Rorecovery 12(incomplete solve)

Challenge

I was given a single, unknown file with no description or URL.
Goal: find a hidden flag or secret, presumed to be related to Roblox.

Step 1 — Identifying the File

What I did:

- Ran `file` → no helpful output.
- Opened in a hex viewer (`xxd` and `hexdump`) — found it starts with `<rblx!`.
- Confirmed: this is a Roblox binary model file (`.rbxm`).

Lesson: Many Roblox files are custom binary containers with recognizable sections like `PROP`, `PRNT`, and `END`.

Step 2 — Trying to Open It

- Tried opening directly in Roblox Studio.
- Studio gave an error: the file didn't load correctly.
- So direct inspection wasn't possible.

Step 3 — Finding Asset IDs Manually

- Instead, searched the hex dump for `SourceAssetId` and other possible numeric references.
- Found multiple numeric IDs — likely asset references to Roblox's online CDN.
- Downloaded them manually.

Step 4 — Automating with Bash

To speed up ID downloads:

- I wrote this Bash script to:
    - Extract all numeric sequences (length ≥ 6)
    - Try downloading each one from Roblox's asset API.

```
#!/usr/bin/env bash
```

```
# Extract numeric IDs from hex dump
grep -oE '[0-9]{6,}' hexdump.txt | sort -u > ids.txt

# Make output directory
mkdir -p assets

# Loop through each ID and download
while read id; do
    echo "Trying ID: $id"
    curl -s -o "assets/$id"
"https://assetdelivery.roblox.com/v1/asset?id=$id"
done < ids.txt

echo "Download attempt complete. Check 'assets/' directory."
```

Step 5 — What I Found

- Some downloaded assets were ZIP files.
- Unzipped them: found images like a sphere, backspace symbol, etc.
- Also found a KeyframeSequence animation named **flip** inside the metadata of a file.
    - But its actual asset ID was not clearly present in the file or dump.
    - So I couldn't download the flip animation itself.

What I Learned

- Inspecting Roblox binary model structure with `xxd` and `hexdump`.
- How Roblox links assets via numeric IDs.
- Automating repetitive download tests with Bash.
- Unzipping and inspecting the CDN assets for hidden clues.

Current Status

- I have extracted all visible IDs and their content.
- I found partial evidence (the "flip" animation name) but no ID to fetch it.
- Couldn't find the flag in any downloaded content.

Next Steps

If I revisit:

- Try deeper parsing: write a Roblox binary decoder to extract strings more accurately.
- Explore alternative sources for the "flip" animation ID: perhaps derived from references or child objects.
- Analyze assets for hidden steganography or metadata.

Files

File	Purpose
`file.rbxm`	Original Roblox binary model
`hexdump.txt`	Full hex dump
`assets/`	Downloaded asset files
`extract_ids.sh`	Bash script for automating downloads

Key Bash Script

```
#!/usr/bin/env bash

# Extract numeric IDs from hexdump and download them.

grep -oE '[0-9]{6,}' hexdump.txt | sort -u > ids.txt

mkdir -p assets

while read id; do
    echo "Trying ID: $id"
    curl -s -o "assets/$id"
"https://assetdelivery.roblox.com/v1/asset?id=$id"
done < ids.txt

echo "All downloads attempted. Check 'assets/' folder."
```

Summary

This challenge was a deep dive into:

- Roblox binary structures,
- Heuristic ID scraping,
- Practical scripting to brute-force asset recovery.

Still unsolved, but a valuable forensics exercise.