""" Capstone Project: Malware Detector Author: Chukwuebuka Tobiloba Nwaizugbe Email: nwaizugbechukwuebuka@gmail.com LinkedIn: https://linkedin.com/in/chukwuebuka-tobiloba-nwaizugbe

Description: This script simulates a basic malware detection system using:

- File hashing (SHA256)
- VirusTotal API for threat intelligence
- Simulated Snort rule validation """

import os import hashlib import requests import time

---------------- Configuration ----------------

VIRUSTOTAL_API_KEY = "YOUR_API_KEY_HERE" # Replace with your actual API key if not VIRUSTOTAL_API_KEY or VIRUSTOTAL_API_KEY == "YOUR_API_KEY_HERE": raise ValueError("Please set a valid VirusTotal API key in VIRUSTOTAL_API_KEY variable.")

HEADERS = { "x-apikey": VIRUSTOTAL_API_KEY }

---------------- Hashing Function ----------------

def calculate_sha256(file_path): sha256 = hashlib.sha256() with open(file_path, "rb") as f: for byte_block in iter(lambda: f.read(4096), b""): sha256.update(byte_block) return sha256.hexdigest()

---------------- VirusTotal Lookup ----------------

def check_virustotal(file_hash): url = f"https://www.virustotal.com/api/v3/files/{file_hash}" try: response = requests.get(url, headers=HEADERS) if response.status_code == 200: data = response.json() malicious_count = data['data']['attributes']['last_analysis_stats']['malicious'] return malicious_count elif response.status_code == 404: return "Not found in VirusTotal" else: return f"Error: {response.status_code}" except requests.exceptions.RequestException as e: return f"Request failed: {str(e)}"

---------------- Simulated Snort Rule Check ----------------

def snort_simulation(file_path): """ Dummy function to simulate a Snort rule check. Real-world Snort integration would analyze packet capture (pcap) files or live traffic. """ print("[Simulated Snort] Checking for suspicious file patterns...") # Simulate threat detection if "malware" in file_path.lower(): return True return False

---------------- Main Detection Function ----------------

def scan_file(file_path): print(f"\nScanning file: {file_path}")

```
if not os.path.exists(file_path):
    print("Error: File not found.")
    return

sha256_hash = calculate_sha256(file_path)
print(f"SHA-256 Hash: {sha256_hash}")

snort_result = snort_simulation(file_path)
vt_result = check_virustotal(sha256_hash)

# Report Summary
print("\n--- Malware Scan Summary ---")
print(f"File Path     : {file_path}")
print(f"Snort Alert   : {'YES' if snort_result else 'NO'}")
print(f"VirusTotal    : {vt_result}")
print("-----------------------------")
```

----------------- Execution -----------------

if **name** == "**main**": # Example usage: place sample files in the same directory files_to_scan = ["example_file.exe", "test_malware_sample.zip"]

```
for file in files_to_scan:
    scan_file(file)
    time.sleep(15)  # Respect VirusTotal API rate limits
```