# 工业物联网智能网关数据分析

*发表于 2021-01-25 更新于 2021-02-16 分类于 Challenge ， 2020 ， 工业信息安全技能大赛 ， 济南站*
*Challenge | 2020 | 工业信息安全技能大赛 | 济南站 | 工业物联网智能网关数据分析*

*点击此处获得更好的阅读体验*

---

## *WriteUp来源*

*来自MO1N战队*

## *题目描述*

> *某工控内网环境有台设备环境被破坏了,安全人员想通过当时抓的流量包进行溯源, 你能帮他发现什么东西么?Flag格式为：flag{}。*

## *题目考点*

- *MQTT协议*
- *zip文件格式*
- *LSB隐写*

## *解题思路*

### *1. 分析流量包*

*对流量包进行分析，发现其中有mqtt协议的数据包。*



*MQTT是一个基于客户端-服务器的消息发布/订阅传输协议。MQTT协议是轻量、简单、开放和易于实现的，这些特点使它适用范围非常广泛。在很多情况下，包括受限的环境中，如：机器与机器（M2M）通信和物联网（IoT）。其在，通过卫星链路通信传感器、偶尔拨号的医疗设备、智能家居、及一些小型化设备中已广泛使用。*

### *2. 过滤出MQTT*

*在wireshark中过滤一下mqtt协议相关的数据包进行分析。*



*过滤后，发现数据包存在异常。查看每条流量信息中的内容。*

### *3. 找到可疑字符串*

*当查看到[d]的时候，出现了一串可疑的字符串*







`pass_1s_ea4y`*然而这并不是flag，在这里把它记录下来，留作后续使用。*

### *4. 追踪流*

*继续按照同样的方法，追踪每条信息的分组字节流。*



*当分析到[f]这条流量信息的时候，里面的内容出现了`504B0304`字符串，显然这是一个zip压缩包的文件头。尝试提取字符串还原成压缩包。*

### *5. 还原ZIP*

*显然，光是[f]里面的字符串不足以构成一个zip，但是确实是zip文件的16进制头。尝试用相同的方法找到zip的十六进制字符串的其他部分。既然在[f]中找到了头部，而flag中的f也是位于头部，猜想完整的zip是由[f],[l],[a],[g]中的数据信息构成。提取出其中的信息。*

*提取出字符串*

### *6.保存zip文件*

*在winhex或hex fiend中还原为zip格式，保存为123.zip，成功还原出zip压缩包*

### *7. 压缩包密码破解*

*打开压缩包提示需要密码，之前在[d]中，已经得到一个特殊的字符串，`pass_1s_ea4y`，尝试作为解压密码。成功解压，得到一张图片。*

### *8. 修复图片*

*怀疑密码就藏在图片之中，得到的图片是半张二维码，尝试根据图片十六进制信息还原图片,查看图片的详细信息，将宽度和高度的像素值转换成对应的16进制数值，通过这个操作，当使用winhex打开图片时，可以比较快速的定位出代表长宽的区域，便于修改。*

*如上图，`130`对应为`0x82`，`260`对应为`0x104`，只需要在winhex中把`0082`改成`0104`，即可把高度也调成了260像素。*

*调整后，出现了一半的flag*

### *9. 图片隐写分析*

*使用stegsolve打开图片。在red，green，blue为0的时候，存在显示异常。*

### *10. LSB隐写分析*

*可以推测，应该是考察lsb隐写的相关知识,使用stegsolve分离出里面隐藏的信息。*

*lsb技术（LSB全称leastsignificant bit，是一种基于图片最低有效位修改储存信息的隐写方法），在里面隐藏着一张png的图片。点击下方的save bin按键，保存为后缀为png的文件。就可把隐藏的png提取出来。*

### *11. 拼接二维码*

*现在我们已经得到了半个flag，并且得到了两个半张的二维码图片，尝试拼接二维码。*

发现二维码经过了反色处理，使用PS（快捷键`Ctrl+I`），将二维码进行反色，恢复原来的样子。

### 12. 扫描二维码

使用QR Research进行扫码，可得到后半段flag。

## Flag

```
1 flag{21png_LSB_is_easy}
```



# Challenge # 2020 # 工业信息安全技能大赛 # 济南站
被篡改的数据
某嵌入式设备固件升级包