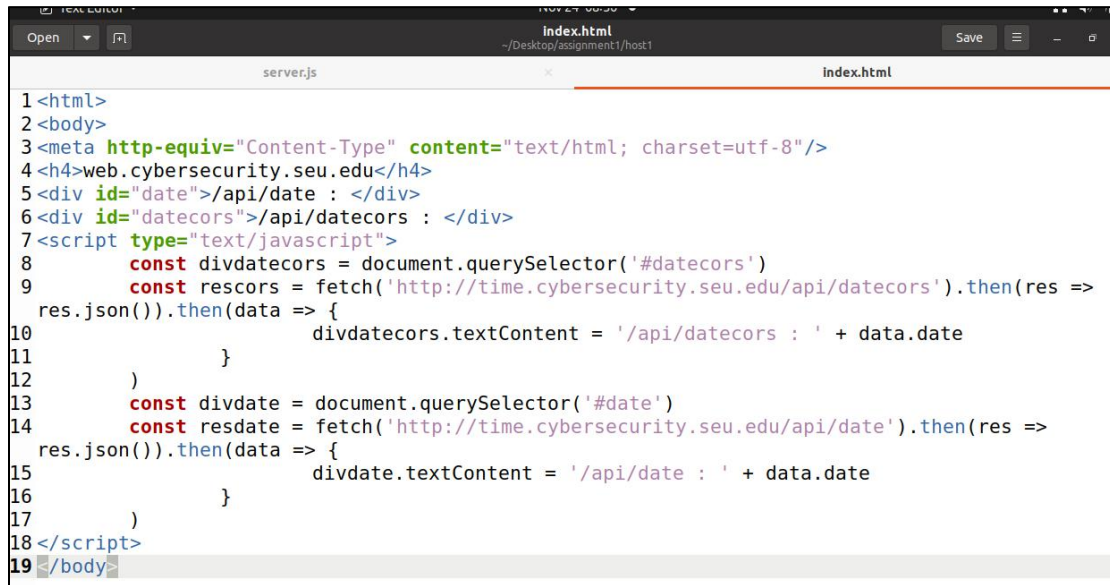
### 3.在浏览器中分别尝试三个接口



## C. 获取数据

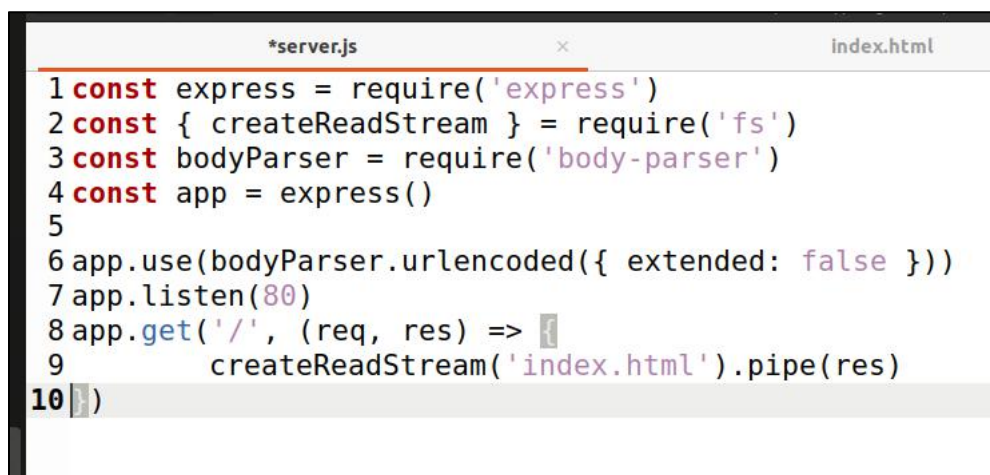
在 Web.cybersicurity.seu.edu 下实现一个页面，在页面中通过 js 代码读取 time.cybersicurity.seu.edu 的接口数据，分别测试在 time.cybersicurity.seu.edu 中设置和未设置 CORS 接口的情况下，Web.cybersicurity.seu.edu 读取接口数据的情况，提供读取成功和未读取成功模式下的截图

#### 1.编写 host1 下的 index.html 文件



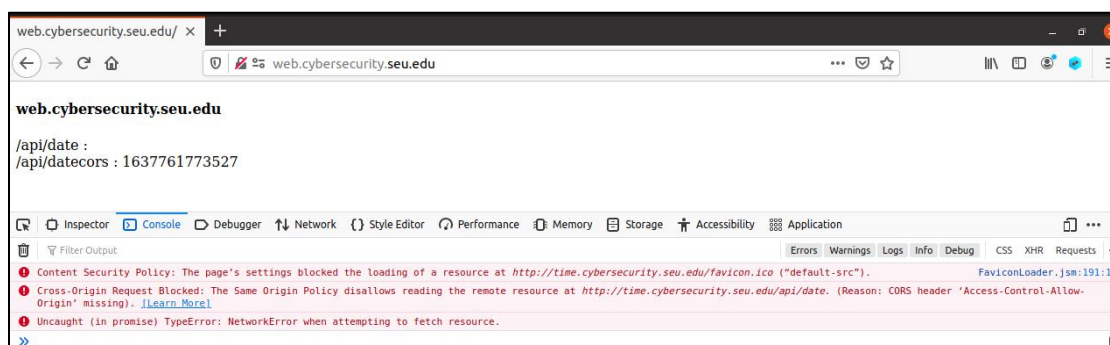
```
1 <html>
2 <body>
3 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
4 <h4>web.cybersecurity.seu.edu</h4>
5 <div id="date">/api/date : </div>
6 <div id="datecors">/api/datecors : </div>
7 <script type="text/javascript">
8   const divdatecors = document.querySelector('#datecors')
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 host1 下的 server.js 文件



```
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 })
```

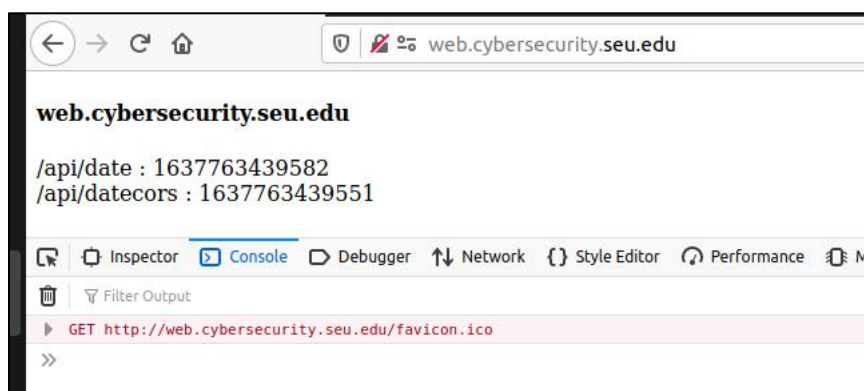
## 3.进入浏览器，访问 web.cybersecurity.seu.edu，发现跨域请求被阻止



## 4.修改 host2 下的 server.js 文件，加入 cors

```
server.js
1 const express = require('express')
2 const { createReadStream } = require('fs')
3 const bodyParser = require('body-parser')
4 const app = express()
5 var cors = require('cors');
6
7 app.use(cors());
8 app.use(bodyParser.urlencoded({ extended: false }))
9 app.listen(80)
10
11 app.get('/', (req, res) => {
12     createReadStream('index.html').pipe(res)
13 })
14
15 app.get('/api/date', (req, res) => {
16     res.send({ date: Date.now() })
17 })
18
19 app.get('/api/datecors', (req, res) => {
20     res.set('Access-Control-Allow-Origin', '*')
21     res.send({ date: Date.now() })
22 })
```

## 5.再次访问，成功

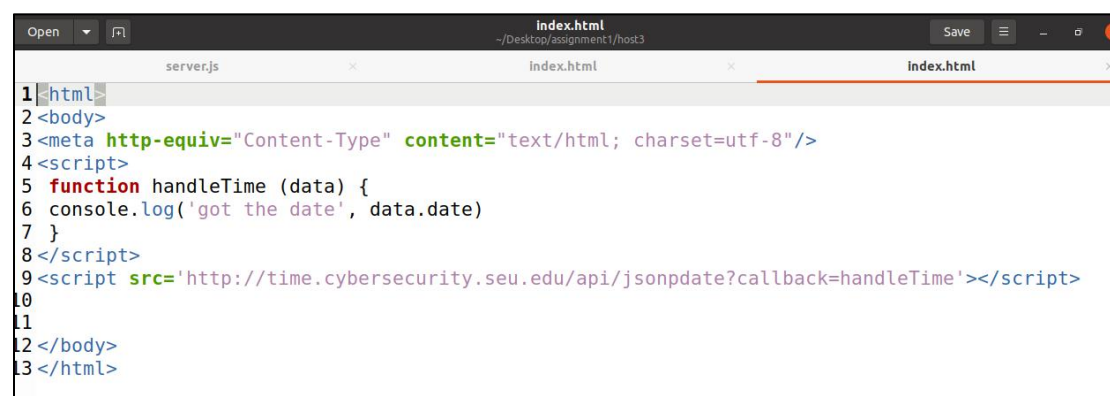


## D.通过回调函数获取数据

在 JsonP.cybersicurity.seu.edu 下实现一个页面，测试在 time.cybersicurity.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.cybersicurity.seu.edu 的接口数据的方法，提供读取成功模式下的截图

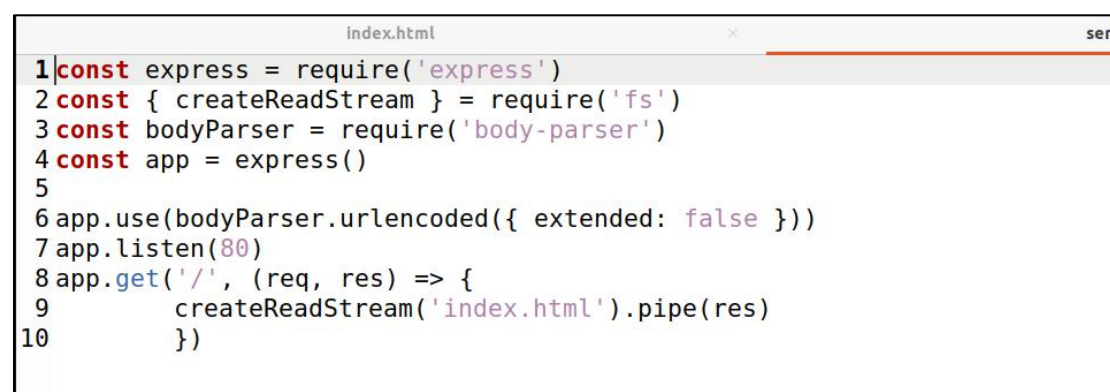


## 1.编写 host3 下的 index.html 文件



```
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 host3 下的 server.js 文件



```
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.浏览器访问，成功读取

