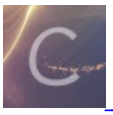
工控ctf

原创

multi4 于 2020-10-29 19:38:53 发布 3770 收藏 12

分类专栏： 安全学习 # misc

本文链接：https://blog.csdn.net/qq_45951598/article/details/109364567

 安全学习 同时被 2 个专栏收录

34 篇文章 3 订阅

订阅专栏

 misc

6 篇文章 0 订阅

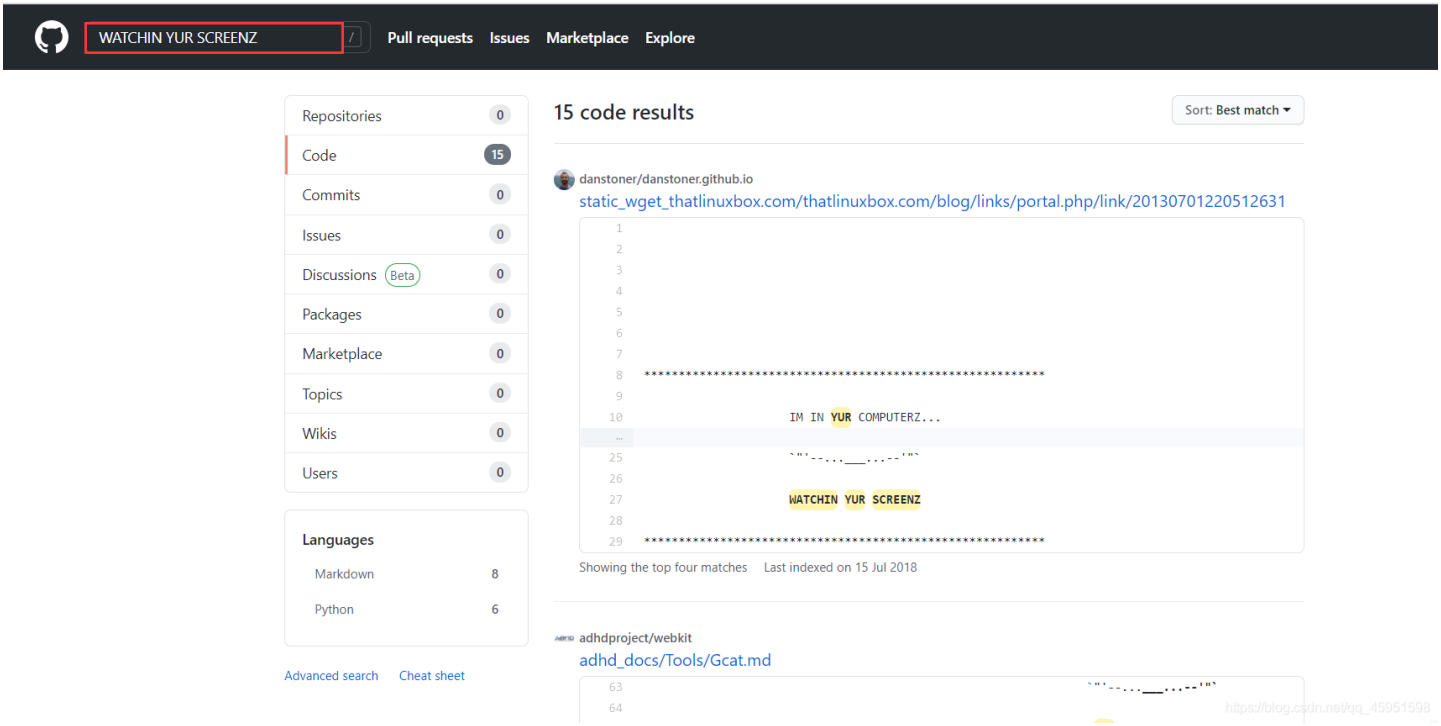
订阅专栏

文章目录

黑客的大意

下载后得到mail，在文件后面jpg
打开图片是一个这样的图片，

这里我们根据题目的意思：
黑客在入侵后不小心留下了这样一个文件，请分析文件进行溯源，找到黑客的邮箱。flag格式为flag{邮箱账号}
这里的主要关键词是对文件进行溯源,以及flag格式为邮箱账号
然后我们看文件有什么特征，然后发现下面有一句英文
我们去搜索一下，这里说一下，我们一般搜索都可以去github看一下，有没有这个项目。
这里直接搜。



这里随便打开一个网页，我打开的是第二个.
这里翻一下看到一个邮箱，然后我们试试，这里的格式是flag{}
发现这个就是flag。

Gcat works in two parts.

Example 1: Deploying an Implant

Configuring gcat and its implant is quite simple. You need to have a gmail account that you plan to use. Then simply pop the username and password into the two scripts gcat.py and implant.py as the "gmail_user" and "gmail_pwd" variables.

```
########################################
gmail_user = 'gcat.is.the.shit@gmail.com'
gmail_pwd = 'veryc00lp@ssw0rd'
server = 'smtp.gmail.com'
server_port = 587
########################################
```

After you have set both gcat.py and implant.py to be properly configured you can then run the implant on a target machine.

It's just a simple python script.

Example 2: Running a Command & Retrieving Output

Once you have an implant or two running you can control them from gcat.py. To start we will want to get the list of all attached implants and their ids. To do this run gcat.py with the -list option.

```
/opt/gcat$ python ./gcat.py -list
```

```
f964f907-dfcb-52ec-a993-543f6efc9e13 Windows-8-6.2.9200-x86
90b2cd83-cb36-52de-84ee-99db6ff41a11 Windows-XP-5.1.2600-SP3-x86
```

Now that we've got a list of attached implants, let's send a command to one.

Modbus协议分析

题目：黑客通过外网进入一家工厂的控制网络，之后对工控网络中的操作员站系统进行了攻击，最终通过工控协议破坏了正常的业务。我们得到了操作员站在攻击前后的网络流量数据包，我们需要分析流量中的蛛丝马迹，找到FLAG。
题目附件连接：https://pan.baidu.com/s/1jGu7-1EKc29HTQc-pCJZlw （提取码：8kqx）

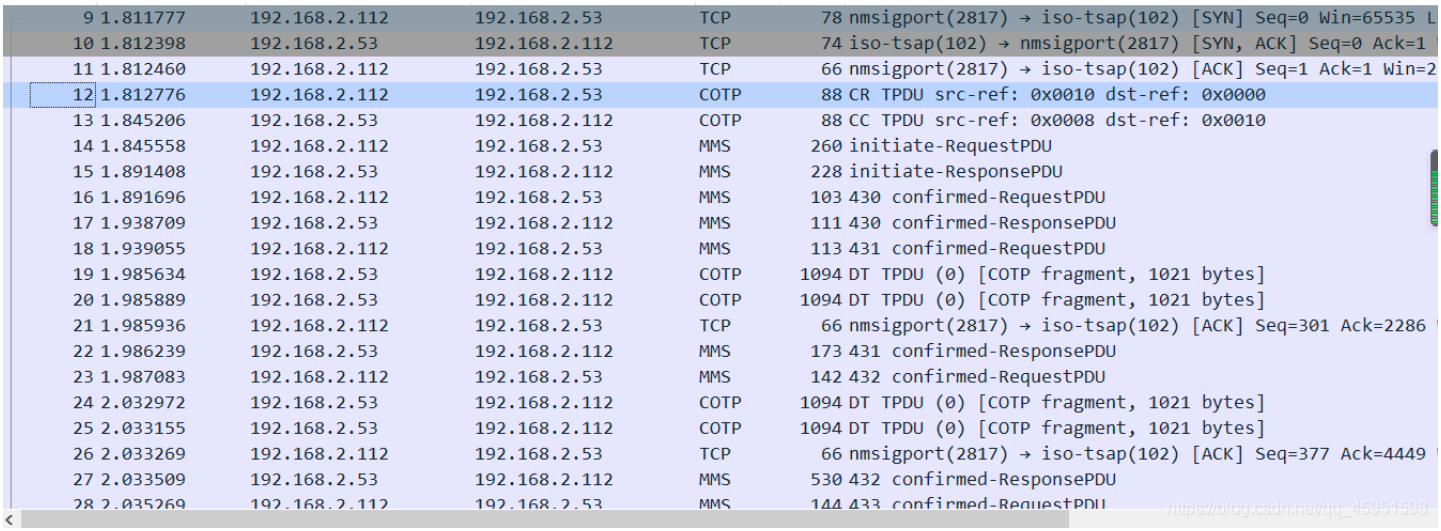
工业协议分析1

题目：工业网络中存在异常，尝试通过分析PCAP流量包，分析出流量数据中的异常点，并拿到FLAG。

题目附件连接：https://pan.baidu.com/s/17jkHLBqcjxP0o9FpGlfObA （提取码：95ds）

解题步骤：

打开流量包，发现存在PRES、TCP、COTP、MMS协议的流量，其中选择一个数据包，追踪TCP流发现存在关键字flag.txt，如图所示：



这里对多个数据包进行tcp追踪流

发现好像都有flag.txt，所以我们分析所有包都有的话，那么肯定不是这个的.

2、然而通过多次分析与flag.txt相对应的流量包中，没有发现flag.txt的内容，于是换一个思路，对流量包进行关键字（jpg、png、zip、rar、flag）搜索，查看是否存在其他的文件。在linux系统中使用grep指令，可以对文件进行指定关键字搜索。linux中grep命令用法，我们使用指令进行关键字搜索

```
grep "flag" -a test.pcap
```

```
[root@10-255-0-66 ~]# grep "flag" -a test.pcap
m_ctime = a CFileStatus at Warning: CFile::GetStatus() returns m_attribute without high-order flags.
lag.txti        2018061411190OZ0, iec61850server.cfg2018061411122OZ0. iec61850server.exei20150929080116Z0%

lag.txtЗ8¹L{{
             )±)`EmX¯(5(pf
                        Di7xn‰a¯
lag.txt¢Ѯ8y{{
             )±)`EmX (5(pf
                        Di:on‰dþo
lag.txti        2018061411190OZ0, iec61850server.cfg2018061411122OZ0. iec61850server.exei20150929080116Z0%

lag.txt□m884{{
              )±)`Eme@X0(5(pf
                          Di>\n‰e5i
lag.txti        2018061411190OZ0- iec61850server.cfgi2018061411220 4Z0. iec61850server.exei20150929080116Z0%

lag.txt□8
                {{
                  )±)`Em!@S(5(pf
                              |¹□§
<!-- *  as well as to set memory debug flags. This module is            -->
m_ctime = a CFileStatus at Warning: CFile::GetStatus() returns m_attribute without high-order flags.
```



```
grep ".zip" -a test.pcap
```

```
[root@10-255-0-66 ~]# grep ".zip" -a test.pcap
[root@10-255-0-66 ~]#
```

```
grep ".jpg" -a test.pcap
```

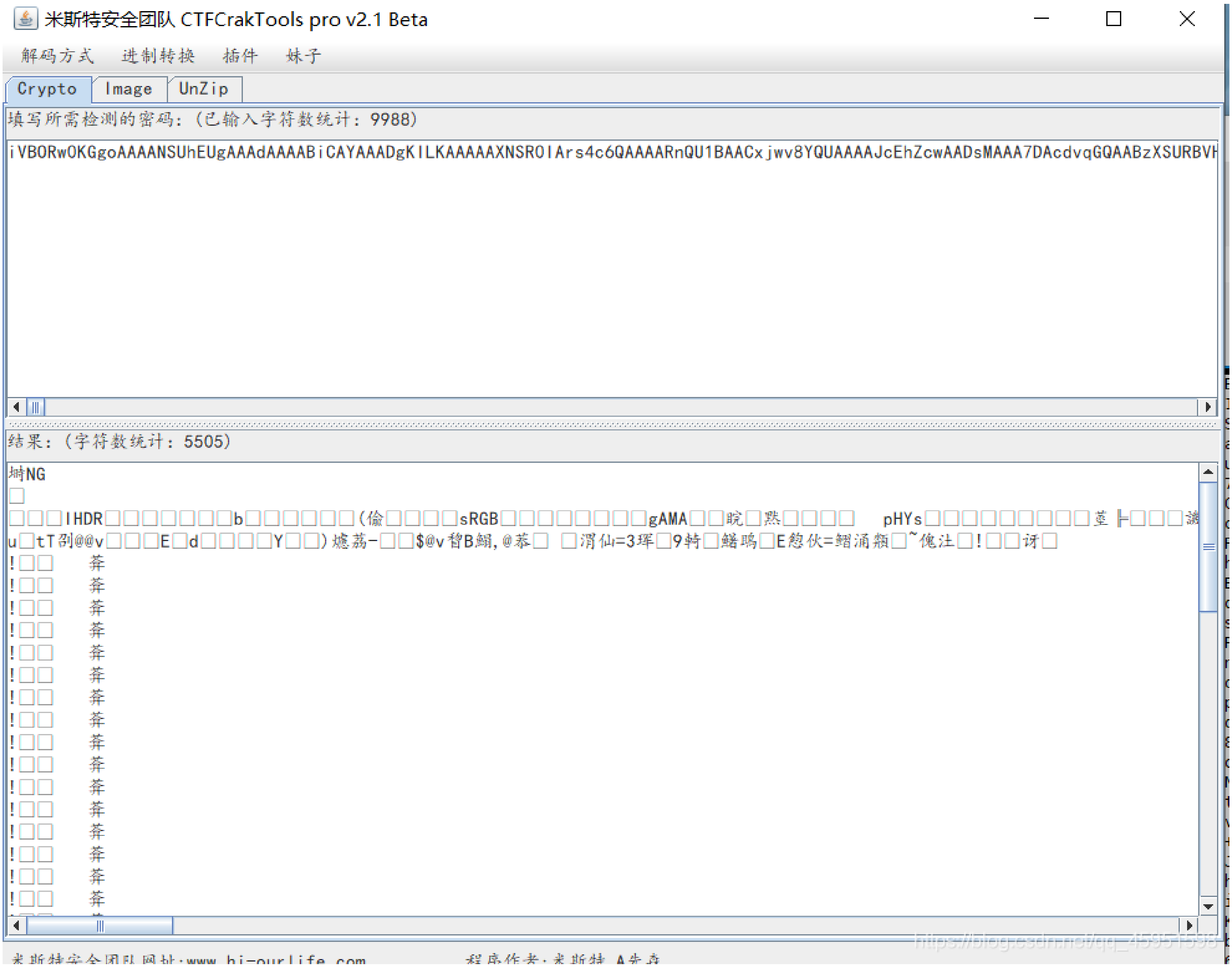
```
[root@10-255-0-66 ~]# grep ".zip" -a test.pcap
[root@10-255-0-66 ~]# grep ".jpg" -a test.pcap
```

这里zip和jpg都没有东西

```
grep ".png" -a test.pcap
```

但是这里搜索png的时候出来了一些字符组合，这里在png后面有提示base64加密

这里解密发现肯存在一张图片，因为开头有NG两个字，所以我们把




```python
# coding = utf -8
import os , base64
img_str= 'iVBORw0KGgoAAAANSUhEUgAAAdAAAABiCAYAAADgKILKAAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA
7DAcdvqGQAABzXSURBVHhe7Z2Js11Fncfn75maZmaZkaS0elXAp1GHRUhGFAQHYQFQRFBWQRiBoBWQyyKaBsxo0tCAgkQHayQEL2jSxAyEoSIA
HOvM/JPTPn9fv10d19+9z3bvh+qr5FkXe77z33nt0/71//fr/+m0IIIYQQ0ciACiGEEAnIgAohhBAJyIAKIYQQCciACiGEEAnIgAohhBAJyIAKIY
QQCciACiGEEAnIgAohhBAJyIAKIYQQCciACiGEEAnIgAohhBAJyIAKIYQQCciACiGEEAnIgAohhBAJyIAKIY
QQCciACiGEEAnIgAohhBAJyIAKIYQQCciACiGEEAnIgAohhBAJyIAKIYQQCciACiGEEAl0ZkDff78oNm/ZV8
xfuK2Y8fxrxex5W4vlK3cVe/Ye6L1CCCGEGF6yGtC33n63uOPuNcW/fHJG8bcffarsf/3UjNLAVtz/h/Xm6/h3IYQQYiKSzYBufePt4thT55qG0N
Xxp8/ttTqIDGj/vP3Oe+VKf/L1y4qTzppffOq/ni/+6bDpxY+ve7n3irHw/VrKzdsjE6tFL24v7p26rrhs0pLy8x3+pZnFhz/zbDnZ+th/PFd86a
```

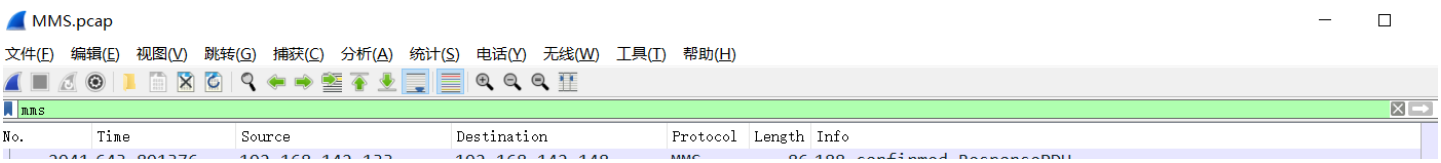
uzi3MuWFhcc+Py4qFpG4s1694cNcmKYf/+94oXFm4rbr1zdXHe9xcVn/+fWcVHPvv/7/WVk+YU371kcXHbyN9nzd1a7JVXRHxAeGdknFi2Ylfx+F
Nbiql/3lA8MPK8/+XJzcXLy3eWz80wcdxpc4t//sSMciw55Zz5xXW/WF4+zwcODO46shhQBqCjvzbHNIKWfnjVS72WB2HQtl7XxWB+KDJz9tbiM1
+eaX6HTQbUej3KxZKXdxRXTV5aehys92kThm/u/Dd6vbXD9sDd960tPvH558z+fGKicfmPl5RGW4hBsnDx9lGT10q5YTuNyemHDn/WfAYQk8xf3L
qyeGPb271W3bLl1X2l0bP01PRXe6/ygwG1ruOLx88uFr+0o/eqbsliQPlhrAvxiZVBHW4Y63Vd3EiHGtOe2Gx+d5XGw4CuWLWrOPv8BWbfsXp25u
u9XptZMDIQ+SYRoXrksU293oToFp6Rc7+3yLwPUS7ee+/94r6p64t//Ph0830sYWSffGZLr4duwLt0wcWLzfdHGNE2fAYU/f1HpxezR1ajXdO3Ad
24aW/xdx+xL+LIY2cV0597rdj95v7yh9y9e3/pysOFUEcGNA1mcKyerO/us0fNLF2iv/vjht6rx2K1Q6ngAvrlHau890OKdu3a3+vdz2NPbs7ynq
++tq/XoxDd8MrGvaUHzrr/6srBu+++X1w5eanZf4gwvF2BXbDes1KIAWXhdta3FxSHHWl7nNiueXNP+/jRD30b0Ft+tcr88J/7ysxi+/YwV4AMaB
pTbls55js75uQ5xcpVu3uvaMZtWymFnbveKc449wWzv1SxV9rG3BfeyGI8+d6E6IrXt75VDvisjKz7z1UOMEJW3zF68un8K9E9ew6U+5bW+1UKMa
AVrGbJ9vj0F58f088fH3ql96pu6NuAEvzhfmjEDCMUGdA08PXXv69/+Nj0cq8jlHrbumLZsfOdoD3wE8+cV9zzwLrS3cpqjz1LghrwUODJwBiyh0
lAAK8norsJ2n3yC2Mfmkpfv2BBOQDQ9959B8pgph073iknGLi+L5+05P/2Z2+/a/S2ghA52LV7f3Hrr1eXwS7u/dmkfmla4bG4mfqnDaU3kBiDm+
9YVfzbp+3Px/OBpysnN94yduLvKsaAVry4ZMeYfggi7JK+DCgrTPcDo3//3LNRkVAyoPGQMuR+X0SWxuC2rxQDRunkrx80eD5d+MPFwaviig0b9x
SbNu/t/Z/NXfesNd+P/Z6/PtMehAAYcaIRyVEWIjd4Uax7tE39wGTRWo0hDIqVi8/E252QVyJyPhfs/YZ4jFIMKBBhX++HbIQu6cuAWhYfff/yF3
uvCEMGNB5r8tK2YnNx21eK4Wc3LDP7QAQjPPNsuCciBvbUfYMEq0shJgKkSln36FfPmFemWvk8eP1wz+/WmX1iXDCuPja8sse7Ul67vv8IdfZk8U
LV+/3a2fPNgMNUA0o0fb0fvHJd0pcBfXrGq6M+bCVcFjHIgMaDa8j9vnhwYnDbVwpl/oJtZnvExv7qtXGrzhhWrdltvu9RJ85Jzh8VIjdLl+0cdX
/iPmVbgQkg5DagBPL5tjVw2bbx69+uMdv+fMqK3ivSYT/S7Ze8zRMco4pSDShZB/V+cEF3SV8G9M+PbBz1YSs1RX5ayIDGw2zO/b64+WNw21cK4c
DI+2OsrPa4UEnM7hLC7K33nnL7qt4rhBh/eE7JSybHkqIFbHnUyW1AMUhWf6d/64XeK5ohnoFVm9uez99PoQVyS90cVAwnk13yvev/jlINKC7qej
9HHD2r95du6MuAYijrH7ZSbOSTDGgargvzhl/GzRLrbesKgT1Gqy2i4lDXsG9pvTdBSkJMJJ6f/bo3HSu3AaVwidXfE0+FR9N+77IXzT4I/kvFSt
2ZOedgnqaVhpJqQL98wujv89RvhE0cUunLgPoM33gaUGY0zHZwV7BKeXBklcznmfb45jLijIiyrlx89EvEJxFwvx8Z4Pn8vO9LS3eU0aa5YVZZ/7
5+ct2y3l/CqLetKwRfcASBCIMoCeYzoKkP3kSHqGXuK6InmaCwz4sbO/Re5jdZ8vLO8n4kv4/VEFswg8h95d4nXuLhaZvKz86zwUqJKOpQSAOhqM
YfHnyl7IPnmmozuZ8rVmDsTfL9Mm48+OjG8n2pUoXXJTc5DShuYcsYEbQTkk9dQUERtw/0q9/EebgqGHfdvtj7rO5dKwI45TmmPzcv/uIfja56l5
soA9qW/Nqm8y+yQ4r7NaC4SbjpybOigIDVV124DNgvzJVki1uGh833MCDqvhIuTjGJihtuXjHmdTGRtKRh1Nte/TN/1SGLetu62li3/k2zHaKowS
DwrYBx2XQx0HXBFT8Zu1pwXdBMBM88z1/VicCQ52b5qzVhELjvcMFZ7REFN2Jd7gScWH3VB2qiPUkP8r03rn6eWT6jBQMi1WRO+6Y/v5iI/7vvXV
tGpaeC4SHYrS2PGRckv9mCRduCJy5t5DSg6zfYvwmBOzGsWWs/39wnsfC7UFDH7Yu8zQqrGEyKASUX3e2HMbZLht6AMpD6wq/bxN5ETK1VC2bW1g
3iEwnEhHKDVSWEKLJQ3A1zSoPFUG9bVxusAKx2zCT37UsfyGJoMuKscIYBCv+7n33StQcnQUwKb7q1PV+uEgNFFZhSwRGCHz8irC4wq5SY2AVWrl
Y/m7YcTD1atXp3OZmxXuOKZ8KtQ0x97ZCKPZX++5S55So1ltdef2uMJydE/Y4bFTkNqC+oM9aI4K2wCj6wCIidODCBcvv5xncWjurH/TtKMaBECr
v9dL2lM/QGlFJOVttQMXBwXSlwikFoZZG6uBEp6fUdoxYkM/JQfvrz0QMwq5EY6m3raoOZqNXu0qvz5Yu1wQPomzjxm3RRQSU37Fm7n52cWa4NQ+
r+rU31IDK2L1IqNBEYGML2HWNn+4h8WlYw3OPW330iX68qYk6lGlx81uuahDGKcQuTZ+xLhWoSK6amdJAYchpQ0tisvnCdx+LuJVaKKTS/bmRF7A
Yk4XVg7KtT/3ulFAPKqtbthzG6S4begOI6tdoiaiFWx3pZf69E7pP7o7bhm+2FigHCzYlClEYMxR2Aud4Y6m3ragL3qC9X7NHHB1uMnYHB+hyVmG
CQ7jNR4VAF9zMT9MD+nvvvIcJgYrzwiqRM7JA1wFlg5Kz2jBExHpm6LrnypXIVzThh/T1E1Qq+DfZOWbVafSC+P8YOa3+OIJtc5DSgbg5kJba3Yv
n2D+zfYLlTx9wHv6O1uMHd7uK+BqUYUCbNbj9UN+uSKAOK24xSa5V8pwnw7/XXVfJV+O/HgFauJAYPfvRH/rKp/Jz1QBZ+TGZDFDq3QrRRzEPB/k
/ToeHlHusD68o8SQKIOJqHmztkUIspKefOOOk/hnrbupogSMpqgyrX9KCg2pWVQ1YXkwoCZnKtGHJCUIb7edkvdI9/I9Gc54frwO3fNCHEq/Gfx4
w2YASWXHvT8jIAiepNbV6bkMozb701thIWcmuc1t+b+7XJaCGrMAclGRl46QMPTZtbmue/jT89bE+8Mf4cD8j5uhU7RlbbRNJi4BlnHv9rvlVNTg
Pq2ysO+T5cJl1je0CqyNk2iIVw25L2ZgV9ua9DKQbUWkwRaNclUQbUpR/DV6fffu4cGRTayr5VEADgM2QhdWRxrzUFdWAAfVGoFBawcp7qiol0w0
jX27IyjKHetq4mLDdJpXqA1KBgzy1kn49gE77b0AMOBgFG0fqslTiejdWkC9fsy8F1xfmOGDsX9vB89U95PnDRNsE9brWtC8PpTlyYzLr3rU8YMy
KHXVj9XnSFf3+Ua27DLfmGcDtvfaP5/iBQJ+d9ntOA+rY0tiXc89ZBFYiDuNsgmMd6Jq17GdzXoRQDykH8bj8p++IxHBIGNJbrp4zde0KEyLcxZ9
7YkOxKv7m/fcOaoIWmA58ZVENxb3Iexhjqbetqgnwyqw0r8tgAg1ysXL07+BBtJhl8bzzk481vR+4X6zMiBqCmCR21gnG3Wm0rtZV2nPG8f0umba
DElW+1q9RmxHwrnEq4T5sGPwy4z/tAtZ+me9FXwzsm/iAXOQ2oL9o5JbDPV5GIlXsbbnAjYjvFh/talGJArSISXR+S/4E0oOSPWe+Hi6YN394Abj
E3CtKHzwghBtUQeC93o59BKYZ627qa8O05E4wxnpDfG1O4mxUpe7bjZfTBd9+jEDehNVBVOvbUuWUkbxNcu+8ggBBjYrVDIbnATA6stpVCAvvY37
Laoibjy1aD1SZm8pqLnAbUF5vQ9ltY+LwjuNGbYJXptmFy2xSL4L4epRhQApzcfrqOyP9AGlD2NKz3a4tixRVitUNLXrbdExbs3VkJzyjkmjGeVg
CKz0Xiw21fqQnfb4VrerxhoCC/1zeQWOLgA+t0ikHg+y4ZcEJOMyJH0mqPQgvq+yZE7Ke1YbVDoZG8FFS32uO6DpmMsgpmImT10RQ4w8lAVhtfkG
OX5DSgVj8oZZJ4/+/jx2SeP+tYw7aTkdzXoxQDCq5rn/sDD1VXfCANKAEC1vuRKN2EFeWFMLyxN+mPfmqX3PJdMwMKqxL28aybNLaMH7h9VGrCVw
GIQW+iwD4Wbvq26OtKDOTj4dL13ffcGyGw6rbao9AzHH2rsZDf02qHQiPa3TSsSjEVtS681C471+SCZg/TaoOowjNIchpQX9pSzNGSFd4VaMOKzj
oFhtq0bWOj2walGlA8D9y79b4IHMW7yGQxNIo4lEPWgPJFUl2ElRqrjONPn1u6GZuiZ1ETnEhgtYk9fQYIfLL68l0zszvr9axWSHtImWVa/aEmrI
161Db5GA/wNOASD9kfJX2kydVFdDe/TayaKvzwd+uzWKH+Fr6JIAqtxkTlIKt9yCkWVjsUWl7P9wyEbmOAL9ilba/uWxfaucyIil5twUS5yGlAfW
NbPaI4FF9Oqe97xSXven74/5DAzHqbSqkGFPjtyHqwJhT99GtxSBlQSvMxy/C5hkLUhC/8n1qZscRes2VACTShtmnTwN+E21+lJkjJsdogK9pzIk
CpRarstCX3NwXd+Aa6NjXdw/3e90yarPYoFN/EDLVhtUGh9Hv94FsptfVBdK/VrhLeC/aBSVnrkpwG1HeMWUwd3ArKP1p9WUXpuQ+t3F1qLofgtk
P9Gjo8MPSRu1+XQ8KA4t4kgrap3meomiC6z2rDnkossdfcNNDhQq5KqMVg9YWasIIEKpFrO5EhyOCb3/WvPBg0fZVWJqIBBas9isFqj9qw2qBQcl
x/P32EptNQn7qr4/lyGlBfjm2oO7+Oz73OvruLVWCHILbQib3bFvVj6LAFvnz/2AM32hh6A8p+BrUVrfYpasLnIglxU7ikXDOuMfLQCDF3UxiIfI
wtqF1vX1cTvhqoiOpMEx32gzihwfr8yFc7UwZ0LFYbFEqO6++nD1ZOBMv49g5dMfhSozcnOQ2ozy2dYvwpKWn15e4hkpPrFs9A1DHmNwiR2xaxpe
K+LiRQ0zpkgpgRjDzpSyHBaTEMtQHFYDTVzGRfjv0MqmIsG/nhycEkJ4pBNOXGtV6P6DeWfr87KwKT/ckY3PaVmuAGdA/GrTQeeXQp4Gr+wnH27+
87P5DBiXsmVuyd+uj3HgCrPYrBao/asNqgUHJcf44+8KpYhRUsMRjnTM7nHrHeJwWrihMKSQlyYQVp9eUWZfAdbN+F2tKMSNtyT+Pi9+oyyn6oDe
iNt9gBBCzf+bKbZospN64vPYKk9lhyfHfuKRKUcYuh3rauNqwi+Ijk92E5SowcUOsaWNnnnqX6yHEPWO1RDFZ71IbVBoWS4/pz9AHctxQetyLcXf
GaiVhM3vddEB0bA/e/NdZRucp9Nppy2nOrzYBa6UldHyoxtAYUv75Vko9/CykgnHLj+vZAFyac1J6SZ+Vym5MLiislhnrbutrw5Q4ia49kIoLXwP
r8yHc+ZW5yPD9WexSD1R61YbVBoeS4/hx91MGtSw5p27YQdbVzkNOAzvMUlsCdGoOv3jVBlC4TyYBanyX2kJBYhtaA+qLvQupgQsqNS1Frq01IeS
sXX6msmO+O0oH1tqy8Y6i3rauNpn1QXKApKTWDpqkUXcyRTf2Q4/mx2qMYrPaoDasNCiXH9efowwdF5X11lknzybEKzWlAmfhZfRGdG/NM+iLtrU
nDRDKgBA+5bVK212IYWgPqSymhYHsIKTcuxtlqQ55pLJxsb/UV8925oea4H2Oot60rBF9JQzQMZ3Gyf259dkRgxCDI8fxY7VEMVnvUhtUGhZLj+n
P00QQnmbin41RixdcvOQ0o+PYuQ8dF8JWItA4R5zliwtmPrPe6avLSMa9rC+CytmU4ZLtLhtaA+maGoUnDHHNltW+CI42sNhiuHS2nV9Rhs9uXsx
Xz3bkFudmDjKHetq4Qmk5lIZ0o9HSc8cJXzo37alDkeH6s9igGqz1qw2qDQslx/Tn6aMMXa/Hgo3FBexa5Dah1RB4KPeWJnGkrHZAMhK6OBHTfC6
WksWDg3X5iy5vGMrQG1Bd6HpJ71FTIuglmQL5UlpiDsCm8YPWBYr47ymTV23IOZAz1tnWFgEvonAv8+0TUxu3afdIPN3m8CT+4It9hyW3keH6s9i

```
gGqZ1qw2qDQs1x/Th6aMMXadpUii6U3AaUiabVH9W4QgqdEEhitaeyTidY75d1QDn70+0nJQI5hqEioL5UCvik28DYWWiRG74EY4kXQmY71FziBS
OhmO/OTZym3xjqbesKhSotTfVmWRGnnIYfSkoBC1i6bKc30XqQ7uccz4/VHsVgtUdtWG1QKDmuP0cfbbDStN4j5GzMNnIbUPAdrM1h6k2wQPCNTV
2u5Kz3SzGgnGHr9kPAY5cMrQE95Rw7/5PI1CY4ysw3eKK2FAaiunwHclMqbv4Cv8Eg+KatzGDMd+fWeOUA4hjqbeuKgTq8Vh91XT5pSbFmbdxeBJ
GATW2qgAlW4ZzRGlIwm1UzM1JfST9qJePCGhQ5nh+rPYrBao/asNqgUHJcf2ofbDGEBNY0eVpynPLRhQH1ndLDuOWb0LKtRLSu1Q6DHBOEFIv1ni
kGlLHbHdtvvyu+TnkMQ2tAfYWo+QI5WNWFL5cZoy8goFKIC5hoNKttJdyAuGnJDyXdZtGL28ui123vjWK+O3evwgozb6Letq4YeLBuuNkusu+KyQ
N7MUQ34m5hQkFwAMaS2qS4yqbcvqo47rSDK2uK9/twD4NmT5tTTFgtMEmiT4pK8/1TwQR324meA5grWXU+uyTH82O1RzFY7VEbVhsUSo7rT+mDe/
aIo2eVk3Aq1/gS7ZlM+Vz9bJe0TbZD6MKAcn1nn28HWeIxoi50tZ/Ja5kINJW47KqMYYX1nqml/Phd6v1MujbujORYhtaAMjA2rSQvuHhxeZgqxd
aZhRx1YlilkZB9Akrq+Q4iDtWZ59k3eOh3x43vroRDj8GqqLetKxYGEo5Ts/rqRxhcH9feNLZQdD+6cvLS8jsdJDmeH6s9isFqj9qw2qBQclx/Sh
94NuqvZRxh8kmwEEUHSJeggpkvUBER8ZmDLgwoMIGk8IHVN+KaqcblC6asxIS2a6z3TTWgZ5w72n2Nh6pLhtaAgltIIFTcNL5jkELPhSTqtlopxY
o9WF8xAmaHITA7dttyTTG47SulwlmKMYdZt4lAMUovumDo3JlmP7ps0pLgwtc5yfH8WO1RDFZ71IbVBoWS4/pT+rDOrYwRxeVzrD6hKwMKeOJ820
0hYhGScpZoLNZ7pxpQt871MSfP6f2lG4bagPLj+g7U9Yl9LlwWCxbZaRjM3ELh+LRLr15i9mOJWR8PLwbAZ0BDN70x4G7bmHMUwW1fqR/4/nyl/l
L00tKxwQt8702rg1DhzuIeyzUYxpLj+bHaoxis9qgNqw0KJcf1p/TRj/eInO+cx/Z1aUCB7ZKQrSNXjGsp54imYL1/qgF1gzyJd+iSoTaggBFlJd
rkzq1E8QIq8gMDvfUa9itjISey6YBe8kQJpKkn9foM6LQnNvde0QzX4bYNzfWqcNtXygH7JrhFU1ekzJyZ6fvOY8SIPjByf/gGoCbh2sIFnHLMU0
5yPD9WexSD1R61YbVBoeS4/pQ+MCqxRpToVFI8crv5uzagQOBj6IlVLDC6uM4mrM+RakAZc+r9YBe6pC8DyiDJjeoqdtM5Rz8EpLB3wV4G0anMuq
gNy+Y4Je/clSWzK+s9+znTklUhEaFEpjK449JcsHi7GaTgK0VoBUBZWJV0Lrkyrualdf0oJ7hg+U7Yh8ZbwKkXBD9VriUmF/xOJ501v5z13n3v2v
L1GMgQeNBJXXp42qZy9klgCOkz1YQKA467lwGE/ZyZc7Z2lhAeS4773mqPYrDaozasNiiUHNffTx9Mzij/xn1Hgfhqz5CtA+4h9tPYFiGyvitXJq
f1WJ+/C/hOeAZ41qprZZzk2tnzJUJ9UKvOOtb1N2UzNHGCEyhIwZou6cuAinSIDqv/0JVWRYTGuzlbPBScjyqEEB80WGnXx0NEClKXyICOE5brhp
lvzP7K5OvHFnXgINyJssISQohBgPfPOhvad0B+LmRAxwH2Qt0fGpGnGAOHhFv9HHbkc+UKF9c17lAhhDjUeOzJzWV1JaLorZQdtodyHn5uIQM6Dp
Cv6f7YiOIQsfhOoa/EyQpCCHGo0ZZGeO/UblefIAM6YNiot35sROWiWAhuuOZGf1EBGVAhxKGIz4CyFcbKdBCRxDKgiVAyjvM4if4NgVzDh6Zt9K
bbpJwpWocC6Rxv5tbHlQEVQhyKuAaUoMrrp6zoK5MiFhnQROp5n6TO3Dd1fbFw8fbS506VIIozE8zDgbwYzqYi8iT0p6w+LZh1UU2J/las2lVs2j
Kxz+UUQogUKMlIURxSFHe/OT7ZBzKgCZDXGVK4IVS56moKIYQYHDKgCVDJxDKEKaLgghBCiOFDBjQBcossYxgj/PXzXlCKiRBCDCsyoIlwQjvlrz
50uH04s0/Hnjq3DEAa5MHNQggh8iMD2icH3n2/WLlqd1kE/tY7VxdXTV5a1nw9/6JF5X8paECR96dnvDruxcuFEELkQwZUCCGESEAGVAghhEhABl
QIIYRIQAZUCCGESEAGVAghhEhABlQIIYRIQAZUCCGESEAGVAghhEhABlQIIYRIQAZUCCGESEAGVAghhEhABlQIIYRIQAZUCCGESEAGVAghhEhABlQIIYRIQAZUCCGESEAGVAghhIimKP
4XBcAIzFfvoBoAAAAASUVORK5CYII='

img_data = base64.b64decode(img_str)
# 这里b64decode的意思是把base64编码转换成ascii码。

with open('1.png', 'wb') as f:
# wb+:以二进制格式打开一个文件用于读写。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。
    f.write(img_data)
# 这里这里write的意思是写入的意思。
print('successful')
```

这里出来了flag

MMS

提示：工业网络中存在的异常，尝试通过分析PACP流量包，分析出流量数据中的异常 点，并拿到FLAG，flag形式为 flag{}。



2953	648.890928	192.168.142.148	192.168.142.133	MMS	124	189 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
2954	648.891215	192.168.142.133	192.168.142.148	MMS	86	189 confirmed-ResponsePDU
2969	653.891024	192.168.142.148	192.168.142.133	MMS	124	190 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
2970	653.891310	192.168.142.133	192.168.142.148	MMS	86	190 confirmed-ResponsePDU
2988	658.891109	192.168.142.148	192.168.142.133	MMS	124	191 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
2989	658.891390	192.168.142.133	192.168.142.148	MMS	86	191 confirmed-ResponsePDU
2992	663.890466	192.168.142.148	192.168.142.133	MMS	124	192 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
2993	663.890797	192.168.142.133	192.168.142.148	MMS	86	192 confirmed-ResponsePDU
2995	668.890812	192.168.142.148	192.168.142.133	MMS	124	193 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
2996	668.891110	192.168.142.133	192.168.142.148	MMS	86	193 confirmed-ResponsePDU
3052	673.891272	192.168.142.148	192.168.142.133	MMS	124	194 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3053	673.891547	192.168.142.133	192.168.142.148	MMS	86	194 confirmed-ResponsePDU
3061	678.890597	192.168.142.148	192.168.142.133	MMS	124	195 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3062	678.890815	192.168.142.133	192.168.142.148	MMS	86	195 confirmed-ResponsePDU
3068	683.890492	192.168.142.148	192.168.142.133	MMS	124	196 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3069	683.890778	192.168.142.133	192.168.142.148	MMS	86	196 confirmed-ResponsePDU
3071	688.891141	192.168.142.148	192.168.142.133	MMS	124	197 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3072	688.891403	192.168.142.133	192.168.142.148	MMS	86	197 confirmed-ResponsePDU
3091	693.891030	192.168.142.148	192.168.142.133	MMS	124	198 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs

```
> Frame 3189: 124 bytes on wire (992 bits), 124 bytes captured (992 bits)
> Ethernet II, Src: VMware_1d:07:17 (00:0c:29:1d:07:17), Dst: VMware_75:b2:38 (00:0c:29:75:b2:38)
0000  00 0c 29 75 b2 38 00 0c  29 1d 07 17 08 00 45 00   ··)u·8·· )·····E·
```

这里先分析一下mms协议

MMS.pcap

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

mms

No.	Time	Source	Destination	Protocol	Length	Info
3300	759.063405	192.168.142.148	192.168.142.133	MMS	124	211 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3301	759.063635	192.168.142.133	192.168.142.148	MMS	86	211 confirmed-ResponsePDU
3316	764.062874	192.168.142.148	192.168.142.133	MMS	124	212 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3317	764.063174	192.168.142.133	192.168.142.148	MMS	86	212 confirmed-ResponsePDU
3325	769.062872	192.168.142.148	192.168.142.133	MMS	124	213 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3326	769.063098	192.168.142.133	192.168.142.148	MMS	86	213 confirmed-ResponsePDU
3335	774.051361	192.168.142.148	192.168.142.133	MMS	124	214 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3336	774.051678	192.168.142.133	192.168.142.148	MMS	86	214 confirmed-ResponsePDU
3419	779.069071	192.168.142.148	192.168.142.133	MMS	124	215 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3420	779.069408	192.168.142.133	192.168.142.148	MMS	86	215 confirmed-ResponsePDU
3431	784.076631	192.168.142.148	192.168.142.133	MMS	124	216 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3432	784.076825	192.168.142.133	192.168.142.148	MMS	86	216 confirmed-ResponsePDU
3438	789.068864	192.168.142.148	192.168.142.133	MMS	124	217 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3439	789.069121	192.168.142.133	192.168.142.148	MMS	86	217 confirmed-ResponsePDU
3441	794.082668	192.168.142.148	192.168.142.133	MMS	124	218 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3442	794.082964	192.168.142.133	192.168.142.148	MMS	86	218 confirmed-ResponsePDU
3455	799.098713	192.168.142.148	192.168.142.133	MMS	124	219 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3456	799.098906	192.168.142.133	192.168.142.148	MMS	86	219 confirmed-ResponsePDU
3460	804.114113	192.168.142.148	192.168.142.133	MMS	124	220 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3461	804.114385	192.168.142.133	192.168.142.148	MMS	86	220 confirmed-ResponsePDU

然后我们一定要相信一个原则，事出反常必有妖。

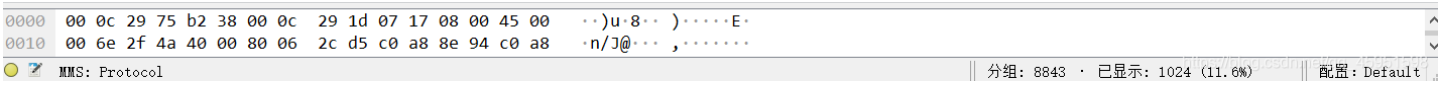
分析mms的包发现，这两个包的内容有点不太□样，后□似乎是16进制字符串

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

mms

No.	Time	Source	Destination	Protocol	Length	Info
3133	703.891295	192.168.142.133	192.168.142.148	MMS	86	200 confirmed-ResponsePDU
3143	708.890659	192.168.142.148	192.168.142.133	MMS	124	201 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3144	708.890842	192.168.142.133	192.168.142.148	MMS	86	201 confirmed-ResponsePDU
3156	713.890993	192.168.142.148	192.168.142.133	MMS	124	202 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3157	713.891286	192.168.142.133	192.168.142.148	MMS	86	202 confirmed-ResponsePDU
3171	718.891063	192.168.142.148	192.168.142.133	MMS	124	203 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3172	718.891348	192.168.142.133	192.168.142.148	MMS	86	203 confirmed-ResponsePDU
3189	723.890659	192.168.142.148	192.168.142.133	MMS	124	204 confirmed-RequestPDU IEDRelay1 LLN666i5250356j4249
3190	723.890985	192.168.142.133	192.168.142.148	MMS	86	204 confirmed-ResponsePDU
3196	728.891287	192.168.142.148	192.168.142.133	MMS	124	205 confirmed-RequestPDU IEDRelay1 LLN6167325579683 56j
3197	728.891562	192.168.142.133	192.168.142.148	MMS	86	205 confirmed-ResponsePDU
3213	733.891417	192.168.142.148	192.168.142.133	MMS	124	206 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3214	733.891705	192.168.142.133	192.168.142.148	MMS	86	206 confirmed-ResponsePDU
3229	738.890584	192.168.142.148	192.168.142.133	MMS	124	207 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3230	738.890760	192.168.142.133	192.168.142.148	MMS	86	207 confirmed-ResponsePDU
3241	743.890765	192.168.142.148	192.168.142.133	MMS	124	208 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3242	743.890958	192.168.142.133	192.168.142.148	MMS	86	208 confirmed-ResponsePDU
3247	748.891442	192.168.142.148	192.168.142.133	MMS	124	209 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs
3248	748.891624	192.168.142.133	192.168.142.148	MMS	86	209 confirmed-ResponsePDU
3275	754.092560	192.168.142.148	192.168.142.133	MMS	124	210 confirmed-RequestPDU IEDRelay1 LLN0EXOpTmh$cdcNs

```
> Frame 3196: 124 bytes on wire (992 bits), 124 bytes captured (992 bits)
> Ethernet II, Src: VMware_1d:07:17 (00:0c:29:1d:07:17), Dst: VMware_75:b2:38 (00:0c:29:75:b2:38)
```



不过□存在i和j这种不属于16进制字符串的字符，想到i和j是连在□起的，第□位66转成字符□是f，□6c的话转陈字符是l，于是将i替换成c，j替换成d，脚本转□下

```
# coding = utf-8

s = "666c5250356d42496167325579683 56d"

flag =" "

for i in range(0, len(s), 2):
    # 表示从0开始，长度为s的字符串，步长为二。

    flag += chr(int(s[i:i+2], 16))
    # 这里chr的意思是把里面的内容转换成字符串
    # s[i:i+2]这个表示在是里面取是里面长度为2的内容，因为一开始i为0
    # 后面的16表示为16进制。


print(flag)
# flRP5mBIag2Uyh5m
# flag{RP5mBI2Uyh5m}
```

有待更新。。。