XCTF-攻防世界-密码学crypto-高手进阶区-writeup

原创

Ryannn- 于 2019-10-25 18:47:38 发布 5155 收藏 21
分类专栏： 密码学 CTF 文章标签： 密码学 CTF WP Crypto 攻防世界

本文链接：https://blog.csdn.net/Ryannn_/article/details/102745063


密码学 同时被 2 个专栏收录
4 篇文章 0 订阅
订阅专栏


CTF
2 篇文章 0 订阅
订阅专栏

目录

0x00 告诉你个秘密

密文：分成了两段

```
636A56355279427363446C4A49454A7154534230526D6843
56445A31614342354E326C4B4946467A5769426961453067
```

Hex->text

```
cjV5RyBscDlJIEJqTSB0RmhC
VDZ1aCB5N2lKIFFzWiBiaE0g
```

base64->text

```
r5yG lp9I BjM tFhB
T6uh y7iJ QsZ bhM
```

在键盘敲一遍，每一组包围的键就是密码

```
T O N G
Y U A N
TONGYUAN
```

0x01 cr3-what-is-this-encryption

Fady同学以为你是菜鸟，不怕你看到他发的东西。他以明文形式将下面这些东西发给了他的朋友，他严重低估了我们的解密能力

```
p=0xa6055ec186de51800ddd6fcbf0192384ff42d707a55f57af4fcfb0d1dc7bd97055e8275cd4b78ec63c5d592f567c66393a06132
q=0xfa0f9463ea0a93b929c099320d31c277e0b0dbc65b189ed76124f5a1218f5d91fd0102a4c8de11f28be5e4d0ae91ab319f4537e
e=0x6d1fdab4ce3217b3fc32c9ed480a31d067fd57d93a9ab52b472dc393ab7852fbcb11abbebfd6aaae8032db1316dc22d3f7c3d63
c=0x7fe1a4f743675d1987d25d38111fae0f78bbea6852cba5beda47db76d119a3efe24cb04b9449f53becd43b0b46e269826a983f8
```

题中给出了p、q、e、c，符合RSA的特征，根据已知信息算出解密密钥d进行解密即可

```
from Crypto.Util.number import long_to_bytes

p=0xa6055ec186de51800ddd6fcbf0192384ff42d707a55f57af4fcfb0d1dc7bd97055e8275cd4b78ec63c5d592f567c66393a06132
q=0xfa0f9463ea0a93b929c099320d31c277e0b0dbc65b189ed76124f5a1218f5d91fd0102a4c8de11f28be5e4d0ae91ab319f4537e
e=0x6d1fdab4ce3217b3fc32c9ed480a31d067fd57d93a9ab52b472dc393ab7852fbcb11abbebfd6aaae8032db1316dc22d3f7c3d63
c=0x7fe1a4f743675d1987d25d38111fae0f78bbea6852cba5beda47db76d119a3efe24cb04b9449f53becd43b0b46e269826a983f8


def exgcd(a, n):
    if n == 0:
        return 1, 0
    else:
        x, y = exgcd(n, a % n)
        x, y = y, x - (a // n) * y
        return x, y


def getReverse(a, n):
    re, y = exgcd(a, n)
    while(re < 0):
        re += n
    return re


if __name__ == "__main__":
    d = getReverse(e, (p - 1)*(q - 1))
    n = p * q
    m = pow(c, d, n)
    plaintext = long_to_bytes(m)
    print(plaintext)
```

0x02 flag_in_your_hand

附件是一个html文件和一份js代码

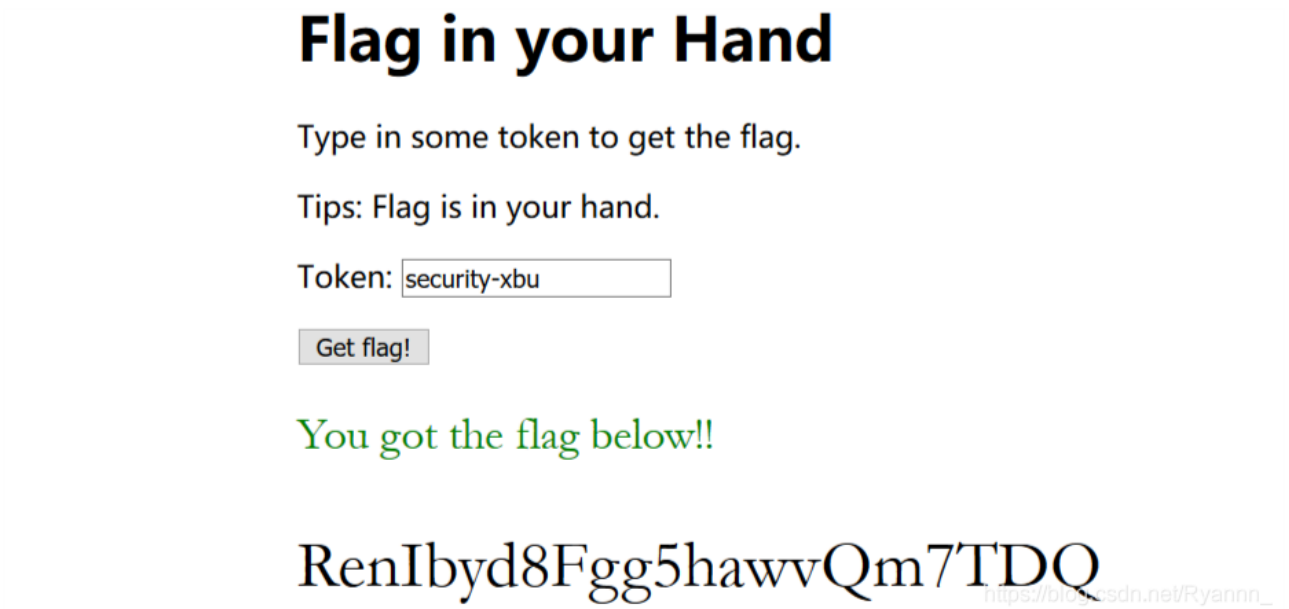
js代码中的核心部分如下，需要使函数饭回true

```
function ck(s) {
    try {
        ic
    } catch (e) {
        return;
    }
    var a = [118, 104, 102, 120, 117, 108, 119, 124, 48,123,101,120];
    if (s.length == a.length) {
        for (i = 0; i < s.length; i++) {
            if (a[i] - s.charCodeAt(i) != 3)
                return ic = false;
        }
        return ic = true;
    }
    return ic = false;
}
```

通过Python解密得到字符串s

```
#!/usr/bin/python
# -*- coding=utf -*-

a = [118, 104, 102, 120, 117, 108, 119, 124, 48,123,101,120]
for i in a:
    print(chr(i-3), end='')
```



0x03 Broadcast

题目描述：粗心的Alice在制作密码的时候，把明文留下来，聪明的你能快速找出来吗？

本来以为题目又开始忽悠了，没想到真这么实诚

```
msg = 'Hahaha, Hastad\'s method don\'t work on this. Flag is flag{fa0f8335-ae80-448e-a329-6fb69048aae4}.'
```

0x04 你猜猜

我们刚刚拦截了，敌军的文件传输获取一份机密文件，请君速速破解。

密文：

```
504B03040A0001080000626D0A49F4B5091F1E0000001200000000800000006666C61672E7478746C9F170D35D0A45826A03E161FB9687
```

504B0304开头，怎么看怎么眼熟，其实是压缩文件的PK头。使用二进制工具WinHex，粘贴进去，确实是压缩文件

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000090	50	4B	03	04	0A	00	01	08	00	00	62	6D	0A	49	F4	B5	PK bm Iôµ
000000A0	09	1F	1E	00	00	00	12	00	00	00	08	00	00	00	66	6C	fl
000000B0	61	67	2E	74	78	74	6C	9F	17	0D	35	D0	A4	58	26	A0	ag.txtlŸ 5Ð¤X&
000000C0	3E	16	1F	B9	68	70	ED	DF	C7	C8	9A	11	86	2F	91	99	> ¹hpíßÇÈš †/'™
000000D0	B4	CD	78	E7	50	4B	01	02	3F	00	0A	00	01	08	00	00	´ÍxçPK ?
000000E0	62	6D	0A	49	F4	B5	09	1F	1E	00	00	00	12	00	00	00	bm Iôµ
000000F0	08	00	24	00	00	00	00	00	00	00	20	00	00	00	00	00	$
00000100	00	00	66	6C	61	67	2E	74	78	74	0A	00	20	00	00	00	flag.txt
00000110	00	00	01	00	18	00	AF	15	02	10	CA	F2	D1	01	5C	AE	¯ ÊòÑ \®
00000120	AA	05	CA	F2	D1	01	5C	AE	AA	05	CA	F2	D1	01	50	4B	ª ÊòÑ \®ª ÊòÑ PK
00000130	05	06	00	00	00	00	01	00	01	00	5A	00	00	00	44	00	Z D
00000140	00	00	00	00													

保存为zip文件，本以为解压就结束了结果还要密码，没办法只能爆破了，爆破完直接提取flag

```
root@kali:~/桌面# fcrackzip -b -c1 -u 1.zip


PASSWORD FOUND!!!!: pw == 123456
root@kali:~/桌面#
```

0x05 工业协议分析2

题目描述：在进行工业企业检查评估工作中，发现了疑似感染恶意软件的上位机。现已提取出上位机通信流量，尝试分析出异常点，获取FLAG。 flag形式为 flag{}

附件是一个.pcapng流量文件，扔进Wireshark。

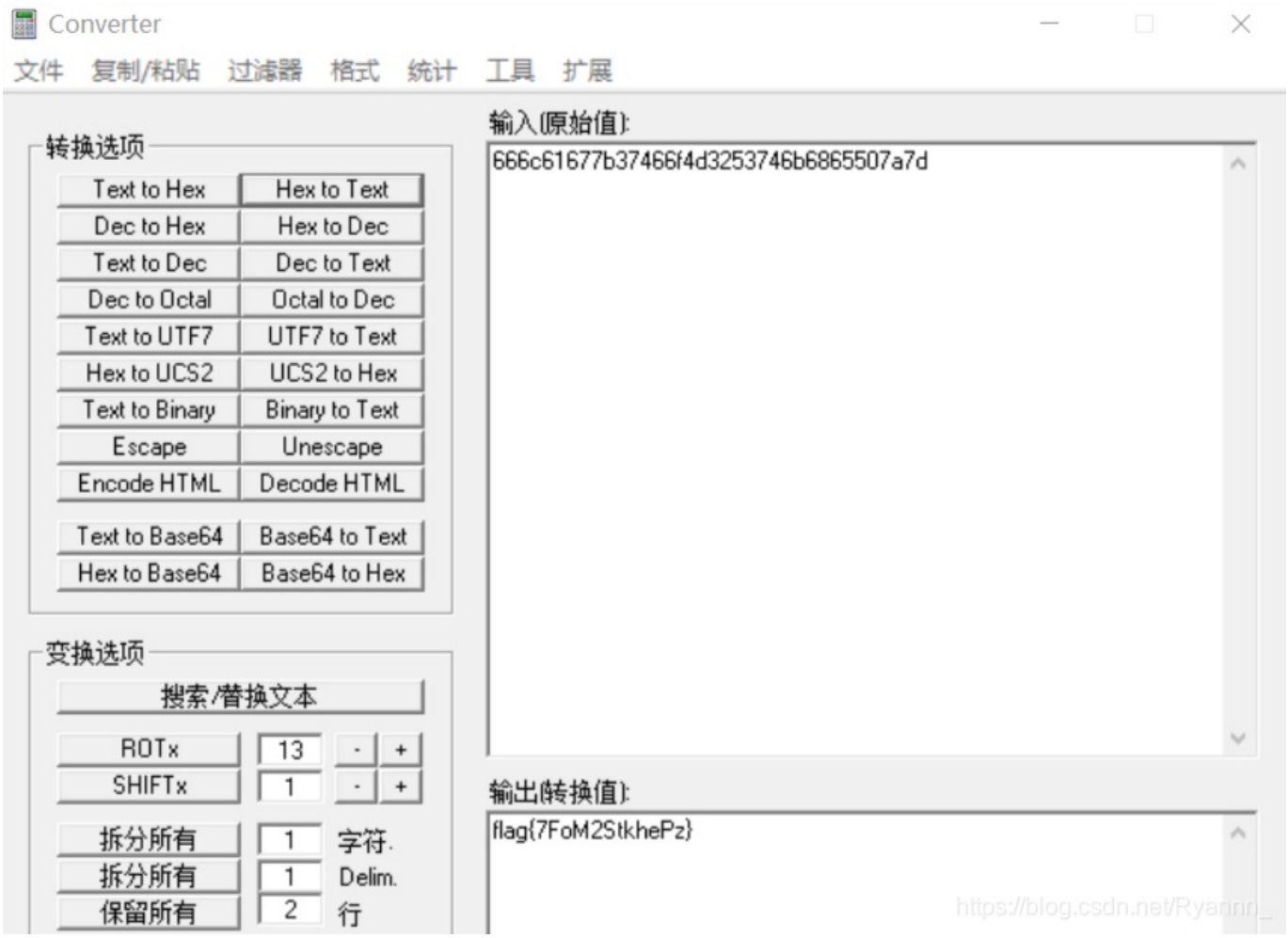
一堆UDP包，按长度排序找到了几个特殊的，其中131和137有搞头。

No.	Time	Source	Destination	Protocol	Length	Info
739	100.436232	192.168.1.181	192.168.1.123	UDP	64	11000→64406 Len=22
743	100.438565	192.168.1.181	192.168.1.123	UDP	64	11000→64406 Len=22
794	143.065734	192.168.1.123	192.168.1.181	UDP	64	64406→11000 Len=22
150	23.965514	192.168.1.123	192.168.1.181	UDP	67	64406→11000 Len=25
175	32.412085	192.168.1.181	192.168.1.123	UDP	74	11000→64406 Len=32
741	100.437598	192.168.1.181	192.168.1.123	UDP	74	11000→64406 Len=32
167	32.404322	192.168.1.181	192.168.1.123	UDP	131	11000→64406 Len=89
203	45.021168	192.168.1.123	192.168.1.181	UDP	137	64406→11000 Len=95
440	65.266055	192.168.1.123	192.168.1.181	UDP	137	64406→11000 Len=95
148	22.492885	192.168.1.123	192.168.1.181	UDP	146	64406→11000 Len=104
790	142.976251	192.168.1.123	192.168.1.181	UDP	147	64406→11000 Len=105
123	15.639029	192.168.1.181	192.168.1.123	UDP	158	11000→64406 Len=116
136	21.026137	192.168.1.181	192.168.1.123	UDP	158	11000→64406 Len=116
733	100.430392	192.168.1.181	192.168.1.123	UDP	173	11000→64406 Len=131
648	91.813108	192.168.1.123	192.168.1.181	UDP	179	64406→11000 Len=137
9	0.329000	192.168.1.243	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
26	1.330596	192.168.1.243	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
46	2.330743	192.168.1.243	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1

```
> Frame 733: 173 bytes on wire (1384 bits), 173 bytes captured (1384 bits) on interface 0
> Ethernet II, Src: 00:e2:36:0b:19:2b (00:e2:36:0b:19:2b), Dst: Vmware_0a:63:9f (00:0c:29:0a:63:9f)
> Internet Protocol Version 4, Src: 192.168.1.181, Dst: 192.168.1.123
> User Datagram Protocol, Src Port: 11000, Dst Port: 64406
> Data (131 bytes)
```

```
0000  00 0c 29 0a 63 9f 00 e2  36 0b 19 2b 08 00 45 00   ..).c... 6..+..E.
0010  00 9f 6b 5b 00 00 80 11  4a 72 c0 a8 01 b5 c0 a8   ..k[.... Jr......
0020  01 7b 2a f8 fb 96 00 8b  31 22 4c 00 4f a1 83 00   .{*..... 1"L.O...
0030  e3 00 25 27 25 27 11 00  62 00 00 00 00 00 00 00   ..%'%'.. b.......
0040  00 00 00 00 00 00 00 00  00 00 00 20 00 00 00 18   ........ ... ....
0050  00 00 00 38 00 00 00 2a  00 00 00 62 00 00 00 00   ...8...* ...b....
0060  00 00 00 62 00 00 00 00  00 00 00 02 14 00 4d      ...b.... .......M
0070  41 49 4e 5f 28 d6 f7 b3  cc d0 f2 29 00 03 01 02   AIN_(... ...)....
0080  00 c8 00 00 01 00 25 00  36 36 36 63 36 31 36 37   ......%. 666c6167
0090  37 62 33 37 34 36 36 66  34 64 33 32 35 33 37 34   7b37466f 4d325374
00a0  36 62 36 38 36 35 35 30  37 61 37 64 00            6b686550 7a7d.
```


Hex -> Text



0x06 wtc_rsa_bbq

附件是未知属性文件，扔进WinHex，PK头，改后缀解压

得到.bin密文文件和.pem公钥文件，kali工具解决

```
root@kali:~/桌面/CTF# cd RsaCtfTool/
root@kali:~/桌面/CTF/RsaCtfTool# python RsaCtfTool.py --pub
lickey ../key.pem --uncipherfile ../cipher.bin
[+] Clear text : Congratulations! Here is a treat for you:
flag{how_d0_you_7urn_this_0n?}
root@kali:~/桌面/CTF/RsaCtfTool#
```

0x07 工控安全取证

题目描述：有黑客入侵工控设备后在内网发起了大量扫描，而且扫描次数不止一次。 请分析日志，指出对方第4次发起扫描时的数据包的编号，flag形式为 flag{}

附件是.log文件，扔进Wireshark，按照题目要求，分出了4次扫描的ICMP包

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.9	192.168.0.99	ICMP	60	Echo (ping) request id=0x7ae9, seq=0/0, ttl=40 (reply in 2)
2	0.000078	192.168.0.99	192.168.0.9	ICMP	42	Echo (ping) reply id=0x7ae9, seq=0/0, ttl=255 (request in 1)
148007	1274.602300	192.168.0.9	192.168.0.99	ICMP	60	Echo (ping) request id=0x1e09, seq=0/0, ttl=47 (reply in 148008)
148008	1274.602365	192.168.0.99	192.168.0.9	ICMP	42	Echo (ping) reply id=0x1e09, seq=0/0, ttl=255 (request in 148007)
150655	1308.472790	192.168.0.99	192.168.0.9	ICMP	370	Destination unreachable (Port unreachable)
150753	1407.256096	192.168.0.9	192.168.0.99	ICMP	60	Echo (ping) request id=0xa373, seq=0/0, ttl=53 (reply in 150754)
150754	1407.256145	192.168.0.99	192.168.0.9	ICMP	42	Echo (ping) reply id=0xa373, seq=0/0, ttl=255 (request in 150753)
153165	1441.428990	192.168.0.99	192.168.0.9	ICMP	370	Destination unreachable (Port unreachable)
155847	1504.127684	192.168.0.99	192.168.0.9	ICMP	370	Destination unreachable (Port unreachable)
155987	1602.084879	192.168.0.1	192.168.0.99	ICMP	60	Echo (ping) request id=0xc77b, seq=0/0, ttl=52 (no response found!)
155988	1602.084912	192.168.0.254	192.168.0.99	ICMP	60	Echo (ping) request id=0xc77b, seq=0/0, ttl=52 (no response found!)
155989	1602.084941	192.168.0.199	192.168.0.99	ICMP	60	Echo (ping) request id=0xc77b, seq=0/0, ttl=52 (no response found!)
155990	1602.084976	192.168.0.199	192.168.0.99	ICMP	60	Echo (ping) request id=0xc77b, seq=0/0, ttl=52 (no response found!)
3	0.000044	192.168.0.9	192.168.0.99	TCP	60	52218→80 [ACK] Seq=1 Ack=1 Win=2048 Len=0

155987到155990属于第4次扫描，逐个尝试后发现155989为正确flag

0x08 简单流量分析

题目描述：不久前，运维人员在日常安全检查的时候发现现场某设备会不时向某不知名ip发出非正常的ICMP PING包。这引起了运维人员的注意，他在过滤出ICMP包分析并马上开始做应急处理很可能已被攻击的设备。运维人员到底发现了什么？flag形式为 flag{}

附件扔进Wireshark看到一堆ICMP包，没头绪，最后只能忍痛看官方WP，居然是把ICMP数据部分的长度全部转换成ASCII

fine，反正我是想不出来，借用官方的脚本

```
#!/usr/bin/python
# -*- coding=utf -*-

import pyshark
import base64

L_flag = []
packets = pyshark.FileCapture('fetus_pcap.pcap')
for packet in packets:
    for pkt in packet:
        if pkt.layer_name == "icmp":
            if int(pkt.type) != 0:
                L_flag.append(int(pkt.data_len))
c = len(L_flag)
for i in range(0, c):
    L_flag[i] = chr(L_flag[i])
print(''.join(L_flag))
print(base64.b64decode(''.join(L_flag)))
```

flag就包含在输出结果中

```
E:\Python\Python37\python.exe D:/CTF/python/简单流量分析.py
Ojpcbm1vbmdvZGI6IToxNzg0MzowOjk5OTk5Ojc6OjpcbnVidW50dTokNiRMaEhSb21URSRNN0M0bjg0VWNGTEFHe3h4MmI4YV82bW02NGNfZnNvY2lldHl9Ojo=
b'::\\nmongodb:!:17843:0:99999:7:::\\nubuntu:$6$LhHRomTE$M7C4n84UcFLAG{xx2b8a_6mm64c_fsociety}::'

Process finished with exit code 0
```

0x09 cr4-poor-rsa

附件属性未知，扔进WinHex，1F 8B 08 00，gzip文件，添加后缀.gz后解压，又是一个未知文件，故技重施得到文件

名称	修改日期	类型	大小
flag.b64	2016/12/11 17:08	B64 文件	1 KB
key.pub	2016/12/11 16:59	PUB 文件	1 KB

尝试了很多次kali的RsaCtfTool，不知道为啥总是失败，只能自己一步一步来了

第一步，提取公钥文件信息

```
root@kali:~/桌面/CTF# openssl rsa -pubin -text -modulus -in warmup -in key.pub
RSA Public-Key: (399 bit)
Modulus:
    52:a9:9e:24:9e:e7:cf:3c:0c:bf:96:3a:00:96:61:
    77:2b:c9:cd:f6:e1:e3:fb:fc:6e:44:a0:7a:5e:0f:
    89:44:57:a9:f8:1c:3a:e1:32:ac:56:83:d3:5b:28:
    ba:5c:32:42:43
Exponent: 65537 (0x10001)
Modulus=52A99E249EE7CF3C0CBF963A009661772BC9CDF6E1E3FBFC6E44A07A5E0F894457A9F81C
3AE132AC5683D35B28BA5C324243
writing RSA key
-----BEGIN PUBLIC KEY-----
ME0wDQYJKoZIhvcNAQEBBQADPAAwOQIyUqmeJJ7nzzwMv5Y6AJZhdyvJzfbh4/v8
bkSgel4PiURXqfgcOuEyrFaD01soulwyQkMCAwEAAQ==
-----END PUBLIC KEY-----
```


得到大素数乘积n，此处为16进制，转换为10进制后进行分解，解出素数p、q。大数分解

```
n = 8338101935649677019123629555397894511398728637945349232597434194230892292064730914084035603111915457642
p = 8636534766163765753088663449845764666449425722469000131569 19
q = 9654453043269981947982822288424847324384571705959995234269 01
```

现已知大素数p、q、素数e，写个python算出来私钥d然后解密即可

```
from Crypto.Util.number import long_to_bytes
from Crypto.Util.number import bytes_to_long
import base64


n = 8338101935649677019123629555397894511398728637945349232597434194230892292064730914084035603111915457642
p = 8636534766163765753088663449845764666449425722469000131569 19
q = 9654453043269981947982822288424847324384571705959995234269 01
e = 65537

def exgcd(a, n):
    if n == 0:
        return 1, 0
    else:
        x, y = exgcd(n, a % n)
        x, y = y, x - (a // n) * y
        return x, y


def getReverse(a, n):
    re, y = exgcd(a, n)
    while(re < 0):
        re += n
    return re


if __name__ == "__main__":
    d = getReverse(e, (p - 1)*(q - 1))
    c = base64.b64decode("Ni45iH4UnXSttNuf0Oy80+G5J7tm8sBJuDNN7qfTIdEKJow4siF2cpSbP/qIWDjSi+w=")
    c = bytes_to_long(c)
    m = pow(c, d, n)
    plaintext = long_to_bytes(m)
    print(plaintext)
```

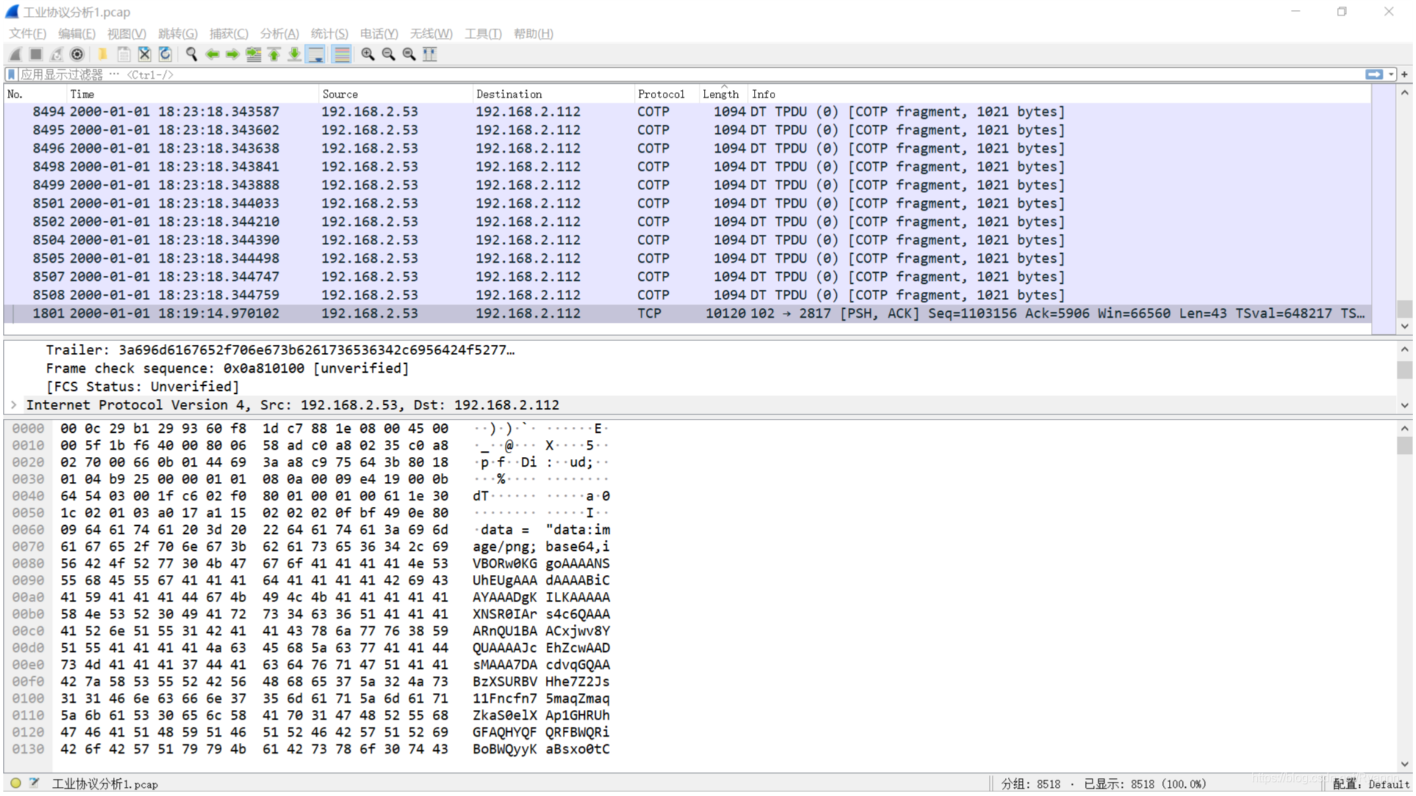
```
E:\Python\Python37\python.exe D:/CTF/python/1.py
b'\x02\x9e&\xded\xa0\x96#H\x8au6L\xdb\xae\x84\x89:\x00ALEXCTF{SMALL_PRIMES_ARE_BAD}\n'

Process finished with exit code 0
```

0x0A 工业协议分析1

题目描述：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。flag形式为 flag{}

流量文件扔进Wireshark，按题目要求的找异常点呗，然后就发现了一个异常的长的包



写的也挺明白的了，base64转图片

0x0B SM

不会 暂放

0x0C OldDriver

题目描述：有个年轻人得到了一份密文，身为老司机的你能帮他看看么？

https://www.xctf.org.cn/library/details/whctf-writeup/

https://www.cnblogs.com/WangAoBo/p/7541536.html

0x0A enc

Fady不是很理解加密与编码的区别 所以他使用编码而不是加密，给他的朋友传了一些秘密的消息。

密文：一堆ZERO和ONE

```
ZERO ONE ZERO ZERO ONE ONE ZERO ZERO ZERO ONE ONE ZERO ONE ZERO ZERO ONE ZERO ZERO ONE ONE ZERO ZERO ZERO Z
```

转换成01再说

```
0100110001101001001100000110011101001100011010010011000001110101010011000110100101000001011101010100100101010
```

太长了，肯定得再短一点，每8位变成一个字符

```
Li0gLi0uLiAuIC0uLi0gLS4tLiAtIC4uLS4gLSAuLi4uIC4tLS0tIC4uLi4uIC0tLSAuLS0tLSAuLi4gLS0tIC4uLi4uIC4uLSAuLS0uIC4
```

base64解码出来熟悉的Morse，直接用新手区Morse的脚本

```
.- .-.. . -..- -.-. - ..-. - .... .---- ..... --- .---- ... --- ..... ..- .--. ....- .-. --- ..... . -.-. .
```

```
alexctfth15o1so5up3ro5ecr3totxt
```

emmmmm提交flag不对，看了别人的wp才知道还有下面这种骚操作，怕是神仙也想不出来

```
alexctf th15 o 1s o 5up3r o 5ecr3t o txt
alexctf th15_1s_5up3r_5ecr3t_txt
alexctf{th15_1s_5up3r_5ecr3t_txt}
```

仍然不对，字母大写，终于对了。来个整合的程序吧

```
# -*- coding=utf -*-
import re
import base64

table ={'a': ".-", 'b': "-...", 'c': "-.-.", 'd': "-..", 'e': ".", 'f': "..-.", 'g': "--.",
        'h': "....", 'i': "..", 'j': ".---", 'k': "-.-", 'l': ".-..", 'm': "--", 'n': "-.",
        'o': "---", 'p': ".--.", 'q': "--.-", 'r': ".-.", 's': "...", 't': "-", 'u': "..-",
        'v': "...-", 'w': ".--", 'x': "-..-", 'y': "-.--", 'z': "--..",

        '0': '-----', '1': '.----', '2': '..---', '3': '...--', '4': '....-',
        '5': '.....', '6': '-....', '7': '--...', '8': '---..', '9': '----.',

        ',': '--..--', '.': '.-.-.-', ':': '---...', ';': '-.-.-.',
        '?': '..--..', '=': '-...-', "'": '.----.', '/': '-..-.',
        '!': '-.-.--', '-': '-....-', '_': '..--.-', '(': '-.--.',
        ')': '-.--.-', '$': '...-..-', '&': '. . . .', '@': '.--.-.'}


def morseDecode(cipher):
    msg = ''
    codes = cipher.split(' ')
    for code in codes:
        if code == '':
            msg += ' '
        else:
            UNCODE = dict(map(lambda t: (t[1], t[0]), table.items()))
            msg += UNCODE[code]
    return msg


def shrink(cipher):
    msg = ''
    codes = re.findall(r'.{8}', cipher)
    for code in codes:
            c = chr(int(code, 2))
            msg += c
    return msg


if __name__ == '__main__':
    file = open(r'D:\CTF\攻防世界\Crypto\高手进阶区\enc', 'r')
    cipher = file.read()
    cipher1 = cipher.replace('ZERO', '0').replace('ONE', '1').replace(' ', '')
    cipher2 = shrink(cipher1)
    cipher3 = base64.b64decode(cipher2).decode()
    plaintext = morseDecode(cipher3)
    plaintext = plaintext.replace('o', '_').upper()
    print(plaintext)
```