duksctf

Bunch of security enthusiasts who sometimes play CTF

Blog About

AlexCTF 2017 - Unknown Format

We received a USB PCAP of an update transaction between a computer and Amazon Kindle. After reconstructing the update, we used a tool called KindleTool in order to deobfuscate the binary, then we used some python code to inflate the malformed gzip inside.

Description

Once more our agents managed to sniff data passed over USB, they told us that this is high profile data hidden by people knows what they are doing, they have dedicated devices for reading that secret file format. Can you help us finding what is the secret message?

Details

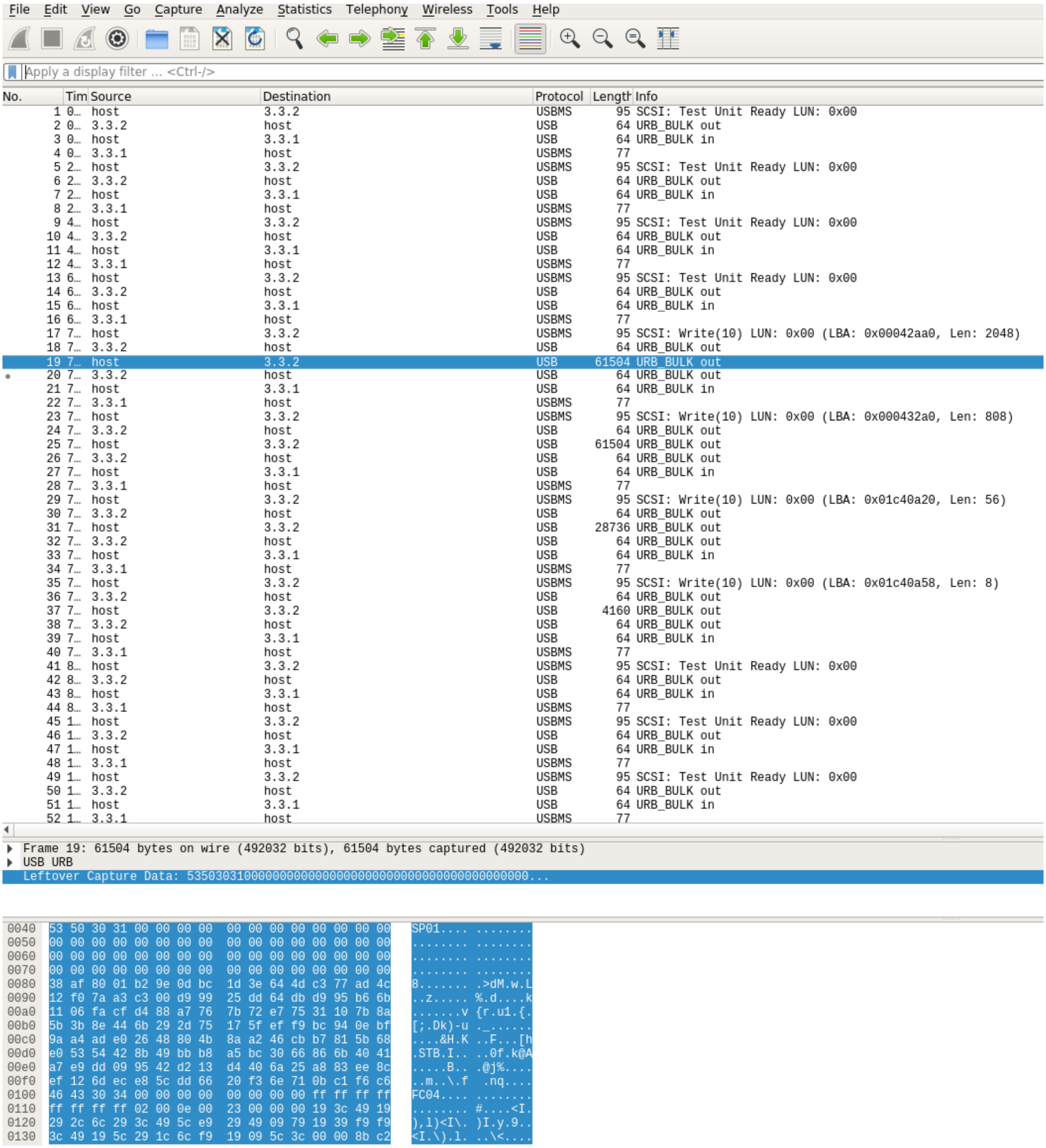
Points: 200

Category: forensic

Validations: 54

Solution

We were given a file called usb_sniff.pcap. After digging around the file for a while it appears that it's a USB transfer of several files. We looked on google first bytes of the transfer "SP01 and FC04" which led us to a github account KindleTool from NiLuJe. This tool help of reversing of image for many Kindle format.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.    Tim Source      Destination   Protocol Length Info
     1 0…  host        3.3.2         USBMS       95 SCSI: Test Unit Ready LUN: 0x00
     2 0…  3.3.2       host          USB         64 URB_BULK out
     3 0…  host        3.3.1         USB         64 URB_BULK in
     4 0…  3.3.1       host          USBMS       77
     5 2…  host        3.3.2         USBMS       95 SCSI: Test Unit Ready LUN: 0x00
     6 2…  3.3.2       host          USB         64 URB_BULK out
     7 2…  host        3.3.1         USB         64 URB_BULK in
     8 2…  3.3.1       host          USBMS       77
     9 4…  host        3.3.2         USBMS       95 SCSI: Test Unit Ready LUN: 0x00
    10 4…  3.3.2       host          USB         64 URB_BULK out
    11 4…  host        3.3.1         USB         64 URB_BULK in
    12 4…  3.3.1       host          USBMS       77
    13 6…  host        3.3.2         USBMS       95 SCSI: Test Unit Ready LUN: 0x00
    14 6…  3.3.2       host          USB         64 URB_BULK out
    15 6…  host        3.3.1         USB         64 URB_BULK in
    16 6…  3.3.1       host          USBMS       77
    17 7…  host        3.3.2         USBMS       95 SCSI: Write(10) LUN: 0x00 (LBA: 0x00042aa0, Len: 2048)
    18 7…  3.3.2       host          USB         64 URB_BULK out
    19 7…  host        3.3.2         USB      61504 URB_BULK out
    20 7…  3.3.2       host          USB         64 URB_BULK out
    21 7…  host        3.3.1         USB         64 URB_BULK in
    22 7…  3.3.1       host          USBMS       77
    23 7…  host        3.3.2         USBMS       95 SCSI: Write(10) LUN: 0x00 (LBA: 0x000432a0, Len: 808)
    24 7…  3.3.2       host          USB         64 URB_BULK out
    25 7…  host        3.3.2         USB      61504 URB_BULK out
    26 7…  3.3.2       host          USB         64 URB_BULK out
    27 7…  host        3.3.1         USB         64 URB_BULK in
    28 7…  3.3.1       host          USBMS       77
    29 7…  host        3.3.2         USBMS       95 SCSI: Write(10) LUN: 0x00 (LBA: 0x01c40a20, Len: 56)
    30 7…  3.3.2       host          USB         64 URB_BULK out
    31 7…  host        3.3.2         USB      28736 URB_BULK out
    32 7…  3.3.2       host          USB         64 URB_BULK out
    33 7…  host        3.3.1         USB         64 URB_BULK in
    34 7…  3.3.1       host          USBMS       77
    35 7…  host        3.3.2         USBMS       95 SCSI: Write(10) LUN: 0x00 (LBA: 0x01c40a58, Len: 8)
    36 7…  3.3.2       host          USB         64 URB_BULK out
    37 7…  host        3.3.2         USB       4160 URB_BULK out
    38 7…  3.3.2       host          USB         64 URB_BULK out
    39 7…  host        3.3.1         USB         64 URB_BULK in
    40 7…  3.3.1       host          USBMS       77
    41 8…  host        3.3.2         USBMS       95 SCSI: Test Unit Ready LUN: 0x00
    42 8…  3.3.2       host          USB         64 URB_BULK out
    43 8…  host        3.3.1         USB         64 URB_BULK in
    44 8…  3.3.1       host          USBMS       77
    45 1…  host        3.3.2         USBMS       95 SCSI: Test Unit Ready LUN: 0x00
    46 1…  3.3.2       host          USB         64 URB_BULK out
    47 1…  host        3.3.1         USB         64 URB_BULK in
    48 1…  3.3.1       host          USBMS       77
    49 1…  host        3.3.2         USBMS       95 SCSI: Test Unit Ready LUN: 0x00
    50 1…  3.3.2       host          USB         64 URB_BULK out
    51 1…  host        3.3.1         USB         64 URB_BULK in
    52 1…  3.3.1       host          USBMS       77
```

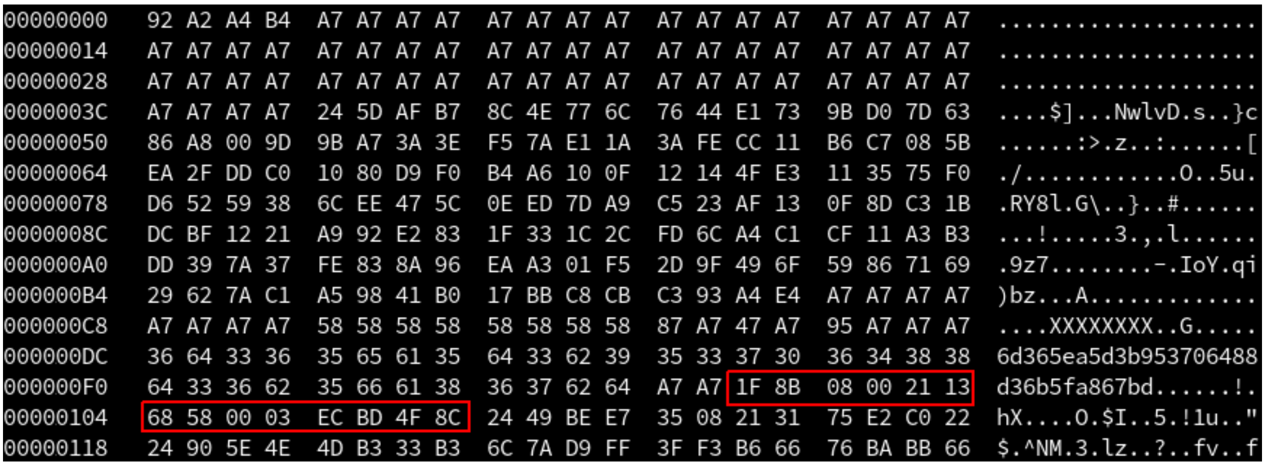
```
▸ Frame 19: 61504 bytes on wire (492032 bits), 61504 bytes captured (492032 bits)
▸ USB URB
  Leftover Capture Data: 535030310000000000000000000000000000000000000000...
```

```
0040  53 50 30 31 00 00 00 00  00 00 00 00 00 00 00 00   SP01.... ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  38 af 80 01 b2 9e 0d bc  1d 3e 64 4d c3 77 ad 4c   8....... .>dM.w.L
0090  12 f0 7a a3 c3 00 d9 99  25 dd 64 db d9 95 b6 6b   ..z..... %.d....k
00a0  11 06 fa cf d4 88 a7 76  7b 72 e7 75 31 10 7b 8a   .......v {r.u1.{.
00b0  5b 3b 8e 44 6b 29 2d 75  17 5f ef f9 bc 94 0e bf   [;.Dk)-u ._......
00c0  9a a4 ad e0 26 48 80 4b  8a a2 46 cb b7 81 5b 68   ....&H.K ..F...[h
00d0  e0 53 54 42 8b 49 bb b8  a5 bc 30 66 86 6b 40 41   .STB.I.. ..0f.k@A
00e0  a7 e9 dd 09 95 42 d2 13  d4 40 6a 25 a8 83 ee 8c   .....B.. .@j%....
00f0  ef 12 6d ec e8 5c dd 66  20 f3 6e 71 0b c1 f6 c6   ..m..\.f  .nq....
0100  46 43 30 34 00 00 00 00  00 00 00 00 ff ff ff ff   FC04.... ........
0110  ff ff ff ff 02 00 0e 00  23 00 00 00 19 3c 49 19   ........ #....<I.
0120  29 2c 6c 29 3c 49 5c e9  29 49 09 79 19 39 f9 f9   ),l)<I\. )I.y.9..
0130  3c 49 19 5c 29 1c 6c f9  19 09 5c 3c 00 00 8b c2   <I.\).l. ..\<....
```

We extracted first two packet of URB’s and then we reconstructed the image with cat:

```
cat packet1.bin packet.bin > packet.bin
```

After compiling the tool we tried some command to extract or convert the update without luck. We used the convert with *-w option* to unwrapp the signatures header and then the *dm* to deobfuscate the binary. After reading the code of KindleTool, we found that after the *FC04* and some bytes there is a gziped sections of data.



```
00000000   92 A2 A4 B4  A7 A7 A7 A7  A7 A7 A7 A7  A7 A7 A7 A7  A7 A7 A7 A7  ....................
00000014   A7 A7 A7 A7  A7 A7 A7 A7  A7 A7 A7 A7  A7 A7 A7 A7  A7 A7 A7 A7  ....................
00000028   A7 A7 A7 A7  A7 A7 A7 A7  A7 A7 A7 A7  A7 A7 A7 A7  A7 A7 A7 A7  ....................
0000003C   A7 A7 A7 A7  24 5D AF B7  8C 4E 77 6C  76 44 E1 73  9B D0 7D 63  ....$]...NwlvD.s..}c
00000050   86 A8 00 9D  9B A7 3A 3E  F5 7A E1 1A  3A FE CC 11  B6 C7 08 5B  ......:>.z..:......[
00000064   EA 2F DD C0  10 80 D9 F0  B4 A6 10 0F  12 14 4F E3  11 35 75 F0  ./............O..5u.
00000078   D6 52 59 38  6C EE 47 5C  0E ED 7D A9  C5 23 AF 13  0F 8D C3 1B  .RY8l.G\..}..#......
0000008C   DC BF 12 21  A9 92 E2 83  1F 33 1C 2C  FD 6C A4 C1  CF 11 A3 B3  ...!.....3.,.l......
000000A0   DD 39 7A 37  FE 83 8A 96  EA A3 01 F5  2D 9F 49 6F  59 86 71 69  .9z7........-.IoY.qi
000000B4   29 62 7A C1  A5 98 41 B0  17 BB C8 CB  C3 93 A4 E4  A7 A7 A7 A7  )bz...A.............
000000C8   A7 A7 A7 A7  58 58 58 58  58 58 58 58  87 A7 47 A7  95 A7 A7 A7  ....XXXXXXXX..G.....
000000DC   36 64 33 36  35 65 61 35  64 33 62 39  35 33 37 30  36 34 38 38  6d365ea5d3b953706488
000000F0   64 33 36 62  35 66 61 38  36 37 62 64  A7 A7 1F 8B  08 00 21 13  d36b5fa867bd......!.
00000104   68 58 00 03  EC BD 4F 8C  24 49 BE E7  35 08 21 31  75 E2 C0 22  hX....O.$I..5.!1u.."
00000118   24 90 5E 4E  4D B3 33 B3  6C 7A D9 FF  3F F3 B6 66  76 BA BB 66  $.^NM.3.lz..?..fv..f
```

We used *dd* to extract the broken gziped archive:

```
dd if=packet.bin of=broken.bin skip=1 bs=254
```

We tried to extract it with *tar xvf* whic didn’t works. We used then a simple python script from stackoverflow:

```
# http://stackoverflow.com/questions/2423866/python-dec
# http://stackoverflow.com/questions/3122145/zlib-error

def read_corrupted_file(filename, CHUNKSIZE=1024):
    d = zlib.decompressobj(zlib.MAX_WBITS | 32)
    with open(filename, 'rb') as f:
        result_str = ''
        buffer=f.read(CHUNKSIZE)
        try:
            while buffer:
                result_str += d.decompress(buffer)
                buffer=f.read(CHUNKSIZE)
        except Exception as e:
            print 'Error: %s -> %s' % (filename, e.mess
        return result_str
```

```
In [3]: read_corrupted_file("broken.bin")
REDACTED
0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
```

The flag was: **ALEXCTF{Wh0_N33d5_K1nDl3_t0_3X7R4Ct_K1ND13_F1rMw4R3}**

Challenges resources are available in the resources folder

Written on February 4, 2017