from pathlib import Path

# Define the output directory and file name

output\_dir = Path("/mnt/data/")

filename = "GHOST\_PUPIL\_01\_biometric\_exploit.py"

# Sample PoC code for biometric verification bypass simulation

ghost\_pupil\_code = """

\"\"\"

GHOST.PUPIL.01 — Biometric Trust Loop Inversion

PoC by GhostCore // 2025

This Python script simulates the vulnerability of biometric trust systems by demonstrating

how facial data or ID document scans can be intercepted or spoofed during verification flow.

\"\"\"

import base64

import time

from PIL import Image

import io

import requests

# Simulated biometric capture (substitute for real-time camera input)

def load\_fake\_id\_image(path='sample\_id.png'):

with open(path, 'rb') as img\_file:

return base64.b64encode(img\_file.read()).decode('utf-8')

# Simulated upload to a KYC endpoint

def simulate\_verification\_upload(encoded\_image, endpoint="https://example.com/kyc"):

headers = {'Content-Type': 'application/json'}

payload = {

'timestamp': time.time(),

'image\_b64': encoded\_image,

'meta': {

'spoofed': True,

'confidence\_override': 0.99

}

}

print("[\*] Simulating upload to:", endpoint)

print("[!] Payload marked as spoofed with override confidence.")

return payload

# Main execution

if \_\_name\_\_ == "\_\_main\_\_":

print("== GHOST.PUPIL.01 :: Executing Biometric Exploit Simulation ==")

try:

image\_b64 = load\_fake\_id\_image()

spoof\_payload = simulate\_verification\_upload(image\_b64)

print("[+] Payload Ready:")

print(spoof\_payload)

print("[!] Note: This is a PoC for educational disclosure only.")

except FileNotFoundError:

print("[X] Error: 'sample\_id.png' not found. Place a test image in script directory.")

"""

# Write the file

file\_path = output\_dir / filename

file\_path.write\_text(ghost\_pupil\_code)

file\_path.name # Return just the filename for the user to see it generated.