HackTM CTF 2020 Writeup

Since the CTF is still active I wont be dropping the flags. You can follow along and complete the challenges for yourself here: https://ctfx.hacktm.ro/

0x01 Strange PCAP

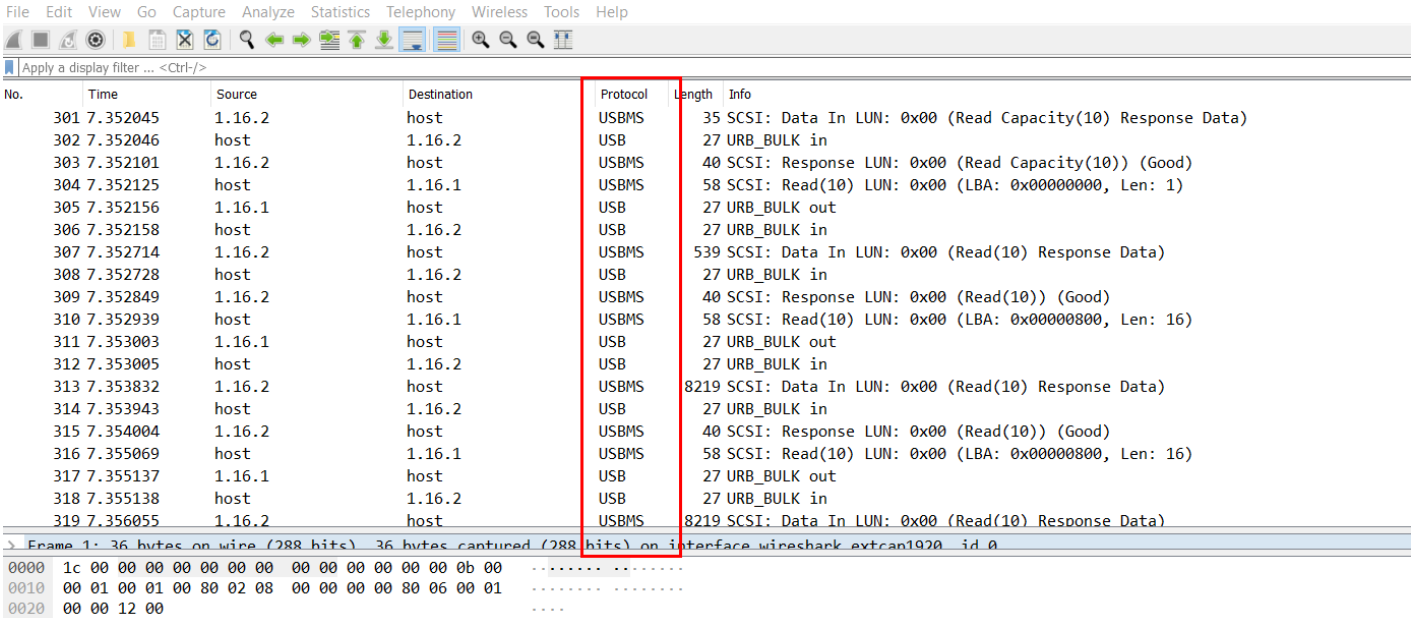
Strange PCAP 144 Points

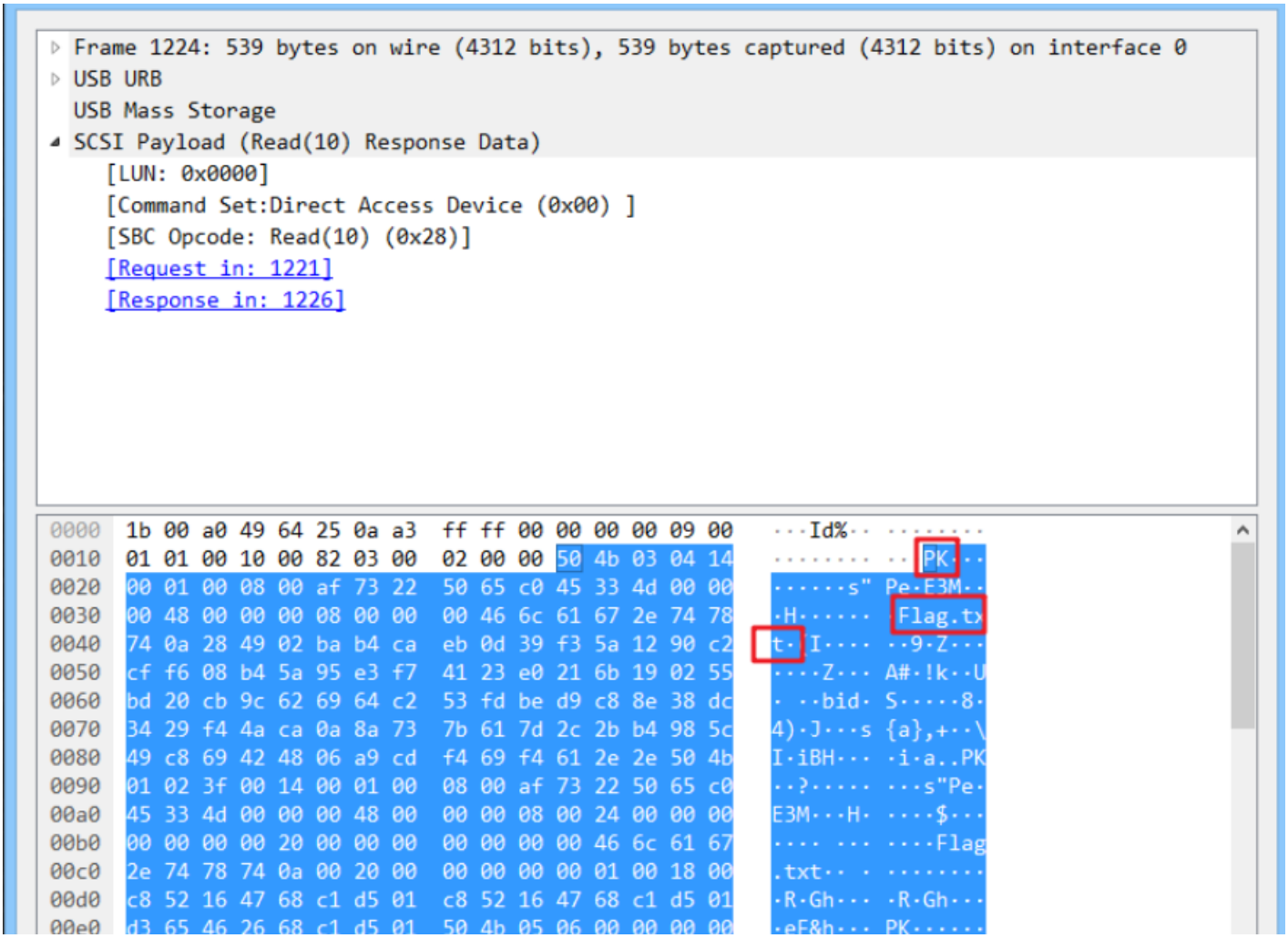
We managed to get all the data to incriminate our CEO for selling company secrets. Can you please help us and give us the secret data that he has leaked?

Author: Legacy

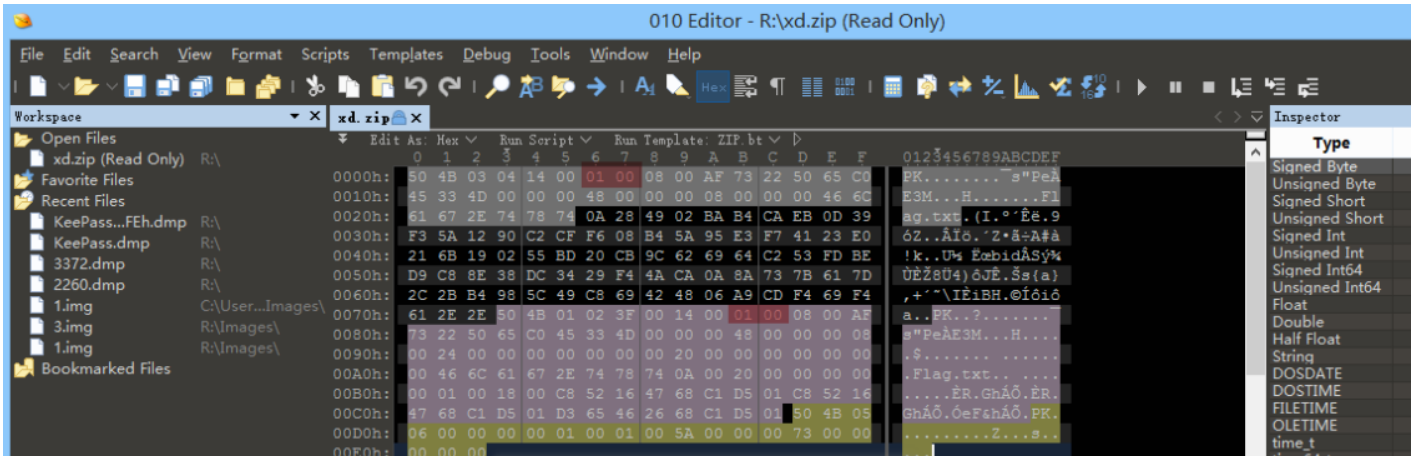
Strange.pcapng ccd21d48fd04137551833d2e4493243e

A basic PCAP forensics question. When you get the file, you will find that there are many USB Massive Storage packages. The initial guess is to find the flag in it.

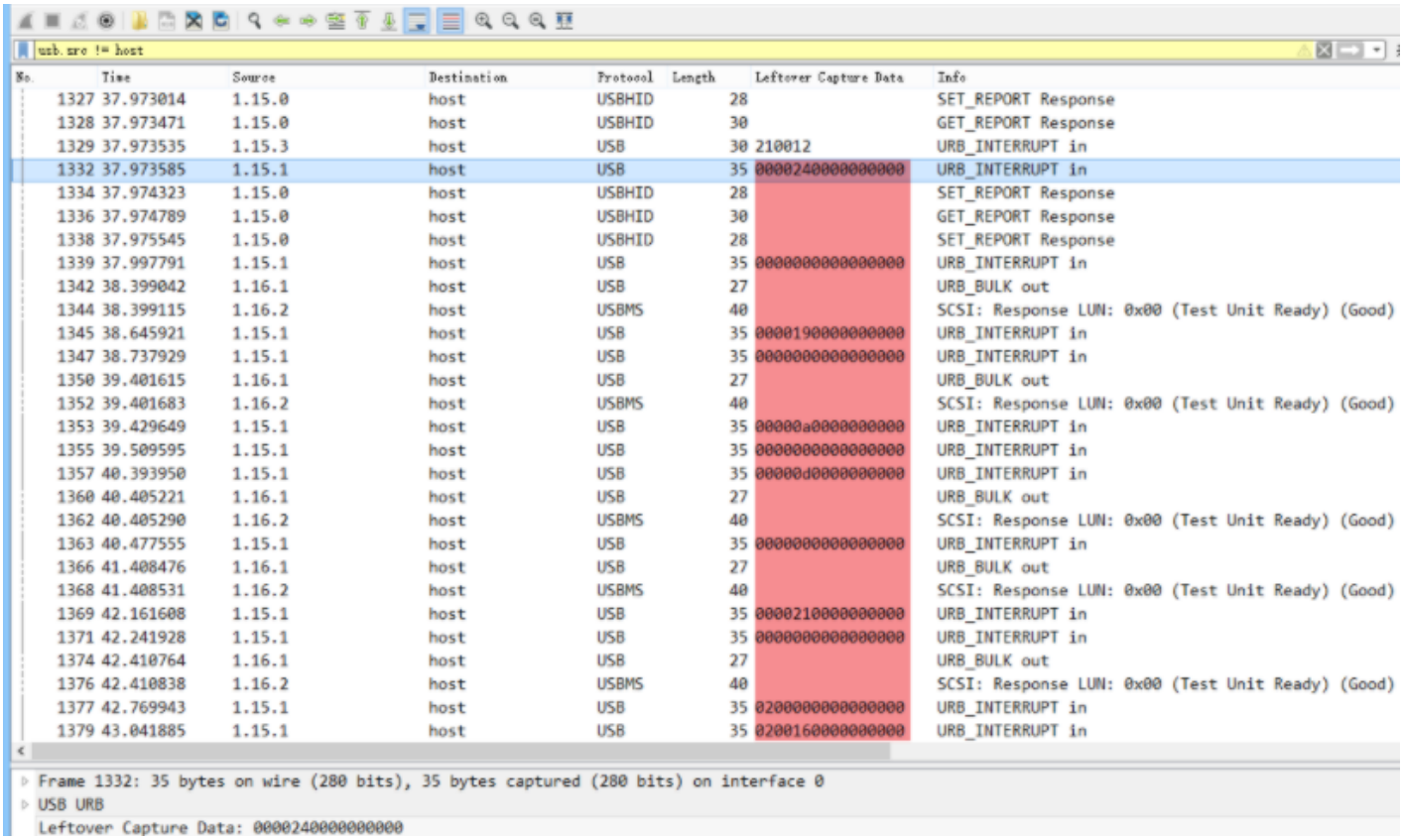




Looking through the data packets one by one, I found the ZIP file header and Flag.txt in Frame 1224, extract them and have a look.



After extracting the data, it is found to be an encrypted compressed package. Check it with 010 Editor and find that it is really encrypted. Next, we have to find the password. Let's make a preliminary guess about the HID package sent by the keyboard because we have seen the USB HID data package before Types of.



Scrolling down, I saw a lot of URB_INTERRUPT types. We exported these data packets with tshark. For the detailed process, please refer to the second level of the 2020 New Year Red Packet Writeup of Milk Ice . I will not repeat it here.

Refer to the USB Keyboard data packet format , you can know that the first Byte of each packet corresponds to the state of the control key, and the third Byte corresponds to the input key.

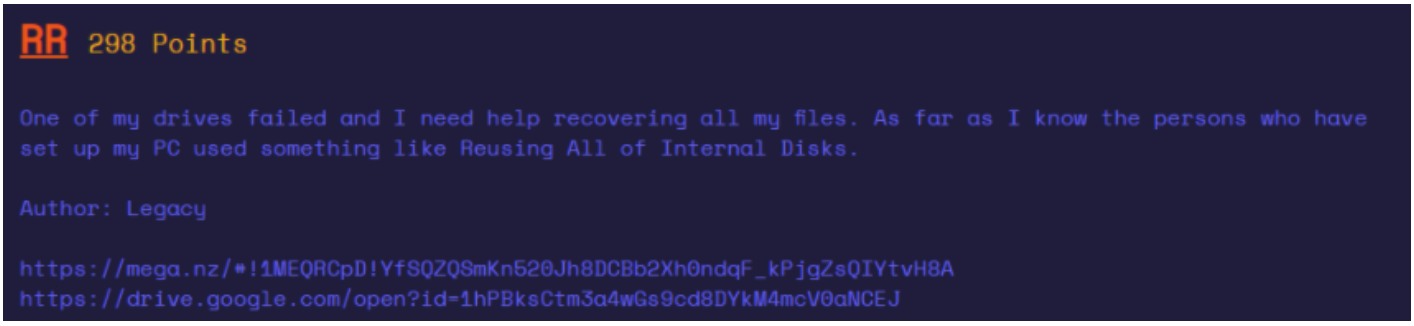
Combined with the USB HID Keyboard scan codes, the following script can be constructed to analyze the data packet.

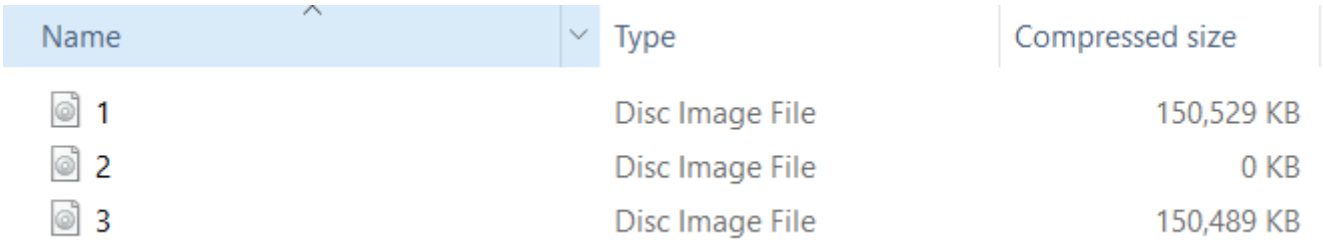
```
usb_codes = {
    0x04: "aA",  0x05: "bB",  0x06: "cC",  0x07: "dD",  0x08: "eE",  0x09: "fF",
    0x0A: "gG",  0x0B: "hH",  0x0C: "iI",  0x0D: "jJ",  0x0E: "kK",  0x0F: "lL",
    0x10: "mM",  0x11: "nN",  0x12: "oO",  0x13: "pP",  0x14: "qQ",  0x15: "rR",
    0x16: "sS",  0x17: "tT",  0x18: "uU",  0x19: "vV",  0x1A: "wW",  0x1B: "xX",
    0x1C: "yY",  0x1D: "zZ",  0x1E: "1!",  0x1F: "2@",  0x20: "3#",  0x21: "4$",
    0x22: "5%",  0x23: "6^",  0x24: "7&",  0x25: "8*",  0x26: "9(",  0x27: "0)",
    0x2C: "  ",  0x2D: "-_",  0x2E: "=+",  0x2F: "[{",  0x30: "]}",  0x32: "#~",
    0x33: ";:",  0x34: "'\"",  0x36: ",<",  0x37: ".>"
}
data = ''
for x in open("xd","r").readlines():
```

```
    code = int(x[4:6],16)
    print(x[4:6])
    if code == 0:
        continue
    if code == 0x28:
        print('ENTER!')
        print(data)
        data = ''
        continue
    upper = 0
    if int(x[0:2],16) == 0x02 or int(x[0:2],16) == 0x20:
        upper = 1
    data += usb_codes[code][upper]
print(data)
```

After parsing, the compressed package password is obtained and completed.

0x02 RR

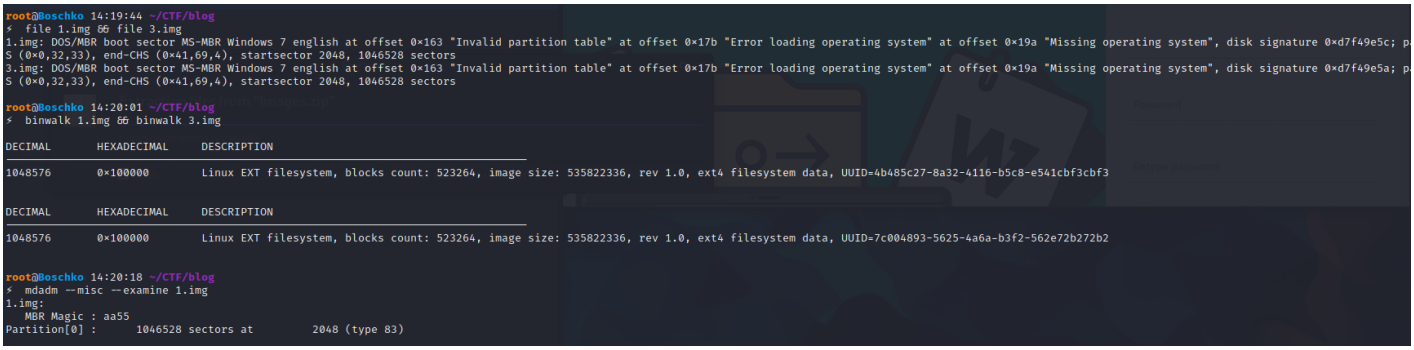




I got the title and found that there are three imgs, two of which are the same size, and the other is 0.

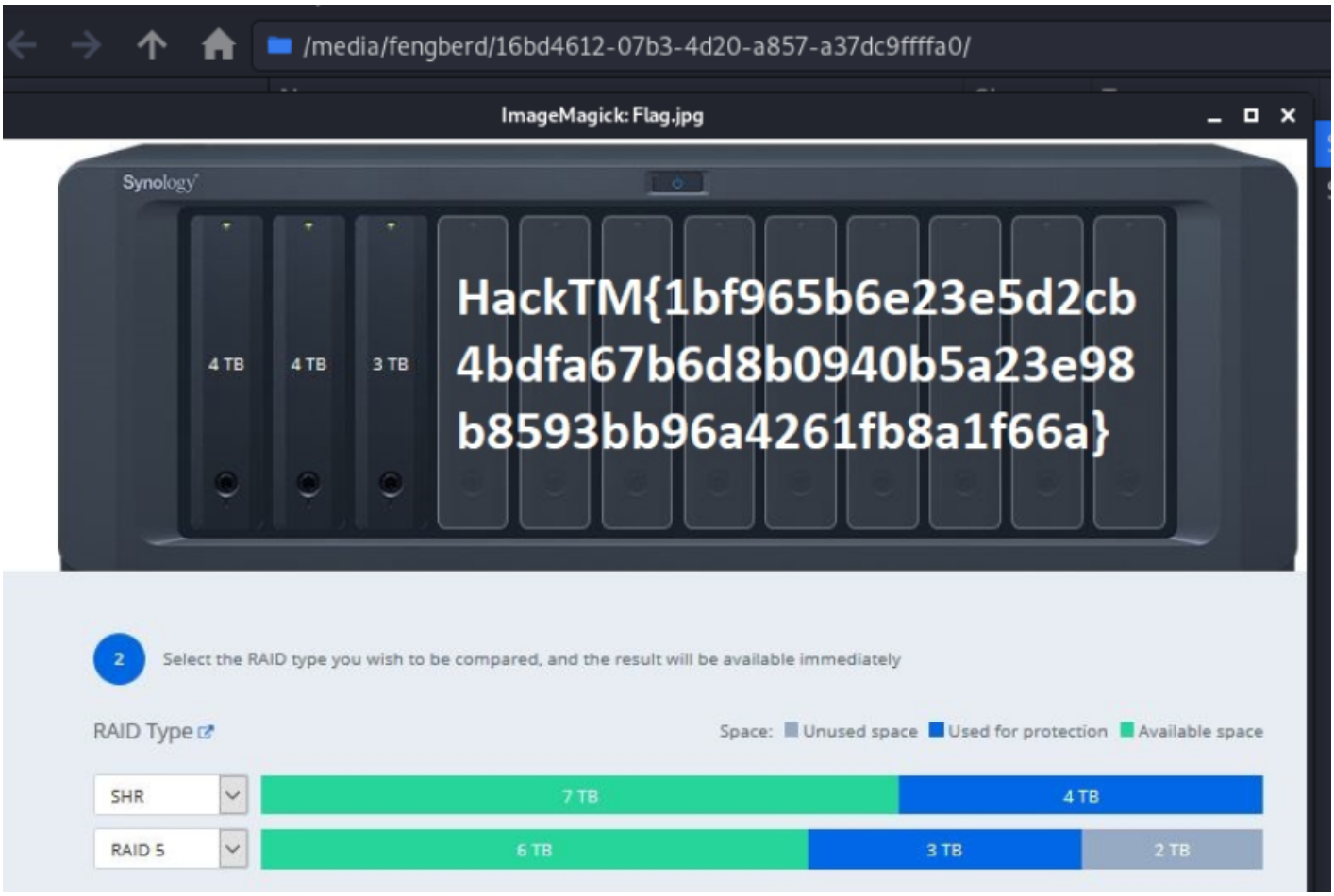
According to the meaning of the title "One of my drives failed", it is guessed that the file with a size of 0 is the damaged disk. According to the two disks can "recovering all my files", it may be a

RAID array



But there is a hole in this question. The beginning of the img file is filled with an invalid partition table. Using the file command or directly checking with mdadm will not be recognized as a RAID disk. Check with binwalk to find the correct offset.

Here, use dd to simply crop this file, and then use mdadm to view the detailed RAID information, it is indeed a RAID5. Next, create a loop and then do with losetup `mdadm --assemble --run /dev/md0 --readonly /dev/loop0 /dev/loop1` directly mount the hard drive on it. Here you go losetup -o rather use the dd process the file because encountered some mysterious Bug.



Revision #2
Created 3 October 2020 16:16:10 by Boschko
Updated 3 October 2020 18:26:31 by Boschko