Pack Man

We are given a zip file named pack_man.zip
we extract this file

```
┌──(docx㉿kali)-[~/pack man]
└─$ ls
pack_man.zip

┌──(docx㉿kali)-[~/pack man]
└─$ unzip pack_man.zip
Archive:  pack_man.zip
   creating: pack_man/.git/
   creating: pack_man/.git/logs/
  inflating: pack_man/.git/logs/HEAD
   creating: pack_man/.git/logs/refs/
   creating: pack_man/.git/logs/refs/heads/
  inflating: pack_man/.git/logs/refs/heads/master
 extracting: pack_man/.git/ORIG_HEAD
 extracting: pack_man/.git/COMMIT_EDITMSG
 extracting: pack_man/.git/HEAD
  inflating: pack_man/.git/index
  inflating: pack_man/.git/description
   creating: pack_man/.git/refs/
   creating: pack_man/.git/refs/tags/
   creating: pack_man/.git/refs/heads/
 extracting: pack_man/.git/refs/heads/master
  inflating: pack_man/.git/packed-refs
   creating: pack_man/.git/objects/
   creating: pack_man/.git/objects/pack/
  inflating: pack_man/.git/objects/pack/pack-a7bd0dc405aa6c6f97081f11d7df9ffaba261cbd.rev
  inflating: pack_man/.git/objects/pack/pack-a7bd0dc405aa6c6f97081f11d7df9ffaba261cbd.pack
   creating: pack_man/.git/objects/info/
  inflating: pack_man/.git/objects/info/commit-graph
 extracting: pack_man/.git/objects/info/packs
   creating: pack_man/.git/info/
  inflating: pack_man/.git/info/refs
  inflating: pack_man/.git/info/exclude
  inflating: pack_man/.git/config
   creating: pack_man/.git/hooks/
  inflating: pack_man/.git/hooks/applypatch-msg.sample
  inflating: pack_man/.git/hooks/pre-merge-commit.sample
  inflating: pack_man/.git/hooks/update.sample
  inflating: pack_man/.git/hooks/prepare-commit-msg.sample
  inflating: pack_man/.git/hooks/sendemail-validate.sample
  inflating: pack_man/.git/hooks/pre-receive.sample
  inflating: pack_man/.git/hooks/pre-applypatch.sample
  inflating: pack_man/.git/hooks/pre-commit.sample
  inflating: pack_man/.git/hooks/fsmonitor-watchman.sample
  inflating: pack_man/.git/hooks/post-update.sample
  inflating: pack_man/.git/hooks/pre-rebase.sample
  inflating: pack_man/.git/hooks/commit-msg.sample
  inflating: pack_man/.git/hooks/pre-push.sample
  inflating: pack_man/.git/hooks/push-to-checkout.sample
```

which gives us a folder

```
┌──(docx㉿kali)-[~/pack man]
└─$ ls
pack_man  pack_man.zip

┌──(docx㉿kali)-[~/pack man]
└─$
```

inside the folder there is only a .git folder, nothing else

```
┌──(docx㉿kali)-[~/pack man/pack_man]
└─$ ls -la
total 12
drwxrwxr-x 3 docx docx 4096 Dec 27 23:29 .
drwxrwxr-x 3 docx docx 4096 Dec 27 23:29 ..
drwxrwxr-x 7 docx docx 4096 Dec 27 23:09 .git
```

so we try git log

```
┌──(docx㉿kali)-[~/pack man/pack_man]
└─$ git log
fatal: cannot read commit object cd0e58b8ebd9e6c2ee8c35c36a33b40f7137b631
commit cd0e58b8ebd9e6c2ee8c35c36a33b40f7137b631
```

so we now know git cannot find the data for that commit hash
so we investigate the file structure to see if the data is missing or just corrupted
we will look into the .git folder to see what files are actually there

```
┌──(docx㉿kali)-[~/pack man/pack_man]
└─$ ls -la .git/objects/pack/
total 5136
drwxrwxr-x 2 docx docx    4096 Dec 27 23:09 .
drwxrwxr-x 4 docx docx    4096 Dec 27 23:09 ..
-r--r--r-- 1 docx docx 5244997 Dec 27 23:09 pack-a7bd0dc405aa6c6f97081f11d7df9ffaba261cbd.pack
-r--r--r-- 1 docx docx      76 Dec 27 23:09 pack-a7bd0dc405aa6c6f97081f11d7df9ffaba261cbd.rev
```

there is a pack file but no .idx file so we will try to extract readable text from the binary file to see if the flag is just sitting there

which returns nothing
so strings failed, which means the data inside the .pack file is compresses (git uses Zlib)
we cant read it with simple tools
and we cant use git commands because the index is missing

so we have to write a script to manually find and decompress the Zlib streams inside the raw binary file
we know Zlib streams usually start with the byte 0x78

so we use this python code to find the flag

```
python3 -c "import zlib,glob;
path = glob.glob('.git/objects/pack/*.pack')[0];
print(f'Scanning {path}...');
with open(path, 'rb') as f:
    data=f.read();
    # Scan for Zlib header (0x78) and attempt decompression
    for i in range(len(data)):
        if data[i] == 0x78:
            try:
                d=zlib.decompress(data[i:]);
                if b'SGCTF' in d:
                    print('\n[+] SUCCESS: Found flag object ->', d);
                    break
        except: pass"
```

which gives us the flag

```
┌──(docx㉿kali)-[~/pack man/pack_man]
└─$ python3 -c "import zlib,glob;
path = glob.glob('.git/objects/pack/*.pack')[0];
print(f'Scanning {path}...');
with open(path, 'rb') as f:
    data=f.read();
    # Scan for Zlib header (0×78) and attempt decompression
    for i in range(len(data)):
        if data[i] == 0×78:
            try:
                d=zlib.decompress(data[i:]);
                if b'SGCTF' in d:
                    print('\n[+] SUCCESS: Found flag object →', d);
                    break
            except: pass"
Scanning .git/objects/pack/pack-a7bd0dc405aa6c6f97081f11d7df9ffaba261cbd.pack...

[+] SUCCESS: Found flag object → b'SGCTF{g1t_int3rnals_ar3_fun_and_scary}\n'
```

the flag is SGCTF{g1t_int3rnals_ar3_fun_and_scary}