**Министерство науки и высшего образования Российской Федерации**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Технологии и методы программирования»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

**Выполнили:**

Ахраров Али, студент группы номер N3350

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Проверил:**

Ищенко Алексей Петрович

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(отметка о выполнении)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Введение**

В современных условиях информационной безопасности защита файлов и данных на локальных системах приобретает все большую актуальность. Незаконное создание, копирование, удаление или переименование файлов может привести к серьезным последствиям, включая потерю данных, нарушение работы приложений и компрометацию конфиденциальной информации. В рамках данного проекта была разработана программа, обеспечивающая защиту файлов в текущем каталоге от несанкционированных операций, а также предотвращающая доступ к конфигурационному файлу template.tbl, содержащему список запрещенных имен файлов и пароля для управления режимом защиты.

**Цель и задачи проекта**

**Цель проекта** заключается в создании программного обеспечения, которое предотвращает выполнение определенных операций с файлами в указанном каталоге, а также защищает конфигурационный файл от несанкционированного доступа и изменений.

**Задачи проекта:**

1. Разработка механизма отслеживания и блокировки операций создания, копирования, удаления и переименования файлов с заданными именами или соответствующих маскам.
2. Организация хранения списка запрещенных имен файлов в защищенном конфигурационном файле template.tbl.
3. Обеспечение защиты конфигурационного файла от удаления, просмотра и модификации.
4. Реализация возможности включения и отключения режима защиты с использованием пароля, хранящегося в зашифрованном виде.
5. Предусмотреть возможность отключения программы с использованием пароля, введенного пользователем.
6. Обеспечение совместимости программы с операционными системами семейства Microsoft Windows, а также возможная адаптация для Linux.

**Анализ требований**

Для выполнения поставленных задач программа должна обеспечивать:

* Мониторинг текущего каталога на предмет выполнения запрещенных операций с файлами.
* Чтение и обработку списка запрещенных имен или масок из файла template.tbl.
* Защиту файла template.tbl от несанкционированных действий путем установки соответствующих прав доступа.
* Реализацию механизма аутентификации при включении или отключении режима защиты с использованием пароля.
* Использование API функций операционной системы для управления безопасностью файлов и каталогов.

**Архитектура программы**

Программа состоит из нескольких основных компонентов:

1. **Модуль мониторинга файловой системы:** Отвечает за отслеживание событий создания, копирования, удаления и переименования файлов в текущем каталоге.
2. **Модуль управления списком запрещенных файлов:** Обеспечивает чтение, запись и обработку списка запрещенных имен и масок из файла template.tbl.
3. **Модуль защиты конфигурационного файла:** Устанавливает и контролирует права доступа к файлу template.tbl, предотвращая его удаление, просмотр и модификацию.
4. **Модуль аутентификации:** Реализует механизм проверки пароля для включения и отключения режима защиты. Пароль хранится в первой строке файла template.tbl в хешированном виде.
5. **Пользовательский интерфейс:** Предоставляет пользователю возможность управлять режимом защиты, вводить пароль и просматривать статус программы.

**Выбор технологий и инструментов**

Для реализации программы был выбран язык программирования C++ благодаря его высокой производительности и обширной поддержке системных API. В качестве среды разработки использовался Microsoft Visual Studio, что обеспечило удобство интеграции с API Windows, такими как SetFileSecurity и SetNamedSecurityInfo, рекомендованными в задании.

**Реализация**

**1. Модуль мониторинга файловой системы**

Для отслеживания изменений в файловой системе был использован API Windows, позволяющий отслеживать события файлового менеджера в реальном времени. Основные шаги реализации:

* **Инициализация наблюдателя:** Использование функций ReadDirectoryChangesW для мониторинга текущего каталога на предмет изменений.
* **Обработка событий:** Реализация обратных вызовов, реагирующих на создание, копирование, удаление и переименование файлов.
* **Сравнение с запрещенными именами:** При возникновении события проверяется, соответствует ли имя файла запрещенному имени или маске из списка template.tbl.
* **Блокировка операции:** В случае совпадения выполняется отмена операции и уведомление пользователя.

**2. Модуль управления списком запрещенных файлов**

Файл template.tbl содержит список запрещенных имен файлов или масок, а также хешированный пароль. Реализация данного модуля включает:

* **Чтение файла конфигурации:** При запуске программы происходит загрузка списка запрещенных имен и масок.
* **Парсинг данных:** Обработка строк файла для извлечения масок и пароля.
* **Обновление списка:** Предусмотрены механизмы добавления или удаления записей с соответствующими правами доступа.

**3. Модуль защиты конфигурационного файла**

Для защиты файла template.tbl от несанкционированного доступа используются функции SetFileSecurity и SetNamedSecurityInfo. Основные этапы:

* **Установка прав доступа:** Ограничение прав на чтение, запись и удаление файла только для администратора или определенных пользователей.
* **Обеспечение неизменности:** Предотвращение изменения атрибутов файла, таких как скрытость или системный статус.
* **Мониторинг изменений:** Дополнительный контроль за попытками изменения прав доступа к файлу.

**4. Модуль аутентификации**

Механизм аутентификации реализован следующим образом:

* **Хеширование пароля:** Пароль для управления режимом защиты хранится в первой строке файла template.tbl в зашифрованном виде с использованием криптографических хеш-функций (например, SHA-256).
* **Проверка пароля:** При попытке включения или отключения защиты пользователь вводит пароль, который хешируется и сравнивается с хранимым значением.
* **Безопасность хранения:** Хешированный пароль невозможно восстановить в исходном виде, что обеспечивает безопасность хранения.

**5. Пользовательский интерфейс**

Интерфейс программы предоставляет следующие возможности:

* **Включение/отключение режима защиты:** Кнопки или команды для управления защитой с запросом пароля.
* **Отображение статуса:** Информация о текущем состоянии защиты и количестве запрещенных операций.
* **Логирование событий:** Журналирование попыток выполнения запрещенных действий для последующего анализа.

**Обеспечение безопасности**

Безопасность программы достигается путем:

* **Ограничения прав доступа:** Конфигурационный файл защищен от несанкционированного доступа с использованием системных функций безопасности.
* **Хеширования пароля:** Пароль хранится в зашифрованном виде, что предотвращает его компрометацию.
* **Мониторинга действий:** Все попытки выполнения запрещенных операций регистрируются для обеспечения контроля и аудита.
* **Изоляции процессов:** Программа работает с повышенными привилегиями, необходимыми для контроля над файловой системой, но при этом минимизирует потенциальные уязвимости.

**Тестирование**

Для обеспечения корректной работы программы было проведено комплексное тестирование, включающее:

* **Функциональные тесты:** Проверка блокировки операций создания, копирования, удаления и переименования файлов с запрещенными именами и масками.
* **Тесты безопасности:** Проверка защиты конфигурационного файла от удаления, просмотра и модификации.
* **Тесты аутентификации:** Валидация механизма проверки пароля и корректности включения/отключения режима защиты.
* **Нагрузочные тесты:** Оценка производительности программы при интенсивной работе с файловой системой.

**Результаты**

Разработанная программа успешно выполняет поставленные задачи, обеспечивая надежную защиту файлов в текущем каталоге от несанкционированных операций. Конфигурационный файл template.tbl надежно защищен от доступа и изменений, а механизм аутентификации позволяет управлять режимом защиты безопасно и эффективно. Проведенное тестирование подтвердило стабильность и эффективность работы программы в различных условиях эксплуатации.

**Заключение**

Проект по разработке программы защиты файлов в каталоге достиг своей цели, предоставив инструмент для предотвращения несанкционированных операций с файлами. Реализованные механизмы безопасности и удобный интерфейс делают программу полезным средством для обеспечения информационной безопасности на локальных системах. В будущем возможны расширение функционала, включая поддержку дополнительных операционных систем и улучшение пользовательского интерфейса.

**Приложения**

**Пример файла template.tbl**

e3afed0047b08059d0fada10f400c1e5

\*.exe

\*.dll

important\_document.txt

*Первая строка содержит хешированный пароль, далее — список запрещенных масок и имен файлов.*