【技术分享】从CTF中学USB流量捕获与解析

发布时间：2016-12-28 13:29:24

作者：Elph

预估稿费：500RMB（不服你也来投稿啊！）

投稿方式：发送邮件至linwei#360.cn，或登陆网页版在线投稿

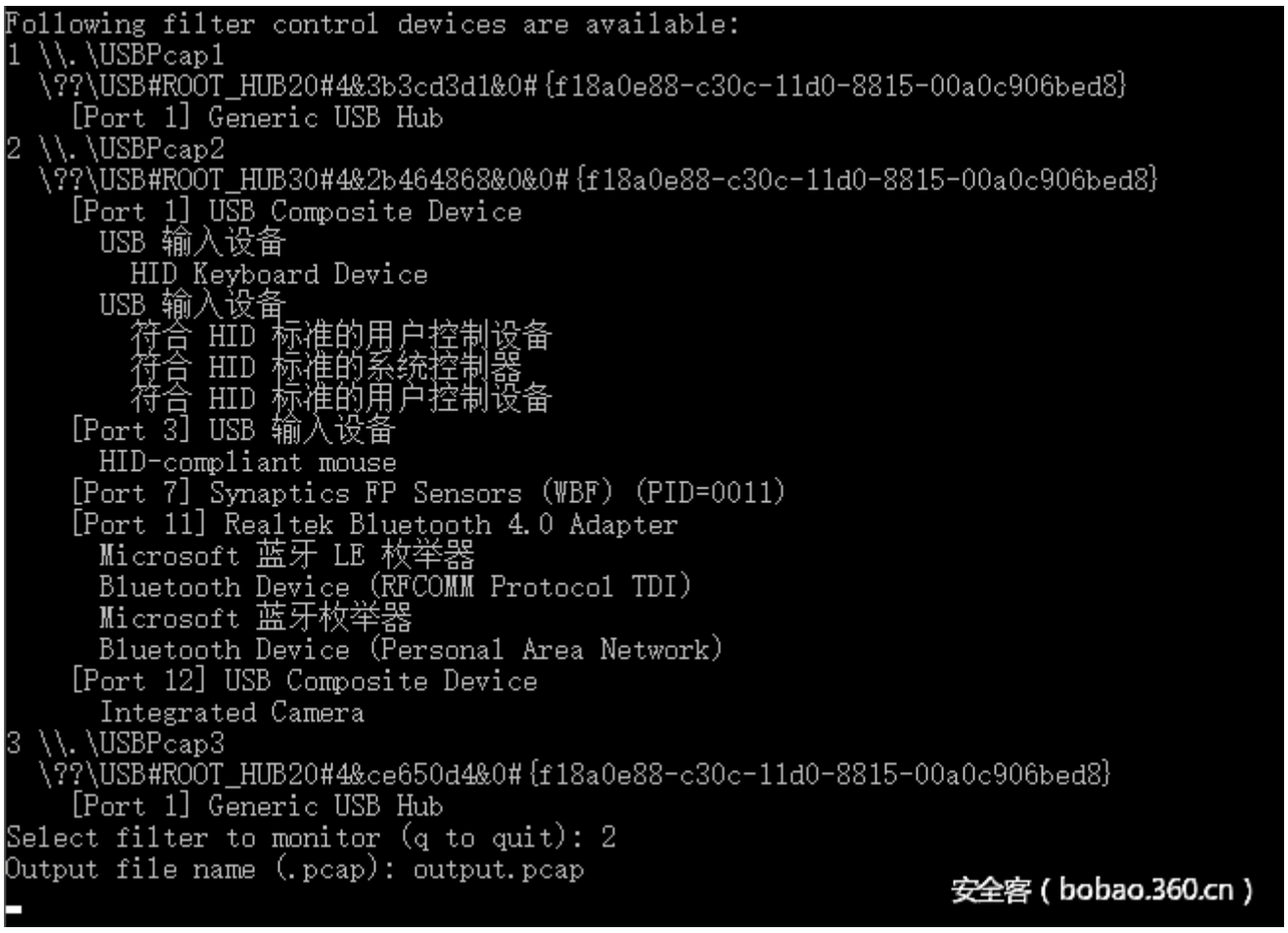
0x00 简介

USB接口是目前最为通用的外设接口之一，通过监听该接口的流量，可以得到很多有意思的东西，例如键盘击键，鼠标移动与点击，存储设备的明文传输通信、USB无线网卡网络传输内容等。本文将通过两个CTF题，讲解如何捕获USB接口的数据，以及键盘鼠标USB协议的具体解析方式。相关下载链接：http://pan.baidu.com/s/1i57b33B

0x01 USB capture

USB流量的捕获可以使用wireshark或者usbpcap来进行，最新版本的wireshark已经支持USB接口的捕获，且在安装时会提示usbpcap的安装，当前网上已有相关中文资料对wireshark抓取usb数据包的方法进行讲解，感兴趣的读者可阅读参考链接，在此我们使用一种相对简单的方式，即直接采用usbpcap捕获USB流量。该软件的下载地址为http://desowin.org/usbpcap/，可支持32位以及64位的winxp至win10操作系统，安装完成后须重启机器或者按照提示选择重启所有USB设备。

安装完成后，直接双击USBPcapCMD.exe，按照提示信息选择filter，输入文件名，便可愉快地等产生的信息被捕获了，程序运行界面如下图：



0x02 键盘流量解析

以今年xnuca misc 专场的old years 题为例，该题目给出一个pcap包，使用wireshark打开，看到Protocol 为USB协议。

USB协议的数据部分在Leftover Capture Data域之中，右键leftover capture data –> 应用为列，可以将该域的值在主面板上显示，键盘数据包的数据长度为8个字节，击键信息集中在第3个字节，每次key stroke都会产生一个keyboard event usb packet ，如下图：

网上查找USB协议的文档，可以找到这个值与具体键位的对应关系，http://www.usb.org/developers/hidpage/Hut1_12v2.pdf

第53页有一个usb keyboard/keypad映射表：

使用wireshark自带的tshark命令行工具，可以将 leftover capture data单独提取出来，具体命令为：

```
tshark.exe -r usb1.pcap -T fields -e usb.capdata > usbdata.txt
```

然后我们需要编写脚本从得出的userdata.txt文件中过滤出键盘击键相关的流量，并根据上述映射表，将键盘按键按照对应关系输出出来，这里附上简要的脚本：

```
mappings = { 0x04:"A",  0x05:"B",  0x06:"C", 0x07:"D", 0x08:"E", 0x09:"F", 0x0A:"G",  0x0B:"H", 0x0C:"I",  0x0D:"J", 0x0
[DEL]",  0X2B:"    ", 0x2C:" ",  0x2D:"-", 0x2E:"=", 0x2F:"
[",  0x30:"]",  0x31:"\", 0x32:"~", 0x33:";",  0x34:"'", 0x36:",",  0x37:"." }
nums = []
keys = open('usbdata.txt')
for line in keys:
    if line[0]!='0' or line[1]!='0' or line[3]!='0' or line[4]!='0' or line[9]!='0' or line[10]!='0' or line[12]!='0' or


         continue
    nums.append(int(line[6:8],16))
keys.close()
output = ""
for n in nums:
    if n == 0 :
        continue
    if n in mappings:
        output += mappings[n]
    else:
        output += '[unknown]'
print 'output :n' + output
```

运行该脚本，便可得到输出结果如下：

这里还有最后一个小弯，由提示中的"不使用拼音 "等信息推测出上文的特殊编码可能是某种拼音之外的古老的输入法，例如五笔，毕竟这也是做keylogger的人需要考虑而且头疼的一个地方。尝试对照着输入，可以得出如下文字：

最后得出flag . xnuca{wojiushifulagehaha}

0x03 鼠标流量解析

这是xnuca第二道usb流量的题，首先可以直接使用上述的解决方案试一下，得到这样一话：

解决本题需要把鼠标流量还原出来，然而鼠标与键盘不同，鼠标移动时表现为连续性，与键盘击键的离散性不一样，不过实际上鼠标动作所产生的数据包也是离散的，毕竟计算机表现的连续性信息都是由大量离散信息构成的。

首先同样使用tshark 命令将cap data提取出来:

```
tshark.exe -r usb2.pcap -T fields -e usb.capdata > usbdata.txt
```

每一个数据包的数据区有四个字节，第一个字节代表按键，当取0x00时，代表没有按键、为0x01时，代表按左键，为0x02时，代表当前按键为右键。第二个字节可以看成是一个signed byte类型，其最高位为符号位，当这个值为正时，代表鼠标水平右移多少像素，为负时，代表水平左移多少像素。第三个字节与第二字节类似，代表垂直上下移动的偏移。

了解协议相关约定之后，可编写脚本将数据包的内容变成一系列点的集合，为了区分左右按键，可以特意对第一个字节的内容作一下判断。相关脚本如下：

```
nums = []
keys = open('data.txt','r')
posx = 0
posy = 0
for line in keys:
    if len(line) != 12 :
         continue
    x = int(line[3:5],16)
    y = int(line[6:8],16)
    if x > 127 :
        x -= 256
    if y > 127 :
        y -= 256
    posx += x
    posy += y
    btn_flag = int(line[0:2],16)  # 1 for left , 2 for right , 0 for nothing
    if btn_flag == 1 :
        print posx , posy
keys.close()
```

本题的flag藏在右键信息中，当btn_flag 取2时，运行脚本可以得到一系列坐标点：

得到这些点之后，需要将他们画出来，因而需要辅以gnuplot 或者其他的绘图工具，gnuplot的命令为"plot inputfile"，运行如下：

最后得到本题的flag： XNUCA{USBPCAPGETEVERYTHING}

安全知识

Elph

推荐阅读

以原生速度在 macOS 上模糊测试 iOS 代码

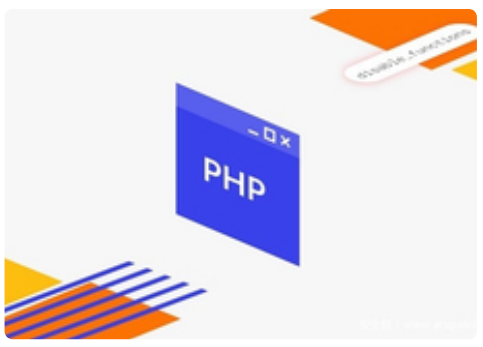
2021-10-09 16:30:29

羊城杯逆向Writeup

2021-10-09 15:30:36

绿城杯2021 By T3ns0r

2021-10-09 14:30:00



深入 FTP 攻击 php-fpm 绕过 disable_functions

2021-10-09 10:29:07