After analizing the description and the .img files, it becomes clear we are dealing with Linux RAID. We are probably either dealing with RAID 0 or 5, so let's test for both. After struggling with RAID0 for a while, I conluded the correct version is RAID 5. And it makes sense, as we have lost one drive only. To recover it, all we need to do is xor the 2 other ones. I didn't get all the details, but a usefull thread is https://www.reddit.com/r/HomeServer/comments/1abunjh/how_does_raid_5_work_exactly/.

After the recovery, I tried to recreate the file system using various things, but I failed everytime. As a result, I decided to look through the recovered img file. Foremost didn't help, but binwalk had a lot of results, so I made a little command to search for some more common file types:

**sudo xxd -p 2.img | **

**grep -o -P "(ffd8ff|89504e47|47494638|25504446|504b0304|4344303000000100|49204920|494433|52617221|52494646|424d|377a277a)" | **

asd

**/ffd8ff/{jpeg++} **

**/89504e47/{png++} **

**/47494638/{gif++} **

**/25504446/{pdf++} **

**/504b0304/{zip++} **

**/4344303000000100/{iso++} **

**/49204920/{tiff++} **

**/494433/{mp3++} **

**/52617221/{rar++} **

**/52494646/{wav++} **

**/424d/{bmp++} **

**/377a277a/{seven_z++} **

END{print "JPEG: "jpeg"\nPNG: "png"\nGIF: "gif"\nPDF: "pdf"\nZIP: "zip"\nISO: "iso"

For a short summary, I am searching for file headers in the file. xxd –p is essential as it outputs only hex data and nothing more.

After executing we find out we have a singular png file, which is a bit suspicious.

sudo xxd -p 2.img | grep -b -o "ffd8ff"

141268582:ffd8ff

Since jpeg files, like png files, display even if they have trailing data, we only care to extract at the beginning.

Also, since linux wasn't doing what I wanted it to do for some reason, I made a python script to grab the image

```
def trim_before_jfif(input_file, output_file, max_size=10 * 1024 * 1024):
    with open(input_file, 'rb') as f:
        data = f.read()
    header_index = data.find(b'\xFF\xD8\xFF')
    trimmed_data = data[header_index:header_index + max_size]
    with open(output_file, 'wb') as f_out:
        f_out.write(trimmed_data)

trim_before_jfif("2_recovered.img", "trimmed_image.jpg")
```

Running this script recoveres an image that contains the flag. (I limited the file size to 10MB so I can use an online text extractor. I used deepseek and despite the fact that it made a couple mistakes, it saved a bit of eye straining).

Conclusion: There are still many things that I did not understand about this challenge, but as a general idea, the process was recovering the lost img file using knowledge about RAID, then extracting a jpeg image from said recovered img file.

Made with love by: AndreiCat