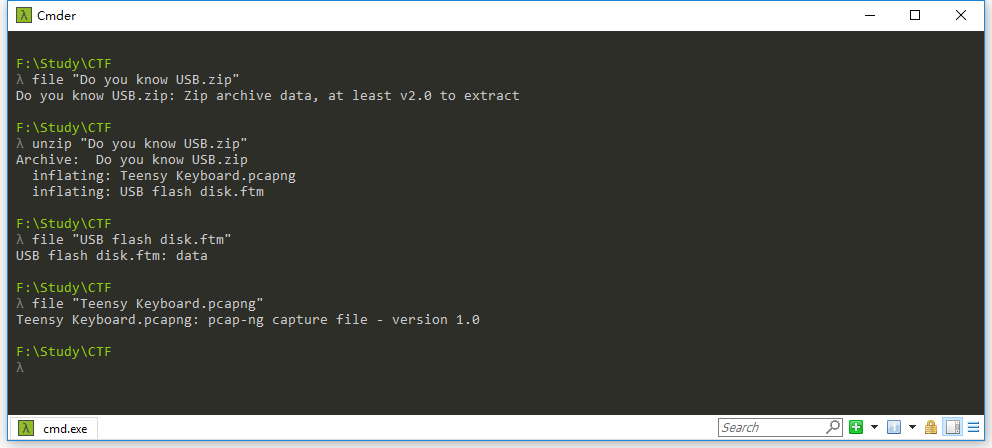
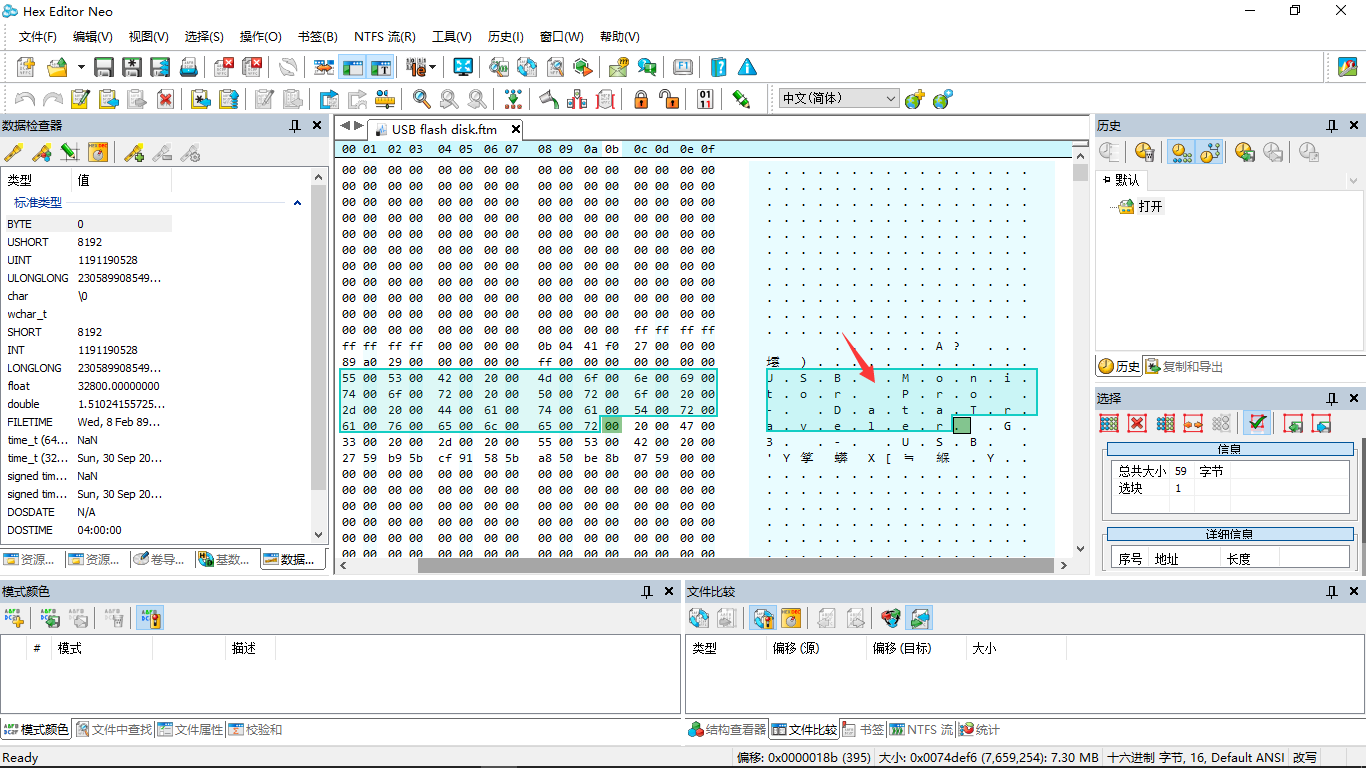
1、提取flag.zip

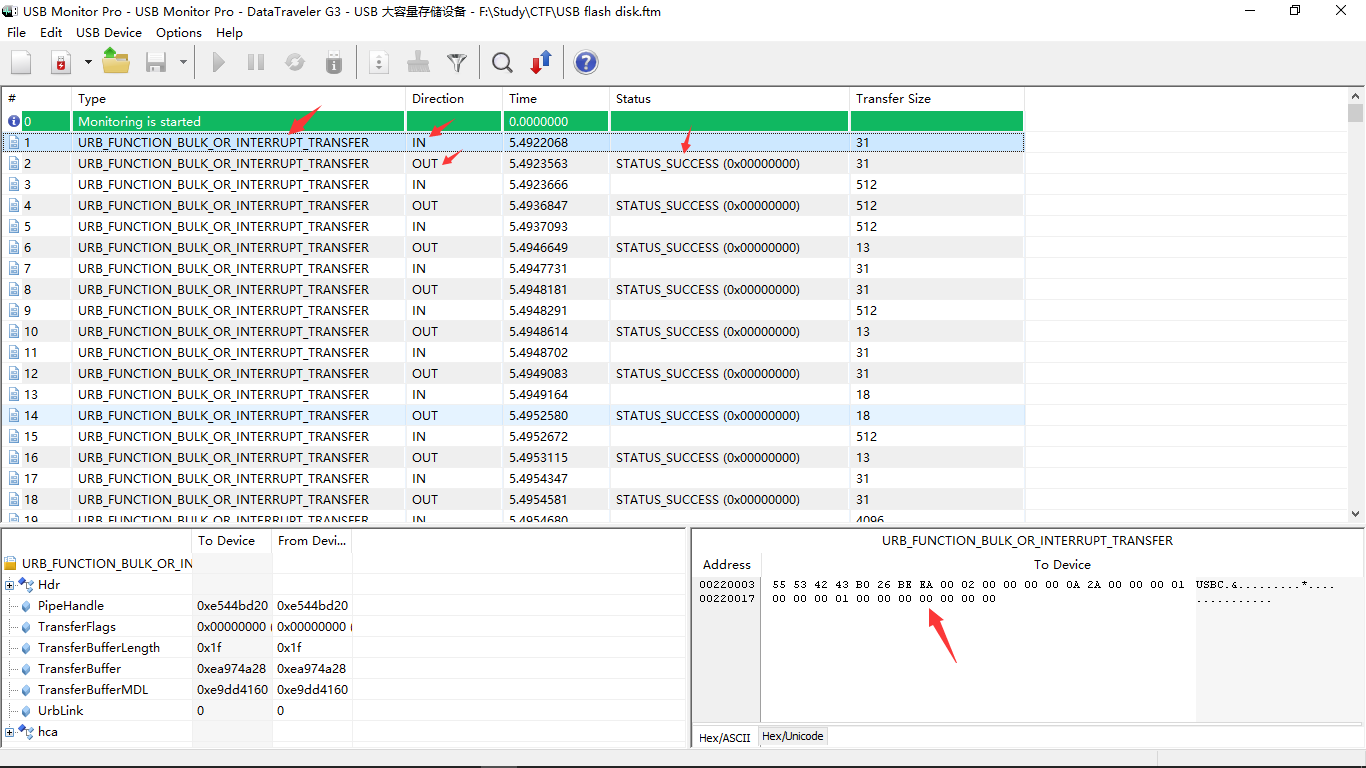
得到题目Do you know USB.zip解压得到USB flash disk.ftm和Teensy Keyboard.pcapng两个文件。



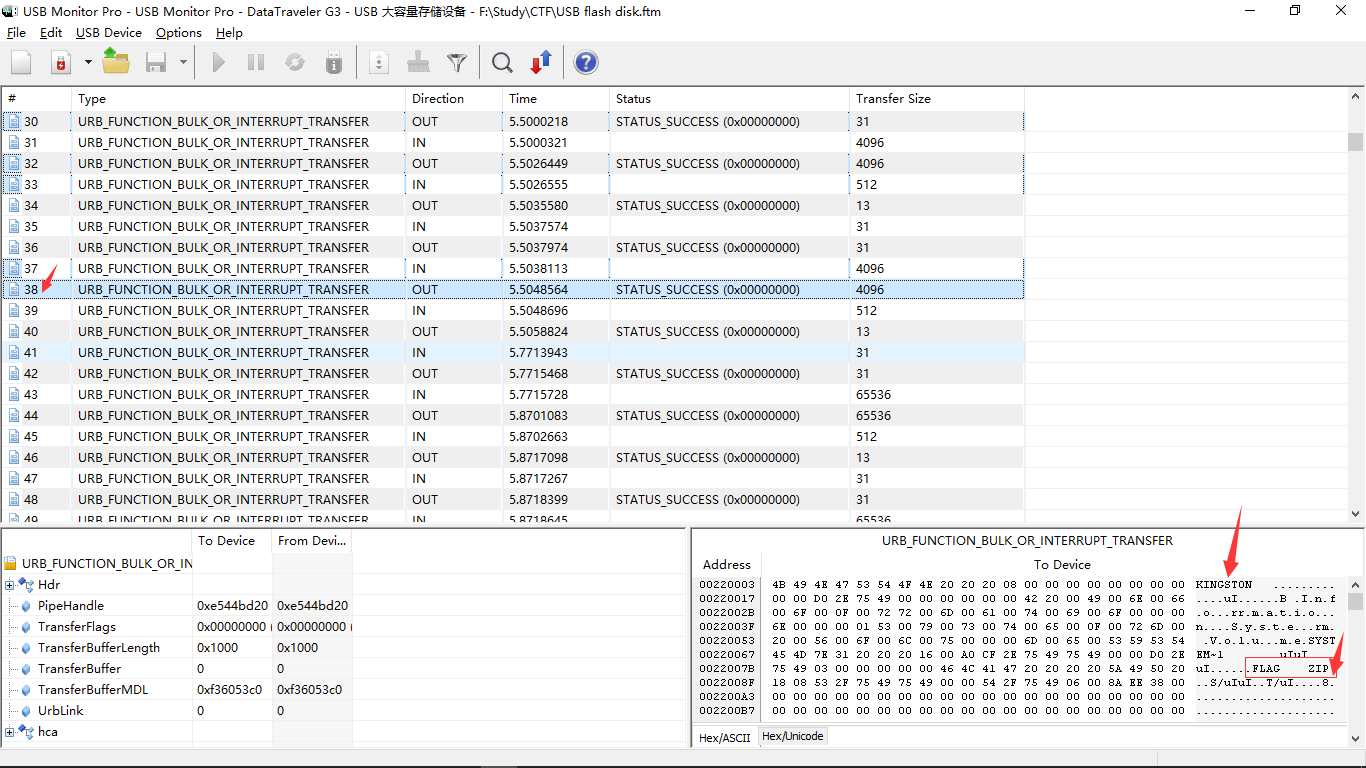
使用Hex Editor Neo分析USB flash disk.ftm，网上查询关于ftm格式和USB Monitor Pro资料发现，这是USB流量抓取和分析工具USB Monitor Pro抓取USB流量生成的文件。

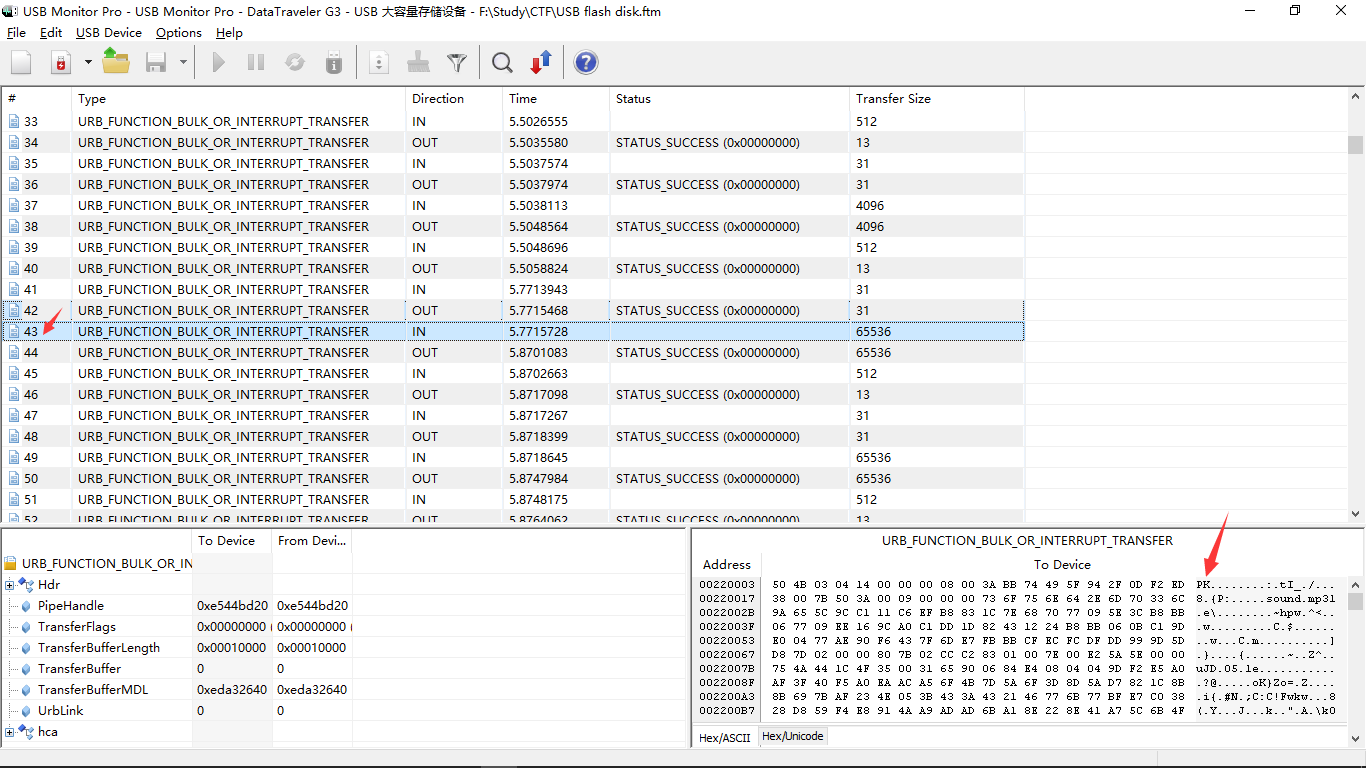


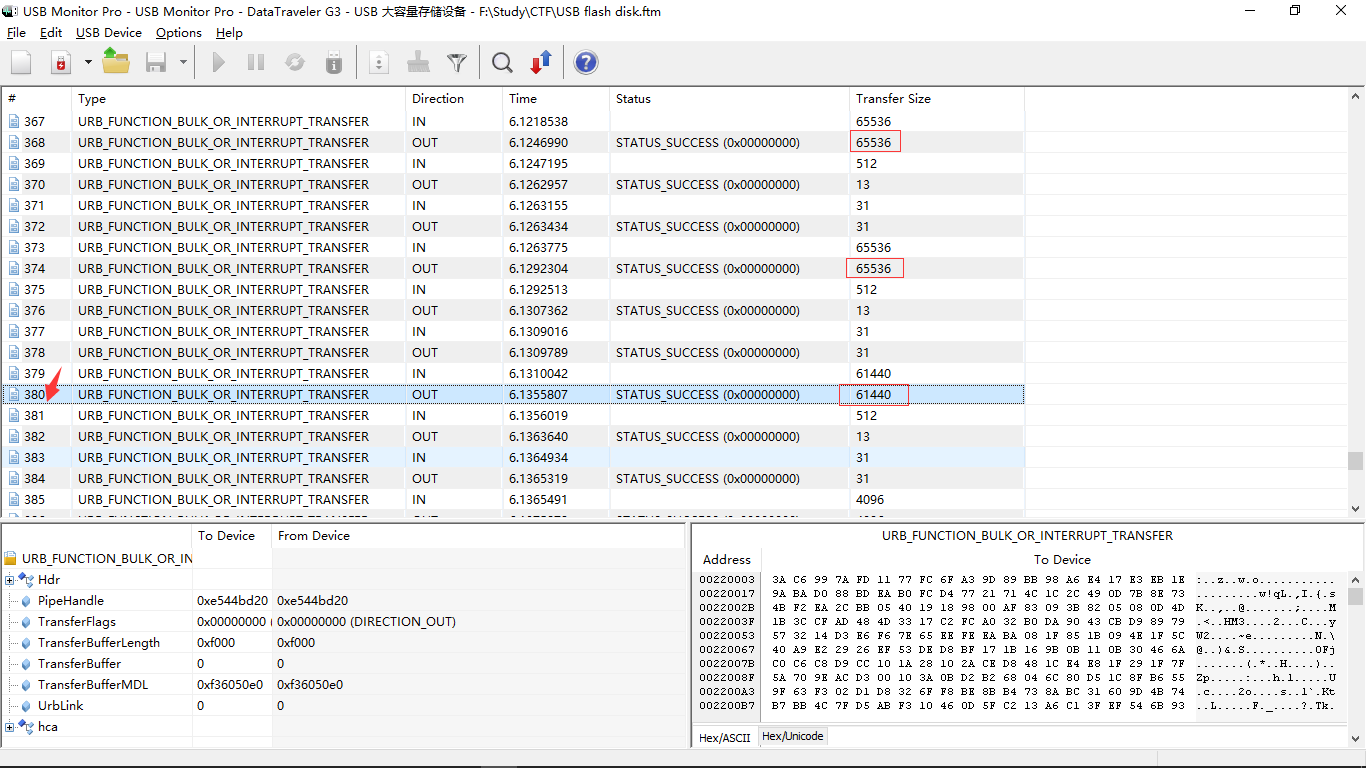
下载使用USB Monitor Pro，打开USB flash disk.ftm分析



网上查阅有关USB协议和URB（USB Request Block）资料，发现URB\_FUNCTION\_BULK\_OR\_INTERRUPT\_TRANSFER 这个URB功能函数就是USB接口的四种数据传输方式中中断传输和批量传输的合并。数据先从（IN）磁盘复制到缓存区，然后从缓存区提取（OUT）到U盘。上图右下角的便是传输的数据。

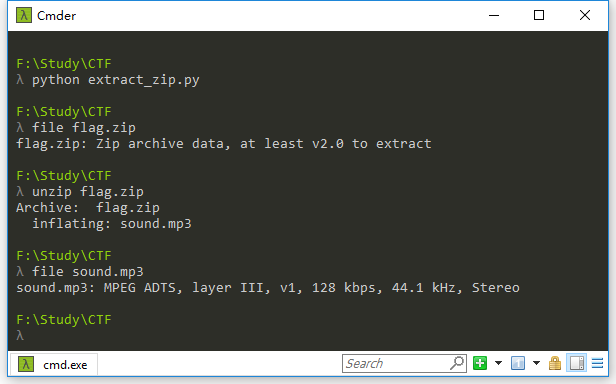






综合分析可知这是主机拷贝flag.zip到KINGSTON U盘的过程。结合包的大小规律分析可知，而传输flag.zip的数据从43号包到第380号结束。所以编写脚本提取出flag.zip：

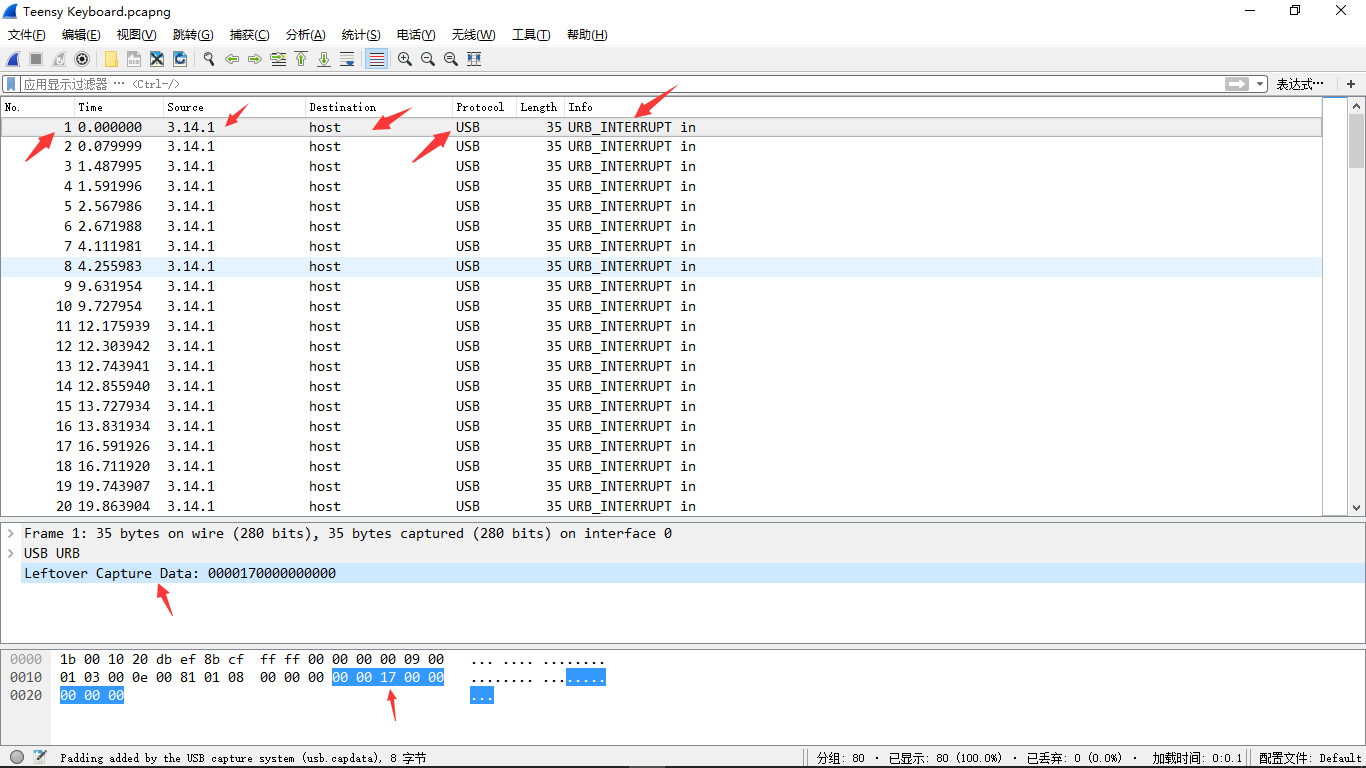
1. #-\*- coding:utf-8 -\*-
2. #!/usr/bin/python
3. # extract\_zip.py
4. ftm\_file = open('USB flash disk.ftm', 'rb')
5. zip\_file = open('flag.zip', 'wb')
6. read\_pointer = 46495
7. offset = 132715
8. zip\_data = ''
10. #循环读取56个大小为65526的包
11. **for** x **in** xrange(1, 57):
12. ftm\_file.seek(read\_pointer)
13. zip\_data += ftm\_file.read(65536)
14. read\_pointer += offset
16. #读取最后一个大小为61440的包
17. last\_read\_pointer = 7478535
18. ftm\_file.seek(last\_read\_pointer)
19. zip\_data += ftm\_file.read(61440)
21. zip\_data = zip\_data[:-374]#除去尾部填充的无用字节
23. zip\_file.write(zip\_data)



得到一个sound.mp3文件，推测flag可能是通过mp3stego加密隐藏在sound.mp3，而还有一个Teensy Keyboard.pcapng可能会有一些关于密码的提示，所以着手分析Teensy Keyboard.pcapng。

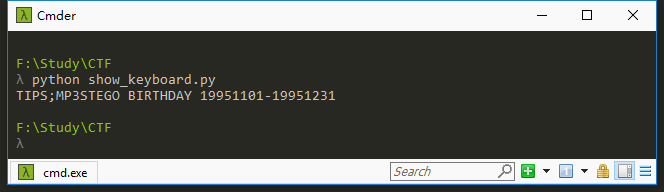
2、得到有关密码的提示

用Wireshark打开Teensy Keyboard.pcapng分析：



根据文件名Teensy Keyboard的提示，在网上查阅关于Teensy Keyboard键盘输入映射和USB协议结合分析得出这是一个编号3.23.1的接口（Teensy Keyboard 键盘）向主机host输入的过程。URB\_INTERRUPT是USB接口的数据传输方式中的中断传输，而在USB协议中usb.capdata（在Wireshark的描述符：Leftover Capture Data）负责储存通过USB的传输数据。其中8个字节大小的Leftover Capture Data中的第三个字节为可显字符，于是将考虑使用python的scapy模块编写脚本自动提取，先将pcapng转换为scapy能解析的pcap格式：

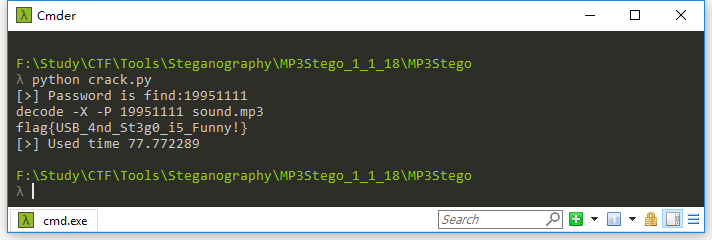
1. root**@kali**:~/Desktop# editcap -F libpcap -T ether "Teensy Keyboard.pcapng" "Teensy Keyboard.pcap"
2. #!/usr/bin/python
3. #show\_keyboard.py
4. **from** scapy.all **import** \*
6. VISIBLE\_KEY\_CODES = {
7. 4: "A",
8. 5: "B",
9. 6: "C",
10. 7: "D",
11. 8: "E",
12. 9: "F",
13. 10: "G",
14. 11: "H",
15. 12: "I",
16. 13: "J",
17. 14: "K",
18. 15: "L",
19. 16: "M",
20. 17: "N",
21. 18: "O",
22. 19: "P",
23. 20: "Q",
24. 21: "R",
25. 22: "S",
26. 23: "T",
27. 24: "U",
28. 25: "V",
29. 26: "W",
30. 27: "X",
31. 28: "Y",
32. 29: "Z",
33. 30: "1",
34. 31: "2",
35. 32: "3",
36. 33: "4",
37. 34: "5",
38. 35: "6",
39. 36: "7",
40. 37: "8",
41. 38: "9",
42. 39: "0",
43. 40: "\n",
44. 44: " ",
45. 45: "-",
46. 46: "=",
47. 47: "{",
48. 48: "}",
49. 49: "\\",
50. 51: ";",
51. 52: "'",
52. 53: "~",
53. 54: ",",
54. 55: ".",
55. 56: "/",
56. }
58. pkts = rdpcap("Teensy Keyboard.pcap")
59. msg = ""
60. **for** packet **in** pkts:
61. keybord\_data = packet.load[-8:]
62. key\_code = ord(keybord\_data[2])
63. ch = VISIBLE\_KEY\_CODES.get(key\_code, False)
64. **if** ch:
65. msg += ch
66. **print** msg



3、编写脚本穷举自动穷举密码提取flag

提示是mp3stego，而密码提示是生日，范围在19951101到19951231于是编写脚本穷举密码获取flag。

1. #coding=utf-8
2. #!/usr/bin/python
3. #crack.py
5. **import** os
6. **import** subprocess
7. **import** time
9. **def** password\_crack(password):
10. command='decode -X -P %s sound.mp3 '% password
11. # print command
12. p = subprocess.Popen(command, stdin = subprocess.PIPE,
13. stdout = subprocess.PIPE, stderr = subprocess.PIPE, shell = True)
14. **if** "unexpected end of cipher message."**not** **in** p.communicate()[1]:
15. **print** '[>] Password is find:%s' %password
16. **print** command
17. flag = open('sound.mp3.txt')
18. **print** flag.read()
19. **return** True
21. **def** generate\_birthday():
22. start = time.clock()
23. year = '1995'
24. **for** month **in** xrange(11,13):
25. month = str(month)
26. **for** day **in** xrange(1,32):
27. day = str(day).zfill(2)
28. birthday = year + month + day
29. password\_find = password\_crack(birthday)
30. **if** password\_find **is** True:
31. end = time.clock()
32. **print** '[>] Used time %f' %(end - start)
33. exit()
35. **print** '[>] Used time %f' %(end - start)
37. **def** main():
38. generate\_birthday()
40. **if** \_\_name\_\_ == '\_\_main\_\_':
41. main()



最终得到flag：flag{USB\_4nd\_St3g0\_i5\_Funny!}