Paleontology

Challenge

Dinosaurs lived a long time ago, but they left traces of their existence all over! Grab your paleontology tools and learn all about fossil hunting on this dino digging adventure!

paleontology.jpg

Solution

paleontology.jpg

Start by extracting hidden files from the image using binwalk:

```
binwalk -e paleontology.jpg
```

Inside the `_paleontology.jpg.extracted` folder, you'll find that two files were successfully extracted, `layers_of_dirt.7z` and `trapped_in.ice`.

You can attempt to unzip `layers_of_dirt.7z` using `7z e layers_of_dirt.7z`, but it will prompt you for a password, so move onto inspecting `trapped_in.ice` for now.

trapped_in.ice

If you run `file trapped_in.ice`, you won't get much info back. With some Googling, however, you may be able to determine that this is an ICEOWS archive.

There are two ways of opening this file.

The Hard Way

This way involves using ICEOWS to extract the file. I originally intended for this to be the only way to extract the file, but I determined that requiring that was beyond the scope of this challenge, since the program doesn't seem to work properly in Windows 10 (it is only officially supported up to Windows XP). To solve it this way, you will need a VM (or just a really old computer) running old version of Windows. Install ICEOWS, then use it to extract `trapped_in.ice`. A new file, `trapped_in` will be created.

The Easy Way

The other way of solving this shouldn't technically be possible, but I added it in to make things a little easier. Using the Extract Files recipe in CyberChef, the image can be extracted from `trapped_in.ice` (binwalk doesn't seem to work quite as well in this case). Unchecking "Archives" will make it easier to find the PNG. Note that this file being extracted isn't actually part of the ICEOWS file, but was rather appended to it to make things a little easier.

trapped_in

If you haven't already determiend the filetype of `trapped_in`, `file trapped_in` will tell you that it is a PNG image.

The file looks like this:

If you inspect the image closely, you'll notice some very small text:

The text reads: `tail -c 97341 paleontology.jpg > la_brea.tar.pit`

Running this command will extract the last 97341 bytes from `paleontology.jpg` and write them to a new file, `la_brea.tar.pit`.

la_brea.tar.pit

Again, the `file` command won't really provide any valuable information, but using Google or another tool such as TrID, you should be able to determine that this is a PackIt archive. Depending on what archival programs you have installed, your computer may also just recognize this file automatcially.)

There are various programs that can extract pit files, but if you don't already have one and don't want to install anything, you should be able to find a website to extract it for you.

extract.me is one option that will work.

la_brea.tar

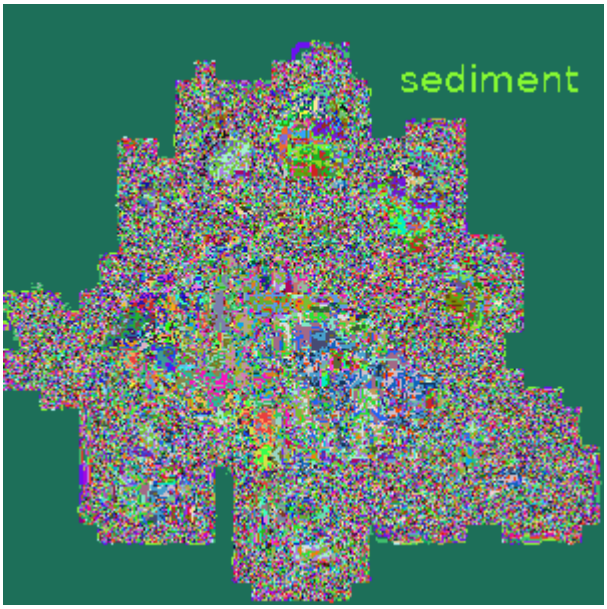
Inside the pit file is a tar file that can be easily extracted:

```
tar -xvf la_brea.tar
```

Inside the extracted folder will be another file, `steg.png`.

steg.png

Using CyberChef's Randomize Color recipe, you can find hidden text in the image.



This reveals the word `sediment`, which can be used as the password for `layers_of_dirt.7z`.

layers_of_dirt.7z

The following command will extract `layers_of_dirt.7z` using the password `sediment`:

```
7z e layers_of_dirt.7z -psediment
```

dirt.zip

Inside `layers_of_dirt.7z` you will find `dirt.zip`. Inside `dirt.zip` you will find... `more_dirt/`, which contains `more_dirt.zip`. And inside that? `even_more_dirt/`. You get the idea.

The zip structure recurses pretty far down, so you'll want to use a script to extract them all.

Your script might look something like this:

```
# based on scripts found here:
# https://unix.stackexchange.com/questions/4367/extracting-nested-zip-files
# https://askubuntu.com/questions/146634/shell-script-to-move-all-files-from-su

while [ "`find . -type f -name '*.zip' | wc -l`" > 0 ]
do
    unzip *.zip
    rm *.zip
    find . -mindepth 2 -type f -print -exec mv {} . \;
    rmdir */
done
```

fossil.jpg

Under all those layers of dirt, you'll find `fossil.jpg`!

Using `strings fossil.jpg` you can find the flag:

```
byuctf{f055il5_4r3_4m4zing!}
```