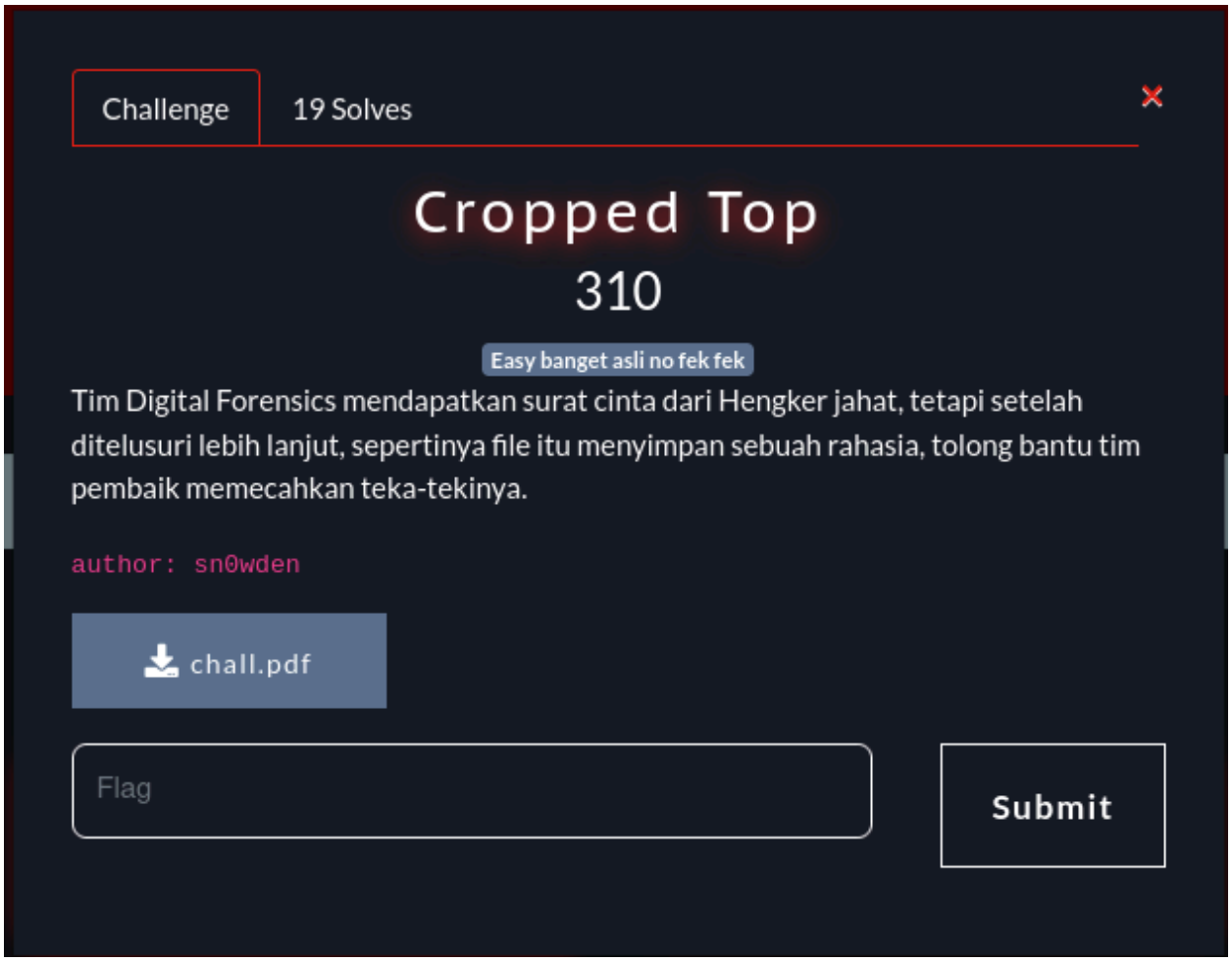
Write-up Cropped Top

Player: Constantine
Kategori: Forensik

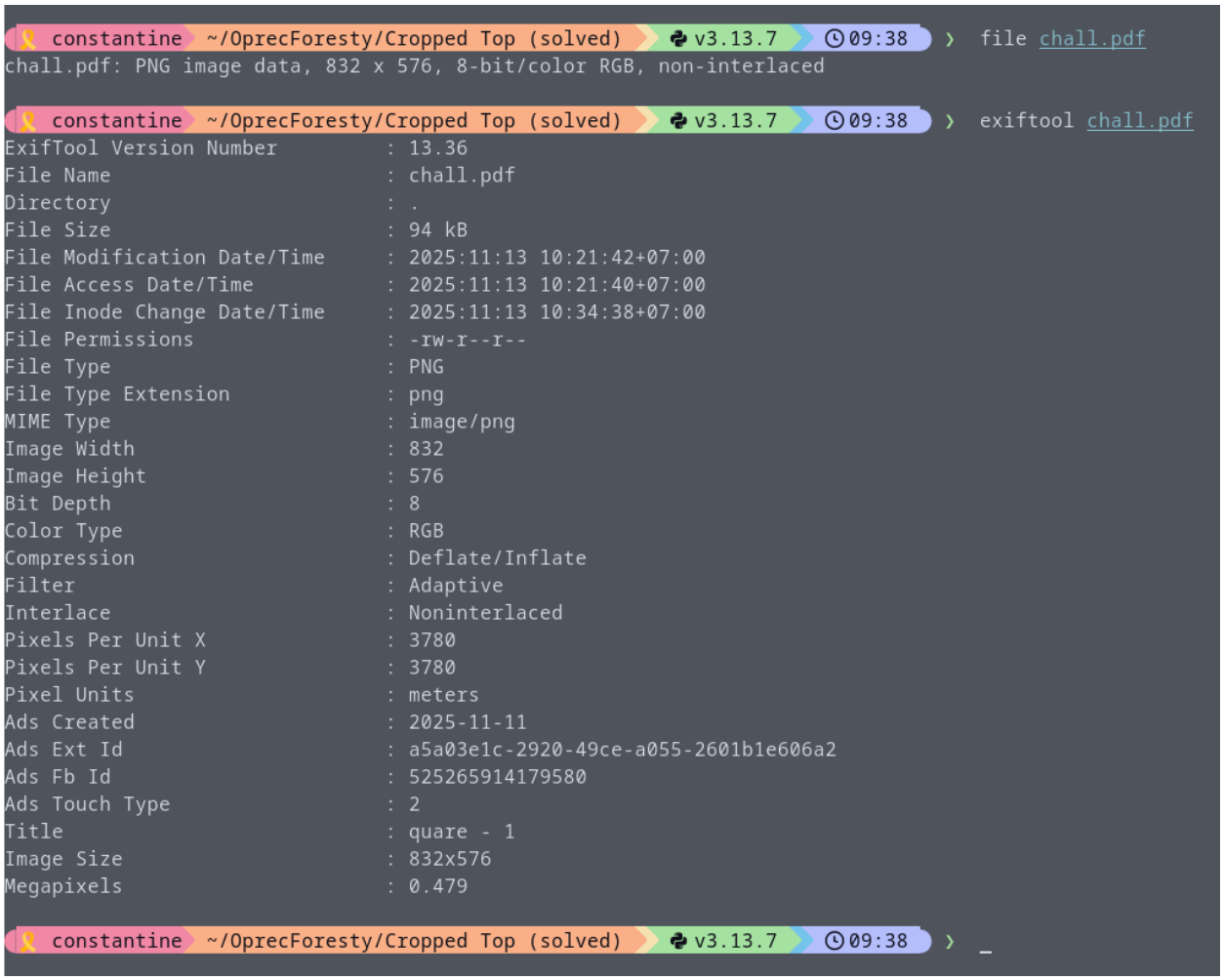


Konteks Dikit

Nama challengenya **Cropped Top**. File yang dikasih: `chall.pdf`. Begitu dibuka, PDF viewer langsung ngasih tau kalau file corrupt. Jadi kita harus bongkar PDF yang aneh ini, cari tahu sebenarnya file apa, dan balikin gambar aslinya sampai flag muncul. Kok bisa tahu kalau ini gambar? Liat aja hint besar dari judul, ada sesuatu yang tercrop di bagian bawah. Jadi kita expect bakal main di level struktur PNG / zlib stream, bukan cuma metadata doang.

Phase 1 – Recon

Pertama seperti biasa: lihat tipe file beneran.



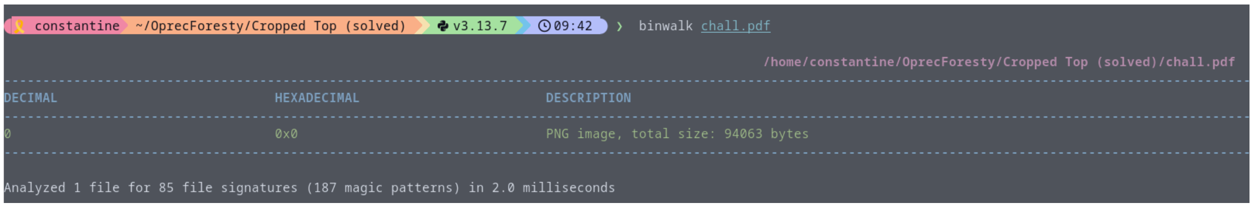
Dari output jadi jelas:

- Ekstensinya `.pdf` tapi,
- Magic bytes & metadata bilang ini PNG image.

Di hex view juga kelihatan kalau ini PNG:

89 50 4E 47 0D 0A 1A 0A ; 0x89504E470D0A1A0A -> PNG signature

Lanjut pakai `binwalk`:



Hasilnya:

0x0 PNG image, total size: 94063 bytes

Tidak ada file lain yang nempel (nggak ada ZIP, PDF beneran, dsb). Berarti satu file PNG utuh, tapi sengaja dibuat rusak.

Cek stego juga:

`zsteg` nunjukin:

extradata:imagedata ..
00000000: 00 f8 f8 f8 ...
meta XML:com.adobe.xmp..
...
Encoding::CompatibilityError

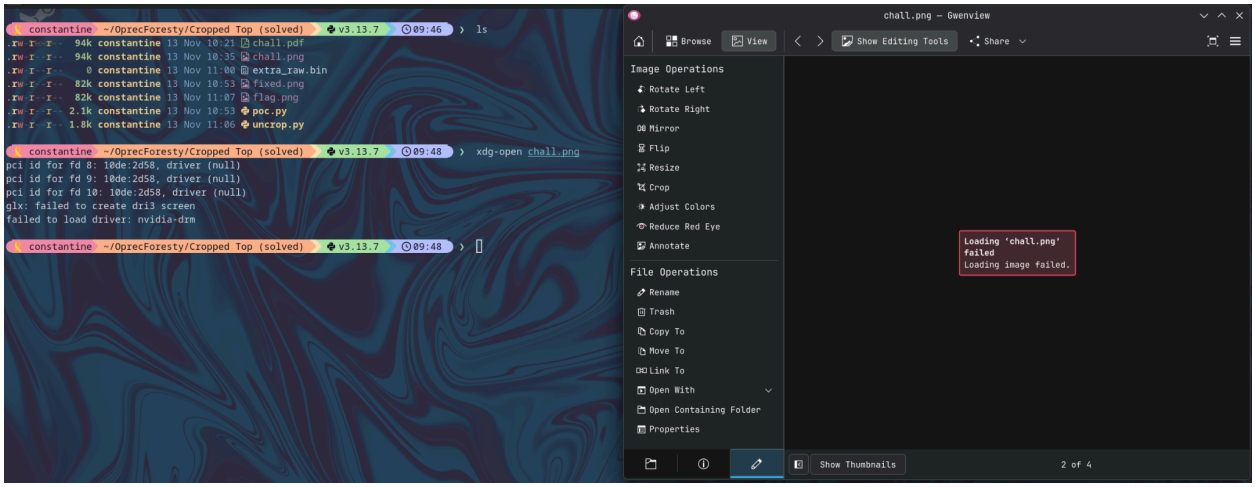
Artinya:

- Ada banyak byte `0xF8` di image data (mungkin noise / junk).
- `zsteg` crash karena XMP metadata dari Adobe (encoding issue).

Coba buka sebagai gambar (rename dulu):

cp chall.pdf chall.png
xdg-open chall.png

Viewer langsung: **Loading image failed**.



Jadi sekarang kita tau:

- File itu PNG
- Metadata normal
- Tapi chunk data gambarnya bikin corrupt.

Langsung aja bedah struktur PNG-nya.

Phase 2 – Benerin PNG biar minimal bisa kebuka

Dugaan awal IDAT (chunk yang nyimpen data gambar ter-compress) berisi zlib stream yang ada junk di belakang, sehingga library PNG gagal decode. Cara restorenya:

1. Parse chunk PNG manual.
2. Gabung semua `IDAT`.

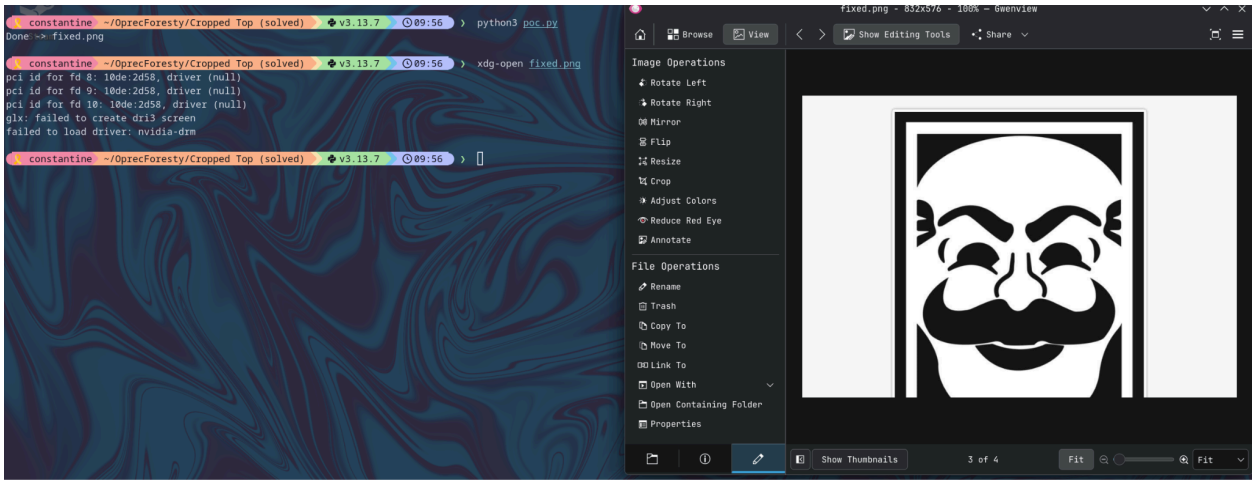
3. `zlib.decompress()` datanya.
4. Simpan output yang valid, buang noise, compress ulang → buat PNG baru.

Di sini aku dibantu ChatGPT buat nulis script Python pertama, `poc.py`:

```python
#!/usr/bin/env python3
import struct, zlib

FNAME = "chall.pdf"

data = open(FNAME, "rb").read()

assert data[:8] == b"\x89PNG\r\n\x1a\n", "Not a PNG file"

pos = 8
ihdr_data = None
other_chunks = []
idat_data = b""

while pos + 8 <= len(data):
    length = int.from_bytes(data[pos:pos+4], "big")
    ctype  = data[pos+4:pos+8]
    pos   += 8
    if pos + length + 4 > len(data):
        break

    cdata = data[pos:pos+length]
    crc   = data[pos+length:pos+length+4]
    pos  += length + 4

    if ctype == b"IHDR":
        ihdr_data = cdata
    elif ctype == b"IDAT":
        idat_data += cdata
    elif ctype == b"IEND":
        break
    else:
        other_chunks.append((ctype, cdata))

assert ihdr_data is not None, "IHDR not found"
assert idat_data, "No IDAT data"
```

```
decomp = zlib.decompressobj()
raw = b""
chunk = idat_data

try:
    raw = decomp.decompress(chunk)
except Exception as e:
    raw += decomp.unconsumed_tail

if not raw:
    raise SystemExit("Failed to decompress IDAT")

# recalculate CRC!
def make_chunk(ctype, cdata):
    length = len(cdata).to_bytes(4, "big")
    crc = zlib.crc32(ctype + cdata) & 0xFFFFFFFF
    crc = crc.to_bytes(4, "big")
    return length + ctype + cdata + crc

new_idat = zlib.compress(raw, 9)

out = b"\x89PNG\r\n\x1a\n"
out += make_chunk(b"IHDR", ihdr_data)

for ctype, cdata in other_chunks:
    out += make_chunk(ctype, cdata)

out += make_chunk(b"IDAT", new_idat)
out += make_chunk(b"IEND", b"")

open("fixed.png", "wb").write(out)
print("Done -> fixed.png")
```

Jalankan poc.py:



Gambarnya akhirnya kebuka… tapi:

- Gambar cuma nunjukin logo fsociety dari Mr.Robot.
- Bottom area kelihatan kecrop, nggak ada teks / flag.

Masih belum ada flag. Lanjut Phase 3.

Phase 3 – Cek data

Langkah berikutnya adalah cek apakah zlib stream punya sisa bytes (unused data).

Aku dibuatkan script cek decompress (lagi-lagi dibantu ChatGPT), sebut saja `uncrop.py`:

```
#!/usr/bin/env python3
import struct, zlib

FNAME = "chall.pdf"

data = open(FNAME, "rb").read()

# 1. pastikan PNG
assert data[:8] == b"\x89PNG\r\n\x1a\n", "Not a PNG"

pos = 8
```

```
ihdr_data = None
idat_data = b""
other_chunks = []

# 2. parse semua chunk
while pos + 8 <= len(data):
    length = int.from_bytes(data[pos:pos+4], "big")
    ctype  = data[pos+4:pos+8]
    pos   += 8
    if pos + length + 4 > len(data):
        break

    cdata = data[pos:pos+length]
    crc   = data[pos+length:pos+length+4]
    pos  += length + 4

    if ctype == b"IHDR":
        ihdr_data = cdata
    elif ctype == b"IDAT":
        idat_data += cdata
    elif ctype == b"IEND":
        break
    else:
        other_chunks.append((ctype, cdata))

assert ihdr_data is not None, "IHDR not found"
assert idat_data, "No IDAT"

# 3. ambil info dasar dari IHDR
width, height, bit_depth, color_type, compression, filter_m, interlace = \
    struct.unpack(">IIBBBBB", ihdr_data)
# bytes per pixel per warna
bpp_map = {0:1, 2:3, 3:1, 4:2, 6:4}
bpp = bpp_map[color_type]
rowlen = 1 + width * bpp   # 1 byte filter + data pixel

print(f"width={width}, height={height}, bit_depth={bit_depth},
color_type={color_type}")
print(f"rowlen={rowlen}")

# 4. decompress IDAT
dec = zlib.decompressobj()
raw = dec.decompress(idat_data)
```

```
extra = dec.unused_data

print(f"len(raw)   = {len(raw)}")
print(f"len(extra) = {len(extra)}")

# simpan extra buat analisis tambahan
open("extra_raw.bin", "wb").write(extra)
print("Saved extra_raw.bin")

# cek apakah extra bisa dianggap scanline tambahan
extra_rows = len(extra) // rowlen
print(f"extra_rows (integer division) = {extra_rows}")
print(f"len(extra) % rowlen = {len(extra) % rowlen}")

def make_chunk(ctype, cdata):
    length = len(cdata).to_bytes(4, "big")
    crc = zlib.crc32(ctype + cdata) & 0xffffffff
    return length + ctype + cdata + crc.to_bytes(4, "big")

# 5a. kalau extra align dengan rowlen → coba treat sebagai baris ATAS
gambar
if extra_rows > 0 and len(extra) % rowlen == 0:
    new_height = height + extra_rows
    # rebuild IHDR dengan height baru
    new_ihdr = struct.pack(">IIBBBBB",
                           width, new_height,
                           bit_depth, color_type,
                           compression, filter_m, interlace)

    # gabung: extra (baris atas) + raw (baris lama)
    new_raw_top = extra + raw
    new_idat_top = zlib.compress(new_raw_top, 9)

    out_top = b"\x89PNG\r\n\x1a\n"
    out_top += make_chunk(b"IHDR", new_ihdr)
    for ctype, cdata in other_chunks:
        out_top += make_chunk(ctype, cdata)
    out_top += make_chunk(b"IDAT", new_idat_top)
    out_top += make_chunk(b"IEND", b"")

    open("uncropped_top.png", "wb").write(out_top)
    print("Wrote uncropped_top.png")
```

```
    # 5b. versi lain: extra sebagai baris BAWAH
    new_raw_bottom = raw + extra
    new_idat_bottom = zlib.compress(new_raw_bottom, 9)

    out_bottom = b"\x89PNG\r\n\x1a\n"
    out_bottom += make_chunk(b"IHDR", new_ihdr)
    for ctype, cdata in other_chunks:
        out_bottom += make_chunk(ctype, cdata)
    out_bottom += make_chunk(b"IDAT", new_idat_bottom)
    out_bottom += make_chunk(b"IEND", b"")

    open("uncropped_bottom.png", "wb").write(out_bottom)
    print("Wrote uncropped_bottom.png")

else:
    print("extra data does NOT align to whole scanlines; analyze
extra_raw.bin manually.")
```

Output:

```
constantine ~/OprecForesty/Cropped Top v3.13.7 11:00 ❯ python uncrop.py
width=832, height=576, bit_depth=8, color_type=2
rowlen=2497
len(raw)   = 2077504
len(extra) = 0
Saved extra_raw.bin
extra_rows (integer division) = 0
len(extra) % rowlen = 0
extra data does NOT align to whole scanlines; analyze extra_raw.bin manually.
```

Kita cuma dapet `extra_raw.bin` dan itu ternyata **kosong**.

```
constantine ~/OprecForesty/Cropped Top (solved) v3.13.7 10:07 ❯ strings extra_raw.bin

constantine ~/OprecForesty/Cropped Top (solved) v3.13.7 10:07 ❯ binwalk extra_raw.bin
Analyzed 1 file for 85 file signatures (187 magic patterns) in 4.0 milliseconds
```

Awalnya keliatan kayak dead end, tapi justru di sini clunya:

Di bantu saudara chatGPT, kita hitung `rowlen`:

```
rowlen = 1 + width * 3 = 1 + 832 * 3 = 2497
```

Kalau height memang 576, seharusnya total bytes raw:
`expected = 2497 * 576 = 1,438,272`

Tapi `len(raw)` hasil decompress = **2,077,504** bytes. Kalau dibagi:

```
2,077,504 / 2,497 = 832
```

Pas banget, tanpa sisa.

Artinya, IDAT berisi **832 scanline**, tapi IHDR bilang height-nya cuma 576.

Inilah kenapa viewer awalnya gagal, jumlah data nggak match sama header. `fix_png.py` yang pertama cuma bantu bikin streamnya legal buat 576 row, jadi bottom sisanya kepotong.

Phase 4 – Rekonstruksi gambar full (uncrop height)

Sekarang solusinya pakai semua scanline, dan ubah `height` di IHDR jadi `total_rows` (832). Lalu compress ulang dan tulis PNG baru. Script kedua (lagi-lagi credit kalkulasi & struktur ke ChatGPT) kita sebut `uncrop_full.py`:

```
#!/usr/bin/env python3
import struct, zlib

FNAME = "chall.pdf"   # atau "chall.png" kalau sudah rename

data = open(FNAME, "rb").read()
assert data[:8] == b"\x89PNG\r\n\x1a\n", "Not a PNG"

pos = 8
ihdr = None
idat = b""

# --- parse IHDR & IDAT ---
while pos + 8 <= len(data):
    length = int.from_bytes(data[pos:pos+4], "big")
    ctype  = data[pos+4:pos+8]
    pos   += 8
    cdata = data[pos:pos+length]
    pos   += length
    crc   = data[pos:pos+4]
    pos   += 4

    if ctype == b"IHDR":
        ihdr = cdata
    elif ctype == b"IDAT":
```

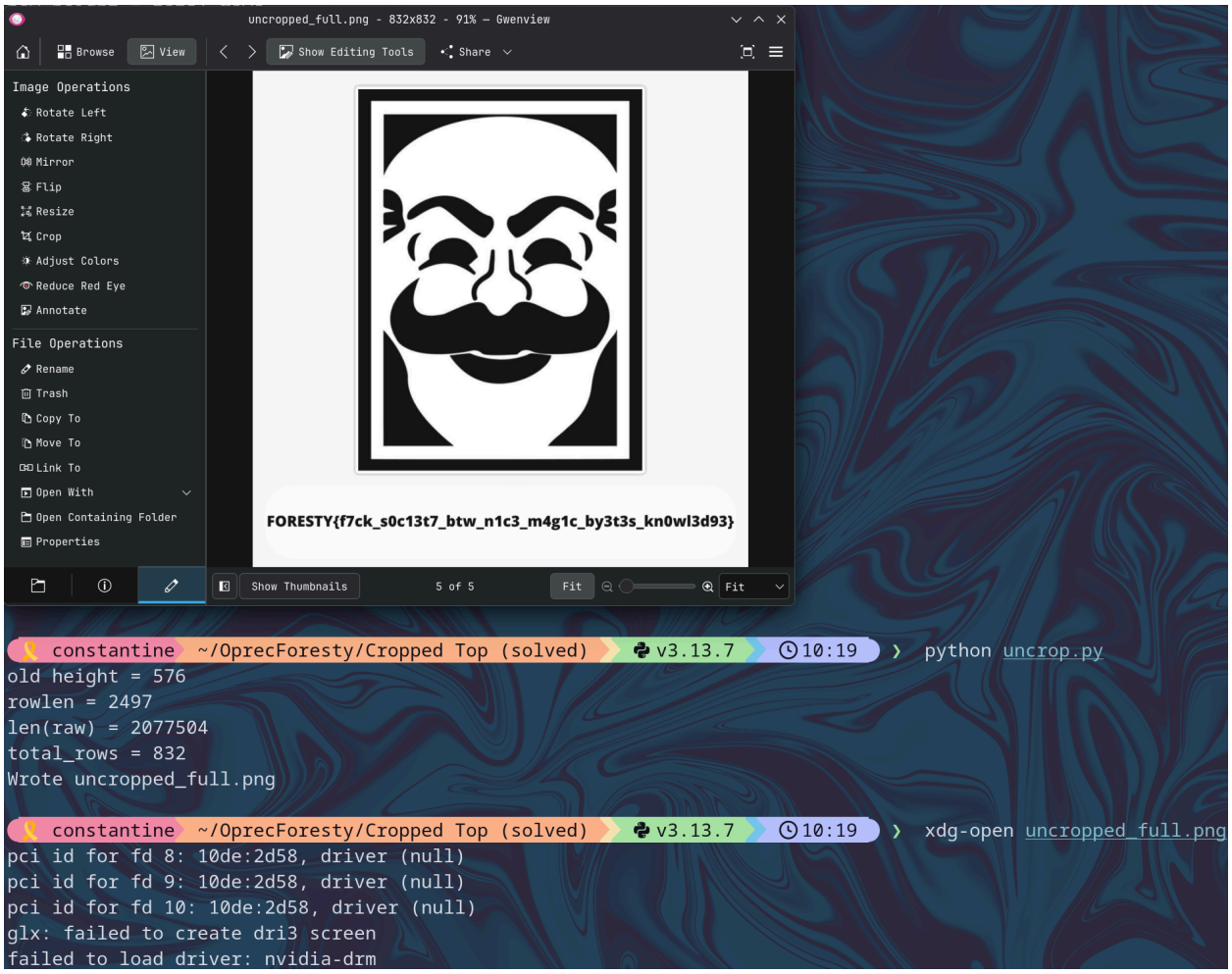
```
        idat += cdata
    elif ctype == b"IEND":
        break

assert ihdr is not None, "IHDR not found"
assert idat, "No IDAT data"

# --- info asli dari IHDR ---
width, height, bit_depth, color_type, comp, filt, inter = \
    struct.unpack(">IIBBBBB", ihdr)

bpp_map = {0:1, 2:3, 3:1, 4:2, 6:4}
bpp = bpp_map[color_type]
rowlen = 1 + width * bpp

print("old height =", height)
print("rowlen =", rowlen)

# --- decompress semua IDAT ---
raw = zlib.decompress(idat)
print("len(raw) =", len(raw))

total_rows = len(raw) // rowlen
print("total_rows =", total_rows)

# pakai semua row sebagai gambar
new_height = total_rows

# --- build IHDR baru ---
new_ihdr = struct.pack(">IIBBBBB",
                       width, new_height,
                       bit_depth, color_type,
                       comp, filt, inter)

# compress ulang seluruh raw
new_idat = zlib.compress(raw, 9)

def chunk(ctype, cdata=b""):
    length = len(cdata).to_bytes(4, "big")
    crc = zlib.crc32(ctype + cdata) & 0xFFFFFFFF
    return length + ctype + cdata + crc.to_bytes(4, "big")

png = b"\x89PNG\r\n\x1a\n"
```

```
png += chunk(b"IHDR", new_ihdr)
png += chunk(b"IDAT", new_idat)
png += chunk(b"IEND")

open("uncropped_full.png", "wb").write(png)
print("Wrote uncropped_full.png")
```

Jalankan:

```
constantine ~/OprecForesty/Cropped Top (solved) v3.13.7 10:19 ❯ python uncrop.py
old height = 576
rowlen = 2497
len(raw) = 2077504
total_rows = 832
Wrote uncropped_full.png
```

Kali ini gambar kebuka dengan tinggi **832** pixel.
Di bagian yang sebelumnya nggak kelihatan, muncul teks flag besar yang jelas banget:



Phase 5 – Ambil Flag

Tinggal dibaca:

FORESTY{f7ck_s0c13t7_btw_n1c3_m4g1c_by3t3s_kn0wl3d93}