HTB x Uni CTF 2020 – Qualifications

Solver: Will Green (UAHDucky)

Challenge: Plug

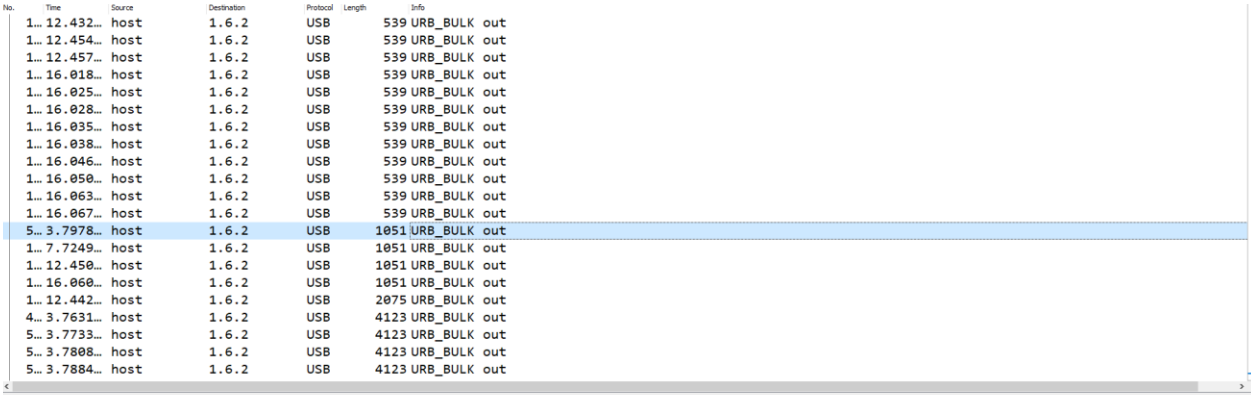
Category: Forensics

Intro:

In this challenge we are given a .pcapng file which contains USB traffic

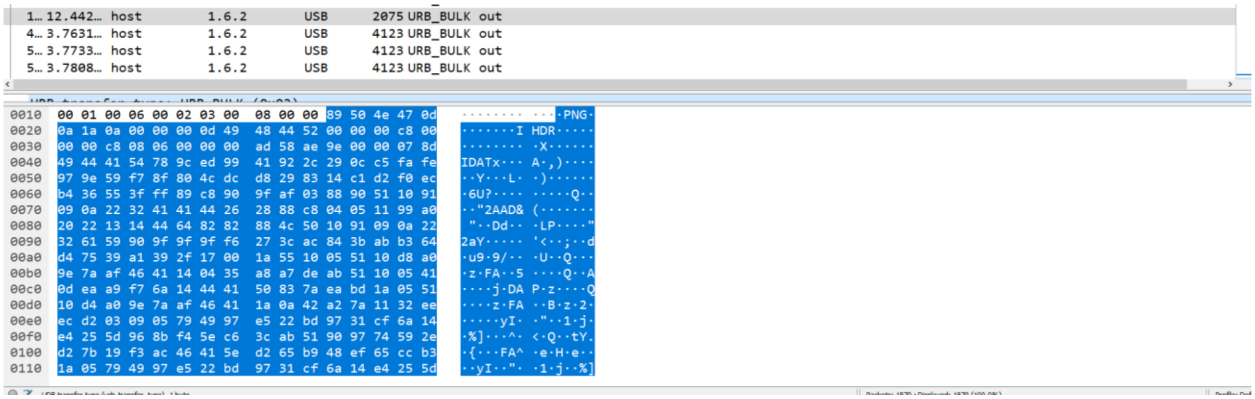
Walkthrough:

I first started analyzing the pcap file just by scrolling through each packet and seeing if I noticed any weird patterns. I noticed that probably 97% of the packets were less than 100 bytes in length and there were a few packets that were way larger, about 4,000 bytes, than the rest. I assumed these large packets was where actual data that we were concerned about was located, so I sorted by length:



No.	Time	Source	Destination	Protocol	Length	Info
1…	12.432…	host	1.6.2	USB	539	URB_BULK out
1…	12.454…	host	1.6.2	USB	539	URB_BULK out
1…	12.457…	host	1.6.2	USB	539	URB_BULK out
1…	16.018…	host	1.6.2	USB	539	URB_BULK out
1…	16.025…	host	1.6.2	USB	539	URB_BULK out
1…	16.028…	host	1.6.2	USB	539	URB_BULK out
1…	16.035…	host	1.6.2	USB	539	URB_BULK out
1…	16.038…	host	1.6.2	USB	539	URB_BULK out
1…	16.046…	host	1.6.2	USB	539	URB_BULK out
1…	16.050…	host	1.6.2	USB	539	URB_BULK out
1…	16.063…	host	1.6.2	USB	539	URB_BULK out
1…	16.067…	host	1.6.2	USB	539	URB_BULK out
5…	3.7978…	host	1.6.2	USB	1051	URB_BULK out
1…	7.7249…	host	1.6.2	USB	1051	URB_BULK out
1…	12.450…	host	1.6.2	USB	1051	URB_BULK out
1…	16.060…	host	1.6.2	USB	1051	URB_BULK out
1…	12.442…	host	1.6.2	USB	2075	URB_BULK out
4…	3.7631…	host	1.6.2	USB	4123	URB_BULK out
5…	3.7733…	host	1.6.2	USB	4123	URB_BULK out
5…	3.7808…	host	1.6.2	USB	4123	URB_BULK out
5…	3.7884…	host	1.6.2	USB	4123	URB_BULK out

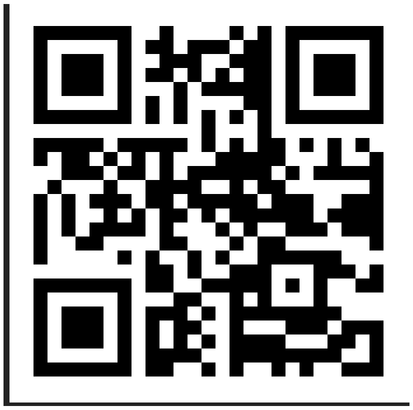
It didn’t take me long till a found a particular packet which looked like a header for a PNG file:



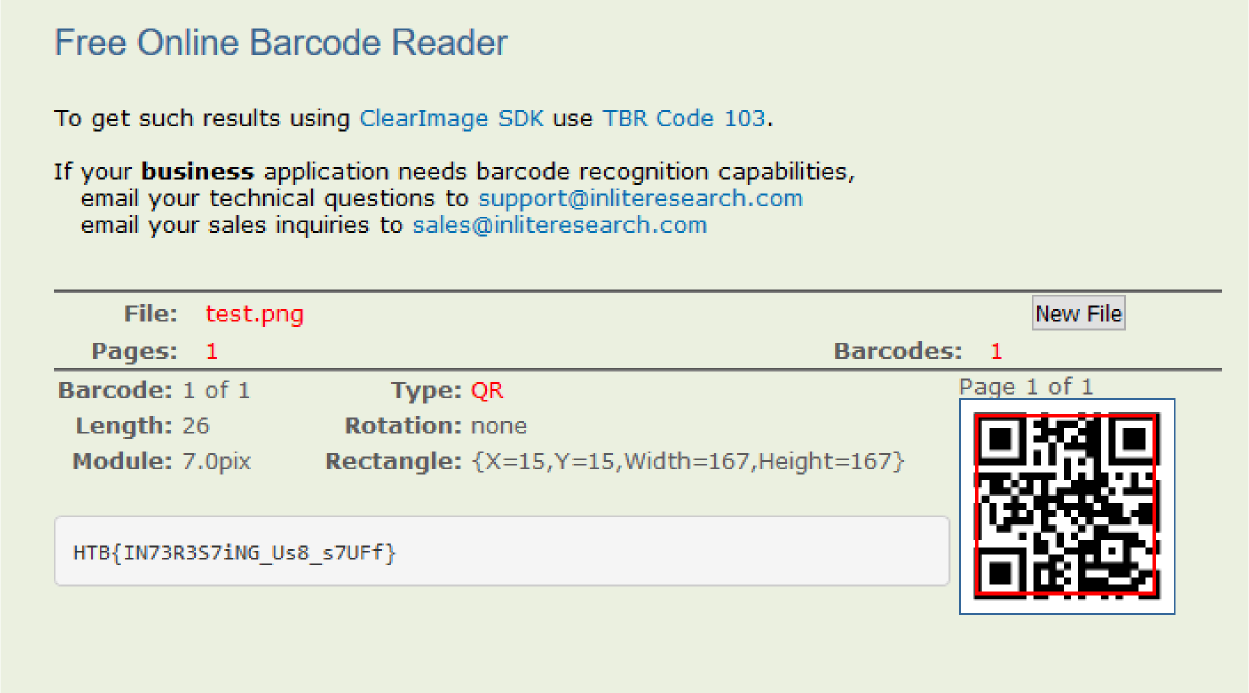
```
1… 12.442… host    1.6.2    USB    2075 URB_BULK out
4… 3.7631… host    1.6.2    USB    4123 URB_BULK out
5… 3.7733… host    1.6.2    USB    4123 URB_BULK out
5… 3.7808… host    1.6.2    USB    4123 URB_BULK out

0010  00 01 00 06 00 02 03 00  08 00 00 89 50 4e 47 0d   ········ ···PNG·
0020  0a 1a 0a 00 00 00 0d 49  48 44 52 00 00 00 c8 00   ·······I HDR·····
0030  00 00 c8 08 06 00 00 00  ad 58 ae 9e 00 00 07 8d   ········ ·X······
0040  49 44 41 54 78 9c ed 99  41 92 2c 29 0c c5 fa fe   IDATx··· A·,)····
0050  97 9e 59 f7 8f 80 4c dc  d8 29 83 14 c1 d2 f0 ec   ··Y···L· ·)······
0060  b4 36 55 3f ff 89 c8 90  9f af 03 88 90 51 10 91   ·6U?···· ·····Q··
0070  09 0a 22 32 41 41 44 26  28 88 c8 04 05 11 99 a0   ··"2AAD& (·······
0080  20 22 13 14 44 64 82 82  88 4c 50 10 91 09 0a 22    "··Dd·· ·LP····"
0090  32 61 59 90 9f 9f 9f f6  27 3c ac 84 3b ab b3 64   2aY····· '<··;··d
00a0  d4 75 39 a1 39 2f 17 00  1a 55 10 05 51 10 d8 a0   ·u9·9/·· ·U··Q···
00b0  9e 7a af 46 41 14 04 35  a8 a7 de ab 51 10 05 41   ·z·FA··5 ····Q··A
00c0  0d ea a9 f7 6a 14 44 41  50 83 7a ea bd 1a 05 51   ····j·DA P·z····Q
00d0  10 d4 a0 9e 7a af 46 41  1a 0a 42 a2 7a 11 32 ee   ····z·FA ··B·z·2·
00e0  ec d2 03 09 05 79 49 97  e5 22 bd 97 31 cf 6a 14   ·····yI· ·"··1·j·
00f0  e4 25 5d 96 8b f4 5e c6  3c ab 51 90 97 74 59 2e   ·%]···^· <·Q··tY.
0100  d2 7b 19 f3 ac 46 41 5e  d2 65 b9 48 ef 65 cc b3   ·{···FA^ ·e·H·e··
0110  1a 05 79 49 97 e5 22 bd  97 31 cf 6a 14 e4 25 5d   ··yI··"· ·1·j··%]
```

So, I selected the packet and saved the raw packet bytes as a PNG file. Opening up the PNG file showed me a QR code:



So I went to an online QR code reader website and had it read the QR code:



Free Online Barcode Reader

To get such results using ClearImage SDK use TBR Code 103.

If your **business** application needs barcode recognition capabilities,
email your technical questions to support@inliteresearch.com
email your sales inquiries to sales@inliteresearch.com

File: test.png

Pages: 1 **Barcodes:** 1

Barcode: 1 of 1 **Type:** QR
Length: 26 **Rotation:** none
Module: 7.0pix **Rectangle:** {X=15,Y=15,Width=167,Height=167}

```
HTB{IN73R3S7iNG_Us8_s7UFf}
```