The Cyberattack Lifecycle: Threat Intelligence, Digital Forensics, and Incident Response

CSC G0040

Fall Semester, 2024

*EndoGuard: A Forensic Monitoring and Data Collection too*l

M M Hasan Tajwar

Grove School of Engineering

The City College of New York
mtajwar000@citymail.cuny.edu

EndoGuard: A Forensic Monitoring and Data Collection tool

Abstract

EndoGuard is an innovative digital forensic application designed to facilitate the real-time monitoring of system activities, enabling rapid detection and analysis of potential security breaches. By incorporating modular components for file monitoring, network traffic analysis, and system log collection, EndoGuard provides a robust solution for incident response and forensic investigations. Its architecture emphasizes scalability, usability, and reliability, making it a valuable asset for cybersecurity professionals.

Introduction

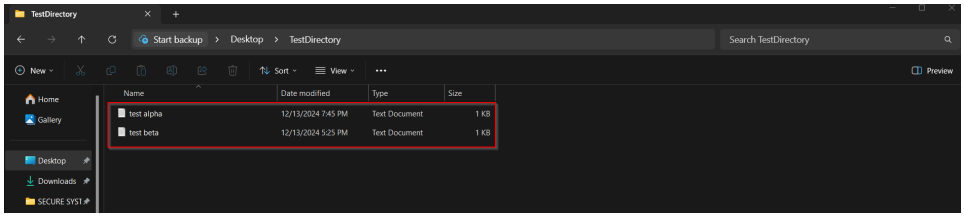
The field of digital forensics has become increasingly vital in identifying and mitigating cyber threats. EndoGuard addresses this need by offering a unified platform for tracking file system changes, monitoring network activities, and collecting forensic data. The application leverages event-driven programming and database integration to deliver high performance and ensure persistent data storage. This report details the system's architecture, features, and implementation, along with a discussion of potential challenges, future enhancements, and ethical considerations.

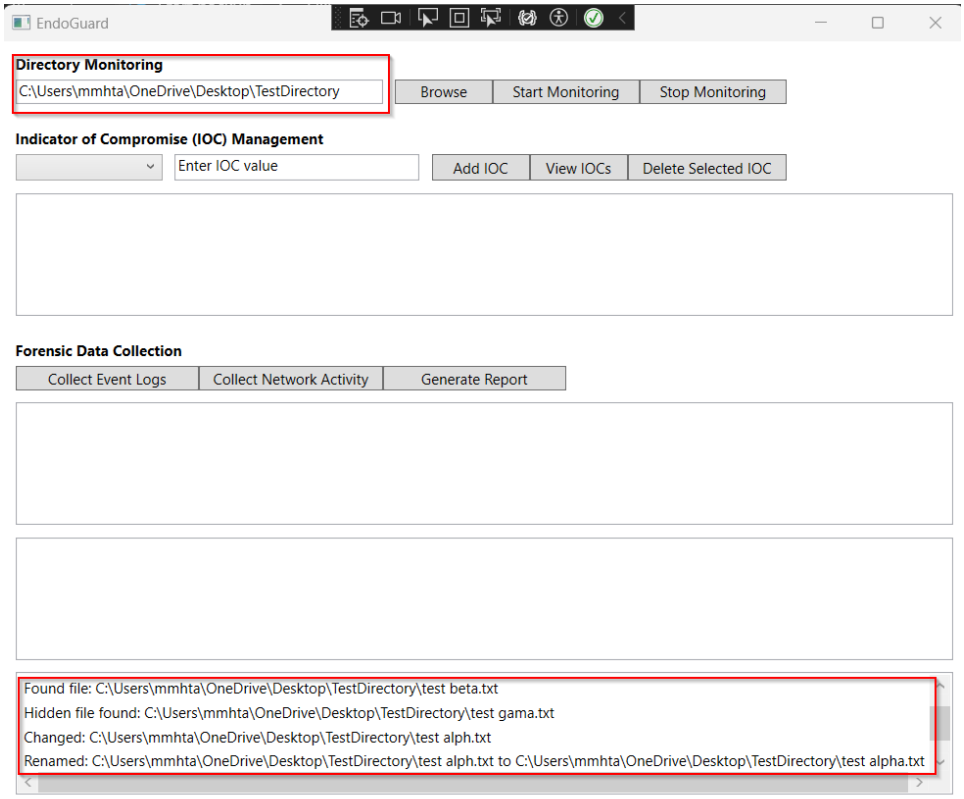
System Architecture

The architecture of EndoGuard adheres to the principles of modularity and separation of concerns. Each major functionality is encapsulated in its own module, facilitating maintainability and scalability. The key components of the system are:

1. File Monitoring

- **Implementation**: Managed by `FileMonitor.cs`, which utilizes `FileSystemWatcher` to track file and directory changes.
- **Capabilities**:
  - Detects modifications, deletions, and renaming of files.
  - Logs file activities in real-time and updates the database for persistence.
  - User interface integration for visualizing file operations in the activity log.



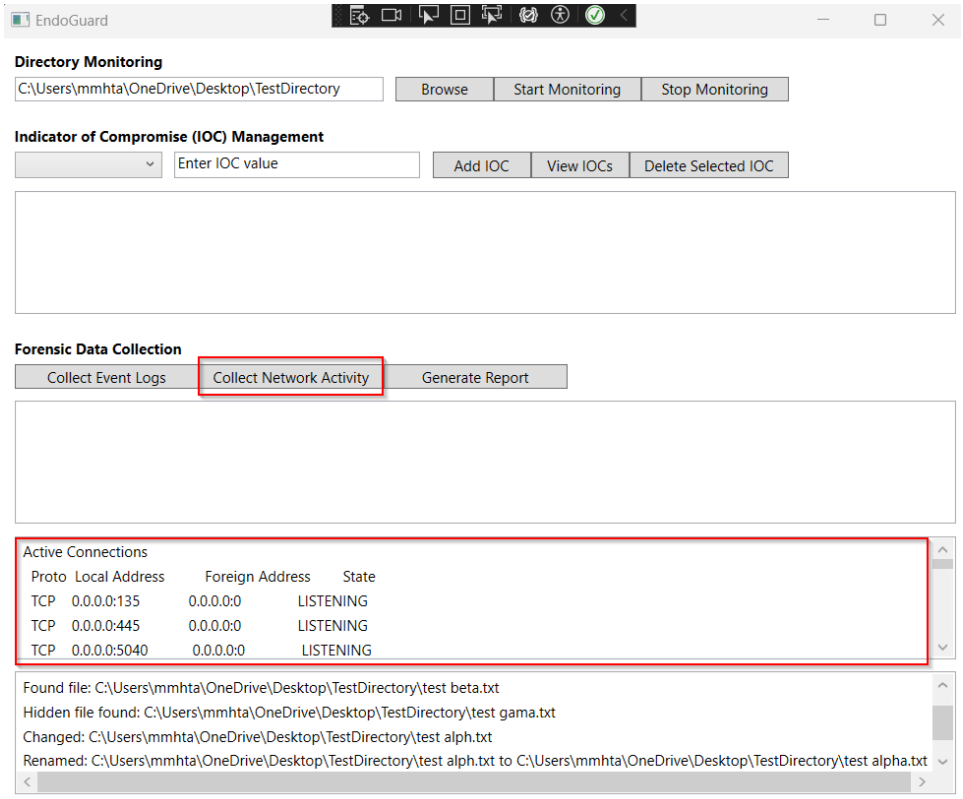


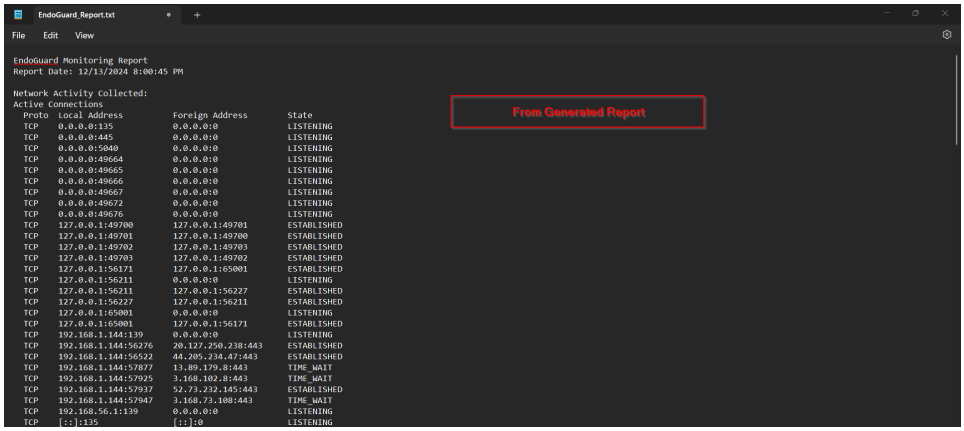
- **Design Choices**:

- Event-driven architecture ensures efficiency by processing events only when changes occur.
- Granular monitoring options with filters for specific file attributes.

2. Network Monitoring

- **Implementation**: Defined in `NetworkMonitor.cs`, the module captures live network activity.
- **Capabilities**:
  - Monitors active connections and logs them into the database.
  - Supports asynchronous operations to maintain application responsiveness.

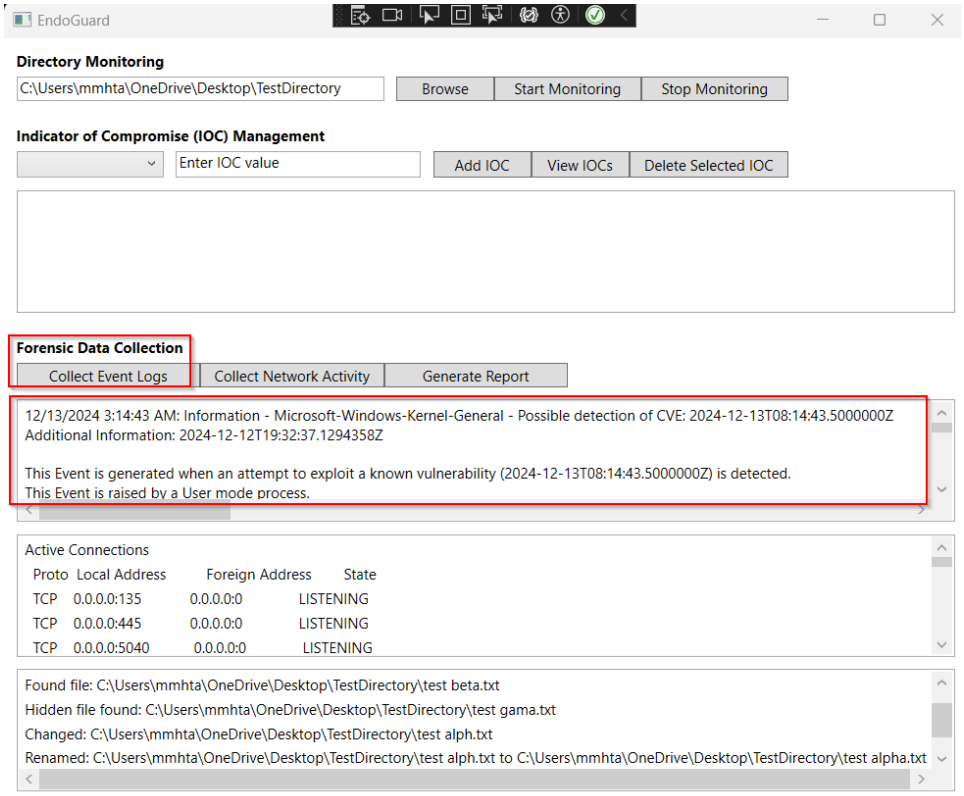


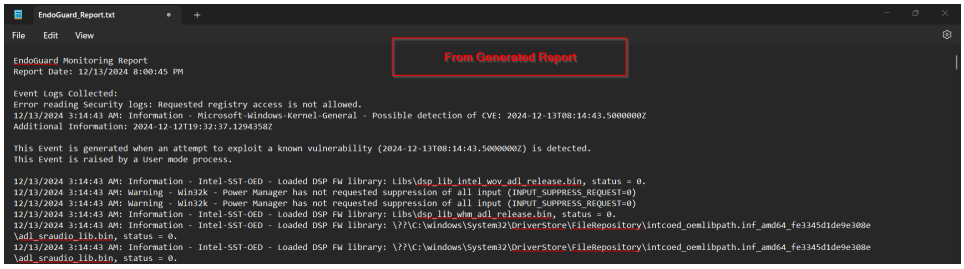


- **Design Choices**:
  - Use of system-level APIs to retrieve real-time network information.
  - Integration with the database for long-term storage and analysis of suspicious connections.

3. Forensic Data Collection

- **Implementation**: Handled by `ForensicDataCollector.cs`.
- **Capabilities**:
  - Collects system event logs from Security, System, and Application categories.
  - Filters logs based on time to focus on recent and relevant events.
  - Prepares detailed records for forensic investigations.

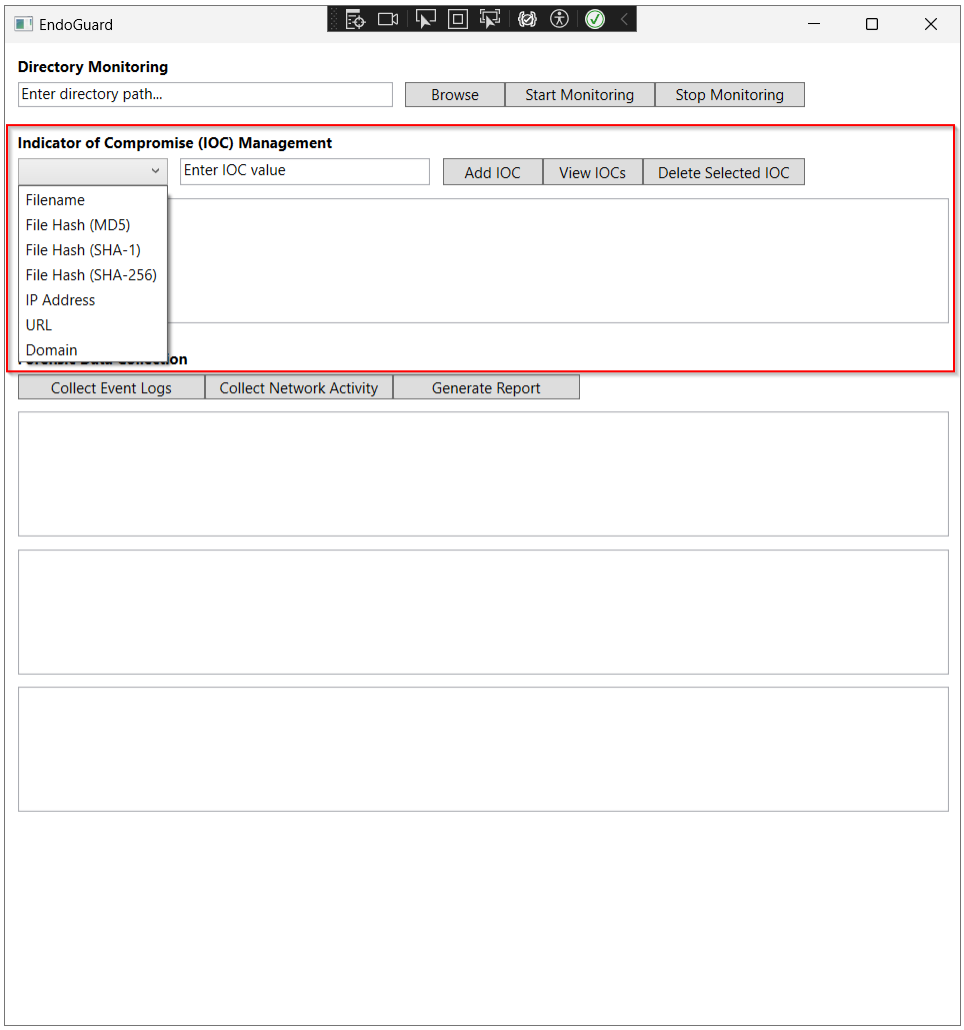




- **Design Choices**:
  - Utilizes built-in Windows APIs to extract event logs efficiently.
  - Modular function design to accommodate future expansion (e.g., additional log sources).

4. Database Integration

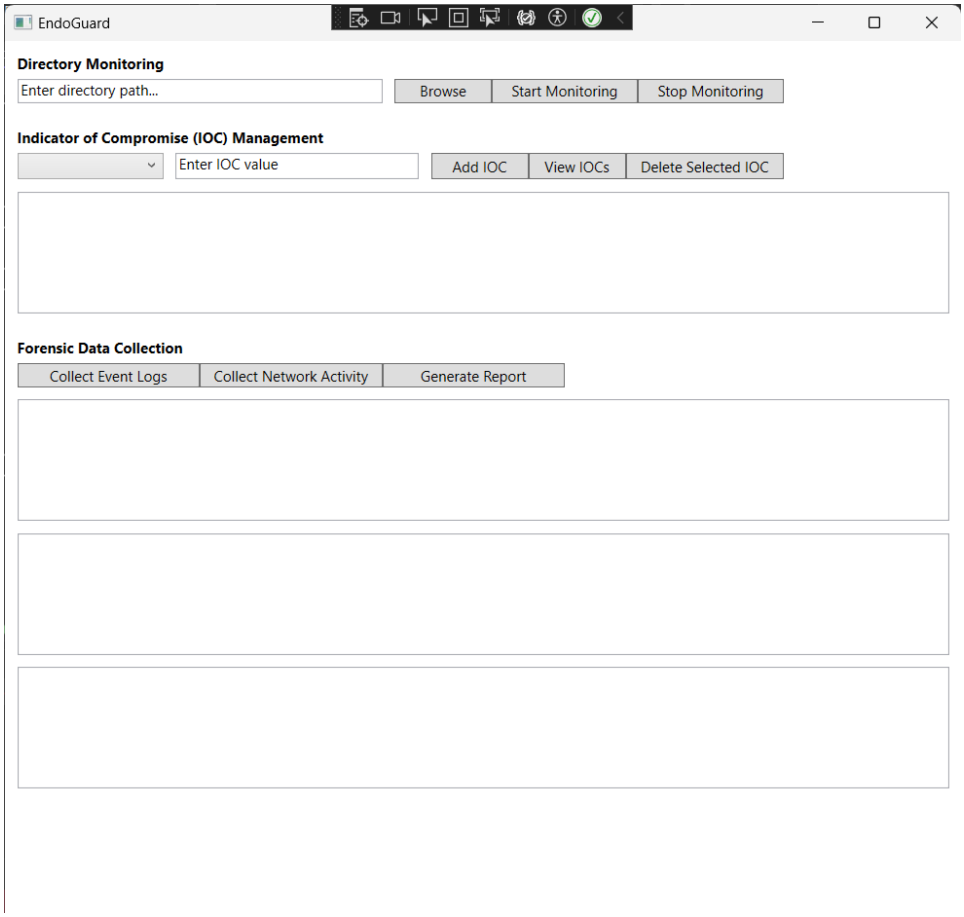
- **Implementation**: Managed by `DatabaseHelper.cs`, leveraging SQLite as the database backend.
- **Capabilities**:
  - Stores Indicators of Compromise (IOCs), network activity logs, and forensic records.
  - Supports CRUD operations for seamless data management.



- **Design Choices**:
  - SQLite chosen for its lightweight and embedded nature, eliminating the need for external database dependencies.
  - Tables for IOCs and NetworkActivities ensure data is well-structured and queryable.

5. User Interface

- **Implementation**: `MainWindow.xaml` and `MainWindow.xaml.cs` define the front-end interface.
- **Capabilities**:
  - Real-time visualization of file and network activities.
  - Interactive controls for initiating monitoring tasks and viewing logs.
  - Simple, intuitive layout suitable for both novice and expert users.



- **Design Choices**:
  - Built with WPF (Windows Presentation Foundation) for a modern, responsive interface.
  - Integration with back-end modules for real-time data updates.

Key Features

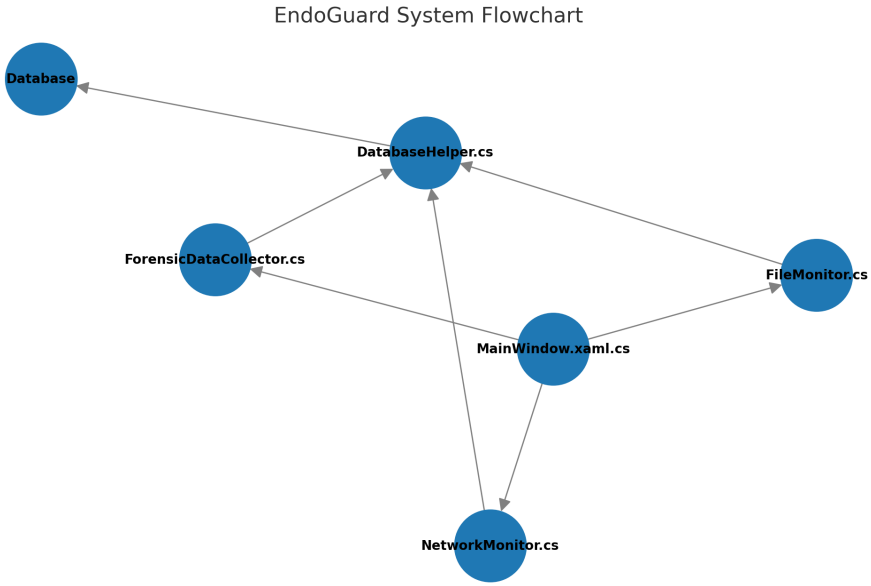
1. **Real-Time Monitoring**:
   - File system changes and network activities are tracked in real-time.
   - Data is logged immediately to ensure minimal data loss.

2. **Persistent Storage**:
    - All monitored data is stored in a structured database, ensuring availability for historical analysis.
3. **Event Filtering**:
    - FileMonitor and ForensicDataCollector use filters to prioritize relevant events, improving performance and accuracy.
4. **Extensibility**:
    - Modular design allows the addition of new features, such as machine learning-based anomaly detection or advanced visualization tools.

Potential Use Cases

1. **Incident Response**:
    - Quickly identify unauthorized changes to critical files or suspicious network activity during an incident.
2. **Compliance Monitoring**:
    - Ensure systems adhere to security and operational policies by tracking key events and activities.
3. **Forensic Investigations**:
    - Provide a detailed timeline of file changes and network interactions to support post-incident analysis.

System Flow Explanation



The flowchart above illustrates the interaction between key components in the EndoGuard system. Here's a detailed explanation of the flow:

1. **MainWindow.xaml.cs**:
   - Acts as the entry point, initializing and managing the core modules: `FileMonitor`, `NetworkMonitor`, and `ForensicDataCollector`.
   - Orchestrates the interaction between the UI and the underlying monitoring modules.
2. **FileMonitor.cs**:
   - Tracks file system activities and logs relevant events.
   - Interacts with `DatabaseHelper` to persistently store file activity logs.
3. **NetworkMonitor.cs**:
   - Monitors network connections and captures activity.
   - Similar to `FileMonitor`, it passes collected data to `DatabaseHelper` for storage.

4. **ForensicDataCollector.cs**:
    - Collects system event logs for forensic analysis.
    - Works in conjunction with `DatabaseHelper` to store structured forensic data.
5. **DatabaseHelper.cs**:
    - Serves as the bridge between the application modules and the SQLite database.
    - Provides a consistent interface for inserting, querying, and managing forensic data.
6. **Database**:
    - Centralized storage for all monitored data, including file activities, network logs, and forensic records.

Implementation Challenges

1. **Real-Time Monitoring**:
    - **Challenge**: Ensuring the application processes events in real-time without performance bottlenecks.
    - **Solution**: Use of asynchronous programming and event-driven architecture.
2. **Data Integrity**:
    - **Challenge**: Maintaining the accuracy and consistency of stored forensic data.
    - **Solution**: Implement transactional operations in the database to avoid corruption during high-volume inserts.
3. **Scalability**:
    - **Challenge**: Adapting to increasing data volumes and potential integration with enterprise systems.
    - **Solution**: Modular design ensures scalability, but future iterations may require database optimization or migration to a distributed database.

4. **Cross-Platform Challenges**:
    - **Challenge**: Adapting the file and network monitoring features to non-Windows systems.
    - **Solution**: Research alternative APIs and libraries for Linux and macOS.
5. **Security Risks**:
    - **Challenge**: Protecting the application and database from tampering or unauthorized access.
    - **Solution**: Encrypt sensitive data and implement role-based access controls for the application.

Future Roadmap

Phase 1: Immediate Enhancements

- **Data Encryption**: Implement encryption for both stored data and database connections.
- **Notification System**: Add support for real-time alerts based on user-defined triggers.

Phase 2: Feature Expansion

- **Cloud Integration**: Provide an option to store logs and forensic data in secure cloud storage for redundancy.
- **Advanced Analytics**: Integrate machine learning models to detect anomalous patterns in file and network activities.

Phase 3: Platform Compatibility

- **Cross-Platform Support**: Extend monitoring capabilities to Linux and macOS using platform-specific APIs.
- **Web Interface**: Develop a web-based dashboard for remote access and real-time monitoring.

Phase 4: Enterprise Deployment

- **Distributed Architecture**: Migrate to a distributed database to handle large-scale data across multiple systems.
- **Role-Based Access**: Add multi-user support with role-based permissions for secure enterprise usage.

Strengths

- **Efficiency**: Event-driven monitoring reduces resource overhead.
- **User-Focused Design**: Simplified UI with comprehensive monitoring options.
- **Scalability**: Modular components can be expanded to support additional forensic features.

Recommendations for Future Improvements

1. **Encryption**:
    - Encrypt stored data to ensure confidentiality and protect against unauthorized access.
2. **Advanced Analytics**:
    - Introduce AI/ML models to detect patterns or anomalies in monitored data.
3. **Cross-Platform Support**:
    - Extend monitoring capabilities to Linux and macOS environments.
4. **Notification System**:
    - Implement real-time alerts for specific events (e.g., access to critical files).
5. **Cloud Integration**:
    - Provide options to store forensic data in the cloud for enhanced accessibility and disaster recovery.

Conclusion

EndoGuard is a well-designed digital forensic application that effectively combines real-time monitoring, persistent data storage, and user-friendly interfaces. Its modular architecture and focus on scalability ensure that it can adapt to evolving cybersecurity challenges. By addressing ethical considerations and implementing the proposed enhancements, EndoGuard can establish itself as a leading tool in digital forensics.