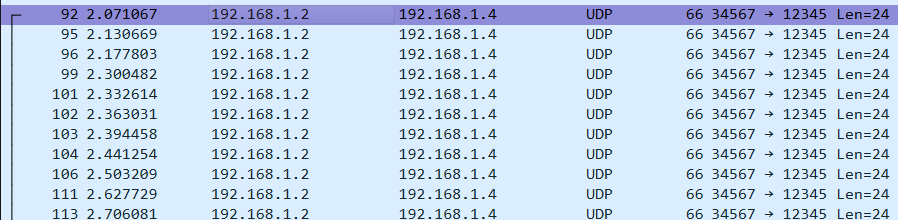
# ISCC2024 WriteUp

AkyOI+李承达+3433778745@qq.com

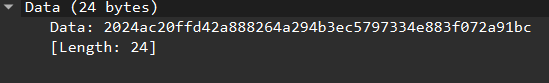
### misc+工业互联网模拟仿真数据分析

### 解题思路（必须包含文字说明+截图）

第一题直接按照ip排序，发现只有这里满足条件，详情见图：



答案：192.168.1.2,192.168.1.4,24

第二题直接看data

发现一堆数据都是以2024开头的，于是不变的就是2024，答案：2024

第三题要看数据包的时间，发现从idx是115的数据包开始，后面每个包相距时间都是0.06秒左右，记录下这些ip就是答案：192.168.1.3,192.168.1.5,0.06

第四题分析每段数据包之间的ip的关系，除了第二段有3个ip互相通信，其他都是2个ip互相通信，因此可以推出第二段内有多个服务，所以答案就是第二段内涉及的ip：192.168.1.2,192.168.1.3,192.168.1.6

第五题，由5个字的算法可以知道是crc16或者crc32，可以知道校验码最后一位为1，经过尝试可以知道开头是4，算法是crc16，于是答案就是CRC16,4,1

之后跑脚本就得到flag：adcca5c2a82064a17a645d35b6b054cd

### Exp

import hashlib

def generate\_flag(\*answers):

# 将所有答案使用英文逗号连接

combined\_answers = ','.join(answers)

# 生成flag格式

initial\_flag = f"ISCC{{{combined\_answers}}}"

# 对flag进行MD5加密

md5\_hash = hashlib.md5(initial\_flag.encode()).hexdigest()

return md5\_hash

# 示例用法

if \_\_name\_\_ == "\_\_main\_\_":

# 每道题目的所有填空写在一个字符串中

answers = [

"192.168.1.2,192.168.1.4,24", # 第一小题答案：IP顺序从小到大排列，涉及的IP个数由选手自己判断，数值为整数

"2024", # 第二小题答案：数值为整数

"192.168.1.3,192.168.1.5,0.06", # 第三小题答案：IP顺序从小到大排列，涉及的IP个数由选手自己判断，数值保留小数点后2位

"192.168.1.2,192.168.1.3,192.168.1.6", # 第四小题答案：IP顺序从小到大排列，涉及的IP个数由选手自己判断

"CRC16,4,1" # 第五小题答案：数据校验算法名称长度为5个字符，其中英文字母大写

]

# 生成MD5加密后的flag

final\_flag = generate\_flag(\*answers)

# 输出最终的MD5加密字符串

print(final\_flag)