Web安全实验报告1

学号：57118317

a)实现三个主机：

Docker配置如下:

```
version: '3.5'
services:
  host1:
    image: node
    container_name: Host1
    tty: true
    command:
      - /bin/bash
      - -c
      - |
        cd data
        node server.js
    networks:
      WSnetwork:
        ipv4_address: 10.0.0.2
    volumes:
      -  ./host1:/data
      -  /usr/local/lib/node_modules:/data/node_modules

  host2:
    image: node
    container_name: Host2
    tty: true
    command:
      - /bin/bash
      - -c
      - |
        cd /data
        node server.js
    networks:
      WSnetwork:
        ipv4_address: 10.0.0.3
    volumes:
      -  ./host2:/data
      -  /usr/local/lib/node_modules:/data/node_modules

  host3:
    image: node
    container_name: Host3
    tty: true
    command:
      - /bin/bash
      - -c
      - |
        cd /data
```

```
        node server.js
    networks:
      WSnetwork:
        ipv4_address: 10.0.0.4
    volumes:
      -  ./host3:/data
      -  /usr/local/lib/node_modules:/data/node_modules


networks:
  WSnetwork:
    driver: bridge
    ipam:
      config:
        - subnet: "10.0.0.0/24"
```

b)在 time.cybersecurity.seu.edu 主机上实现三个接口:

```
const express = require('express')
const { createReadStream } = require('fs')
const bodyParser = require('body-parser')
const app = express()
app.use(bodyParser.urlencoded({ extended: false }))
app.listen(80)
app.get('/', (req, res) => {
    createReadStream('index.html').pipe(res)
})
app.get('/api/date', (req, res) => {
    res.send({ date: Date.now() })
})
app.get('/api/datecors', (req, res) => {
    res.set('Access-Control-Allow-Origin', '*')
    res.send({ date: Date.now() })
})
app.get('/api/jsonpdate', (req, res) => {
    res.send(req.query.callback + '({ "date": ' + Date.now() + ' })')
})
```

api/date 接口:

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

api/datecors 接口:

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

/api/jsonpdate 接口:

```
app.get('/api/jsonpdate', (req, res) => {
    res.send(req.query.callback + '({ "date": ' + Date.now() + ' })')
})
```

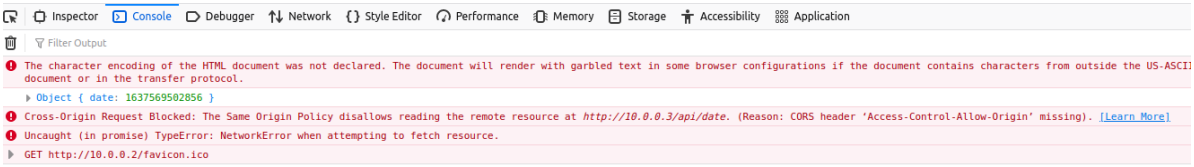
c) 在 web.cybersecurity.seu.edu 下实现一个页面，通过 js 代码读取 time.cybersecurity.seu.edu 的接口数据:

```
<html>
<body>
<script type="text/javascript">
    const rescors =
 fetch('http://time.cybersecurity.seu.edu/api/datecors').then(res =>
res.json()).then(data => console.log(data))
    const res =  fetch('http://time.cybersecurity.seu.edu/api/date').then(res =>
res.json()).then(data => console.log(data))
</script>
</body>
</html>
```

在设置CORS头的情况下，通过api/datecors读取到{ date: 1637569502856 }，而未设置CORS头的api/date会提示：

```
Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote
resource at http://10.0.0.3/api/date. (Reason: CORS header ‘Access-Control-Allow-
Origin’ missing)
```

无法读取到数据。

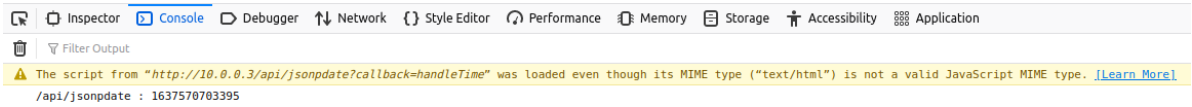


d) 在 jsonp.cybersecurity.seu.edu 下实现页面通过回调 js 代码读取 time.cybersecurity.seu.edu 的接口数据:

回调代码如下：

```
<html>
<body>
<script>
    function handleTime (data) {
        console.log( '/api/jsonpdate : ' + data.date)
    }
</script>
<script src='http://time.cybersecurity.seu.edu/api/jsonpdate?
callback=handleTime'></script>
</body>
</html>
```

在 time.cybersecurity.seu.edu 未 设 置 CORS 头的情况下可以读取/api/jsonpdate接口数据：


Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Filter Output
The script from “http://10.0.0.3/api/jsonpdate?callback=handleTime” was loaded even though its MIME type (“text/html”) is not a valid JavaScript MIME type. [Learn More]
/api/jsonpdate : 1637570703395