

```

import os
import random
import time
from datetime import datetime, timedelta
from faker import Faker

fake = Faker()
folder_path = "C:\\\\Users\\\\ALEKSA\\\\Desktop\\\\haunted_folder"
log_file = os.path.join(folder_path, "warnings.log")

CRYPTIC_MESSAGES = [
    "I see you.",
    "It won't stop.",
    "Do not open this file.",
    "She's still here.",
    "He knows your name.",
    "You changed the folder, the folder changed you.",
    "This is your last warning.",
    "Don't look in the mirror.",
]

def setup_folder():
    if not os.path.exists(folder_path):
        os.makedirs(folder_path)

def log_warning(message):
    cwd
    with open(log_file, "a", encoding="utf-8" ) as f:
        f.write(f"[{datetime.now()}] WARNING: {message}\n")

def create_random_file():
    message = random.choice(CRYPTIC_MESSAGES)
    filename = f"note_{random.randint(1000,9999)}.txt"
    file_path = os.path.join(folder_path, filename)
    with open(file_path, "w") as f:
        f.write(message)
    log_warning(f"A new file might appear: '{filename}' whispers: \"{message}\"")

def randomize_timestamps():
    files = [f for f in os.listdir(folder_path) if f.endswith(".txt") and f !=
"warnings.log"]
    if not files:
        return
    file_to_change = random.choice(files)

```

```

file_path = os.path.join(folder_path, file_to_change)

back_date = datetime.now() - timedelta(days=random.randint(1, 666),
hours=random.randint(0, 23))
mod_time = time.mktime(back_date.timetuple())
os.utime(file_path, (mod_time, mod_time))
log_warning(f"The past is bleeding into the present. '{file_to_change}' was
touched by time itself.")

def rename_based_on_content():
    files = [f for f in os.listdir(folder_path) if f.endswith(".txt") and f !=
"warnings.log"]
    if not files:
        return
    file_to_rename = random.choice(files)
    path = os.path.join(folder_path, file_to_rename)
    with open(path, "r") as f:
        content = f.read().strip()
    if content:
        new_name = f"{content[:10].replace(' ', '_').lower()}.txt"
        new_path = os.path.join(folder_path, new_name)
        os.rename(path, new_path)
        log_warning(f"The message has consumed its identity: '{file_to_rename}' →
'{new_name}'")

def haunt():
    setup_folder()
    print(" The haunting has begun... Press Ctrl+C to escape.")
    try:
        while True:
            event = random.choice(["create", "timestamp", "rename"])
            if event == "create":
                create_random_file()
            elif event == "timestamp":
                randomize_timestamps()
            elif event == "rename":
                rename_based_on_content()
            time.sleep(random.randint(4, 10))
    except KeyboardInterrupt:
        print("\n The folder has gone quiet... for now.")

if __name__ == "__main__":
    haunt()

```

