Underhanded Storage [350]

Task description: ./description.txt

TL;DR: hidden data in ext2 filesystem

Resources: The Sleuth Kit

So, we receive the same filesystem used for all the tasks in Operation Bad F.S. The same story as before, an EXT2 filesystem, we can mount it in linux, but for this task ***we shouldn't***.

```
root@mecha:/home/catalin/ctf/16-csc-quals# file bad_fs
bad_fs: Linux rev 1.0 ext2 filesystem data, UUID=e8118091-96d9-444f-8abd-67c097afa60c
```

After analyzing the last tasks in this category, we can see that the only difference between this filesystem and the original OpenWRT firmware, is the key.gif file, and I already exhausted all the checks on it. So the next thought I had in mind, was that there has to be something in the filesystem.

```
[root@longhorn ~]# git diff --no-index x/ y/
diff --git a/x/www/luci-static/resources/cbi/key.gif b/y/www/luci-static/resources/cbi/key.gif
index 95687a6..e3853e5 100644
Binary files a/x/www/luci-static/resources/cbi/key.gif and b/y/www/luci-static/resources/cbi/key.gif differ
[root@longhorn ~]#
```

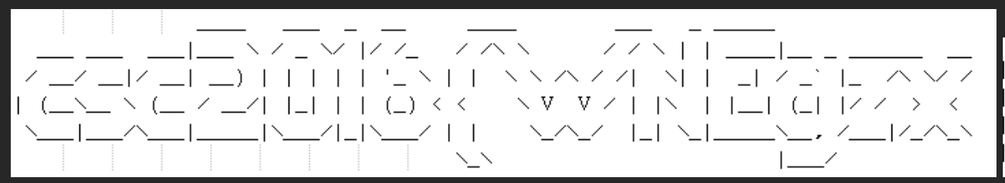
Since EXT2 is "cool", we know that we can use the filesystem structure to hide data in blocks. That's why tools like The Sleuth Kit appeared. Running strings on it gives a bunch of "csc2016" strings, which are not really found in the filesystem when mounted. After turning over to istat, (a tool from TSK, which, according to the manual pages, "istat - Display details of a meta-data structure (i.e. inode)") we can find some details about the structure of the inodes from the filesystem. After manually checking the first few inodes, I ran into the inode nr. 12 which had "Extended attributes" user.csc2016=<some ascii>. Ohwell, this also happens on the inode nr 14, and so on, so there has to be some flag somewhere.. Let's find out the number of inodes using: **tune2fs –l bad_fs | grep inode**

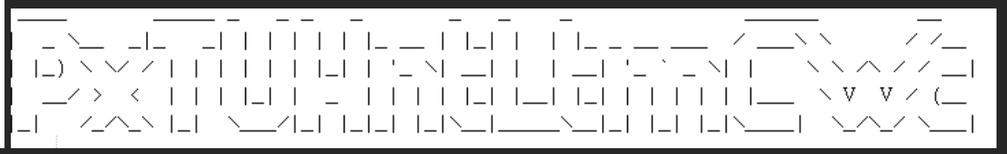
```
root@mecha:/home/catalin/ctf/16-csc-quals# tune2fs -l bad_fs | grep inode
Free inodes:              5041
First inode:              11
```

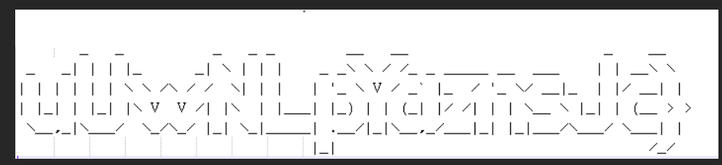
Since there seems to be a lot of data, we should extract it into a file and analyse it afterwards. Bash is here to help us!

for i in `seq 1 6000`; do istat bad_fs $i | grep "Extended Attributes" -A 15 ; done > underhanded_storage_inode_data

But after that, manually skimming through almost 9 thousand lines of text is kinda harsh.. (Also found there a subtle joke "Additionally, please enjoy this issue of Phrack while skimmin"). Also, ctrl+f after the flag didn't really work, but it had to be somewhere among those lines. Seems like we need to do the manual check through it. We can notice some sort of ASCII art, and that should be it. After extracting all the parts of the ASCII art, and "rebuilding" it.. (around 1 hour spent only on doing this, twice actually since I skipped a part the first time). Also, here is a screenshot from the building process:







Final Flag: **csc2016{WNEgzxPxTUHntLtmCWcuUwNLpYaznsJc}**