""" test_detector.py

Unit tests for detector.py in the Cybersecurity Capstone Project. Tests include alert parsing, VirusTotal API interaction (mocked), and logging. """

import os import unittest from unittest.mock import patch, MagicMock from src import detector # Ensure your project structure has src/detector.py

class DetectorTests(unittest.TestCase):

```
def setUp(self):
    # Setup: create a sample Snort alert file for testing
    self.test_alert_file = "logs/test_snort_alerts.txt"
    os.makedirs(os.path.dirname(self.test_alert_file), exist_ok=True)
    with open(self.test_alert_file, "w") as f:
        f.write("[**] [1:123456:1] ET MALWARE Test Alert [**]\n")
        f.write("04/11-14:22:30.123456 192.168.1.20 ->
198.51.100.15\n")

def tearDown(self):
    # Cleanup test alert file
    if os.path.exists(self.test_alert_file):
        os.remove(self.test_alert_file)
    # Cleanup log file created by detector.log_detection
    if os.path.exists(detector.LOG_OUTPUT):
        os.remove(detector.LOG_OUTPUT)

def test_read_snort_alerts(self):
    alerts = detector.read_snort_alerts(self.test_alert_file)
    self.assertTrue(len(alerts) > 0)
    self.assertIn("ET MALWARE", alerts[0])

@patch('src.detector.requests.get')
def test_check_hash_virustotal_success(self, mock_get):
    # Mock a successful VirusTotal API response
    mock_response = MagicMock()
    mock_response.status_code = 200
    mock_response.json.return_value = {
        "data": {
            "attributes": {
                "last_analysis_stats": {"malicious": 5, "harmless":
70}
            }
        }
    }
    mock_get.return_value = mock_response

    result = detector.check_hash_virustotal("dummyhash")
    self.assertIsInstance(result, dict)
    self.assertIn("malicious", result)

@patch('src.detector.requests.get')
def test_check_hash_virustotal_not_found(self, mock_get):
    # Mock a 404 VirusTotal API response (hash not found)
    mock_response = MagicMock()
    mock_response.status_code = 404
    mock_get.return_value = mock_response

    result = detector.check_hash_virustotal("dum
```