**Ransomware detection and prevention application**

The **Ransomware Detection and Prevention System** is a comprehensive Python-based application designed to detect ransomware activities using cuckoo sandbox-based features like behavioral analysis and file system monitoring, network traffic analysis registry monitoring, process monitoring, API calls analysis, static analysis, and Report Generation. Additionally, it employs Moving Target Defense (MTD) as a prevention mechanism to secure data. The system includes an interactive user interface (UI) built with Tkinter, providing real-time results, graphical representations, and severity analysis of detected ransomware. A report generation feature is also incorporated to summarize the findings.

**Features**

**1. Ransomware Detection**

* Uses multiple detection techniques, same used by cuckoo sandbox including:
  + Behavioral Analysis
  + File System Monitoring
  + Network Traffic Analysis
  + Registry Monitoring
  + Process Monitoring
  + API Calls Analysis
  + Static Analysis
* Each technique analyzes specific aspects of system behavior to detect potential ransomware activity.
* The detection results are displayed on the UI with graphical representation and color-coded results (red for detected, green for not detected).

**2. Severity Analysis**

* After detection, the application determines the severity of the ransomware:
  + **Severe**: High potential harm (red).
  + **Mild**: Medium potential harm (yellow).
  + **Normal**: Low threat (green).

**3. Prevention Mechanism (MTD)**

* Implements **Moving Target Defense (MTD)** by rotating sensitive files to secure locations.
* Displays the source and destination of file movement during prevention.
* The prevention status is shown in the UI, along with the file paths.

**4. Graphical Representation**

* The results of the detection process are visualized using Matplotlib bar charts embedded in the UI.
* Displays a bar chart with ransomware components (e.g., Behavioral Analysis) indicating whether ransomware was detected or not.

**5. Report Generation**

* Generates detailed and summary reports of the detection and prevention processes.
* Reports are saved as .txt files and can be accessed directly from the application.
* Includes a summary report with ransomware detection results and prevention actions.

**Technical Specifications**

* **Language**: Python
* **UI Framework**: Tkinter
* **Graphical Plotting**: Matplotlib
* **Logging**: Python logging module
* **File Handling**: shutil and os modules
* **System Monitoring**: subprocess module

**Installation and Setup**

1. **Prerequisites:**
   * Python 3.x
   * Required Python libraries: Tkinter, Matplotlib, PIL (Pillow), subprocess, shutil, os, logging
2. **Installation Steps:**
   * Clone the project repository.
   * Install the necessary libraries using pip:

bash

Copy code

pip install matplotlib pillow

1. **Running the Application:**
   * Navigate to the project folder.
   * Run the main.py file:

bash

Copy code

python main.py

1. **Usage:**
   * The UI will load, providing options for starting the detection and prevention processes.
   * The detection section will evaluate the system and provide graphical results.
   * The prevention section will secure files using MTD and display file movement information.
   * Reports can be generated at the end of the detection/prevention processes.

**Project Structure**

|-- detection/

| |-- behavioral\_analysis.py

| |-- file\_system\_monitoring.py

| |-- network\_traffic\_analysis.py

| |-- registry\_monitoring.py

| |-- process\_monitoring.py

| |-- api\_calls\_analysis.py

| |-- static\_analysis.py

|-- ransomware\_model.py

|-- ransomware\_data.csv

|

|-- prevention/

| |-- moving\_target\_defense.py

|-- prevention\_model.py

|-- ransomware\_prevention\_data.csv

|

|-- reports/

| |-- detection\_and\_prevention\_report.txt

| |-- generate\_report.py

|

|-- main.py

|-- welcome\_page.py

|-- bg.jpeg (background image)

**Core Files**

1. **main.py**:
   * The main application file that initializes the Tkinter window and ties together the detection, prevention, and report generation processes.
2. **detection/**:
   * Contains individual modules for various detection techniques (Behavioral Analysis, File System Monitoring, etc.).
3. **prevention/moving\_target\_defense.py**:
   * Implements Moving Target Defense (MTD) to secure important files.
4. **reports/**:
   * Contains both detailed and summary reports generated after the detection and prevention processes.