Ransomware Defender v2 - Documentation

Overview

Ransomware Defender v2 is a real-time monitoring and response tool designed to detect and mitigate ransomware-like activity. It operates by identifying suspicious patterns such as rapid file modifications and high-entropy content, attempting to quarantine affected files and terminate malicious processes.

Platform & Requirements

• OS: Windows (tested)

• Python: 3.8 or later

• Dependencies:

- watchdog>=2.1.9

- psutil>=5.9.0

• GUI Toolkit: Tkinter

Ensure the system has permissions for file monitoring and process inspection.

Installation

Before using the application, create a virtual environment:

```
python -m venv venv

./venv/Scripts/activate  (Windows)
```

Then install required packages:

```
pip install -r requirements.txt
```

Running the Application

To start the GUI:

```
python main.py
```

Typical use:

1. Launch the application

2. Add directories to monitor

3. Adjust detection configuration if needed

4. (Optional) Enable auto-quarantine

5. Press START to begin monitoring

6. Use SCAN for manual checks

7. Review logs and quarantine results as needed

Key Features

• Real-time monitoring of directories

• Detection based on file modification frequency and entropy analysis

• Composite threat scoring for response decisions

• Optional auto-quarantine of affected files

• Background unsafe process analysis and termination attempts

• GUI-based control panel with logs and indicators

• Export logs as ZIP

• Structured JSON event logs

• Safe copies of recent files created during threat response

Project Structure

main.py - GUI entry point

gui.py - Tkinter-based interface

monitor.py - Core file monitoring logic

detector.py - Entropy and scoring functions

quarantine.py - Move/restore quarantine operations

logger.py - Rotating logs and structured event tracking

utils.py - Helper functions

ransom_attack.py - Simple testing script

logs/ - rdefender.log, events.jsonl, recovery_log.json, safeguards/*

Detection Logic

• Tracks file create/modify/move events over a sliding time window

• Samples up to 4KB of each modified file and computes Shannon entropy

• High-entropy events and high modification frequency contribute to scoring

• Threat triggers when score exceeds configured threshold or heuristic detects a wave

Responses to Detection

• Files may be automatically quarantined (moved or copied)

• Event results logged to recovery_log.json

• Attempts made to identify and stop potentially malicious processes

• Creates safeguard copies of recently affected files for restoration

GUI Details

Controls include:

• Add Path

• Remove Path

• START / STOP monitoring

• SCAN (manual)

• View and Restore quarantine entries

• Export logs as ZIP

• Quit

A status bar provides activity state and watch counts, while logs display significant events.

Configuration Parameters

window_seconds: 10

check_interval: 3

modified_threshold: 12

entropy_threshold: 7.5

high_entropy_count: 6

sample_entropy_count: 20

process_suspicion_score: 5

quarantine_dir: ./quarantine

auto_quarantine: False

detection_score_threshold: 60

Restore and Recovery

GUI provides restore option to recover quarantined files.

Alternatively, restore.py can be used from CLI to recover files listed in recovery_log.json.

Security Notes

• Do not run on production data without testing

• Quarantine operations require proper permissions

• Tool avoids terminating low-PID critical Windows processes