

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

**“Большие данные и распределенная цифровая платформа”**

**ОТЧЕТ**

**по лабораторной работе №3**

**по дисциплине «Программирование в Linux»**

**на тему**

**«Разработка системного инструмента для аудита и мониторинга системы  
Linux»**

**Студент гр. 23Б15-пу  
Трофимов И.А.**

**Преподаватель  
Киямов Ж. У.**

**Санкт-Петербург**

**2024 г.**  
**Оглавление**

1. Цель работы	3
2. Описание задачи	3
3. Теоретическая часть	5
4. Основные шаги программы	9
5. Описание программы	11
6. Рекомендации пользователя	15
7. Рекомендации программиста	16
8. Исходный код программы	17
9. Контрольный пример	17
10. Вывод	18
11. Источники	19

## **Цель работы**

Цель работы заключается в разработке системного инструмента для аудита и мониторинга системы Linux, который будет регистрировать различные события, такие как запуск и завершение процессов, изменение файлов, сетевые операции и другие значимые действия. Программа должна сохранять информацию о событиях в журнале, обеспечивать возможность фильтрации и поиска событий по различным критериям, а также оповещать пользователя о событиях. Кроме того, должна быть предусмотрена возможность создания отчетов на основе данных журнала, включая статистику и графики. Важным аспектом является обеспечение безопасности, конфиденциальности данных и ограничение доступа к системным ресурсам, а также механизмы ротации и архивации журнала событий.

## **Описание задачи**

Задача заключается в разработке системного инструмента для аудита и мониторинга событий на операционной системе Linux. Инструмент должен собирать данные о различных событиях, происходящих в системе, таких как запуск и завершение процессов, изменение файлов, сетевые операции и другие значимые действия, а также предоставлять механизмы для поиска, фильтрации и анализа этих событий. Для этого программа должна иметь интерфейс, который позволит администратору системы эффективно работать с журналами событий, фильтровать их по различным критериям и получать уведомления о важных событиях.

## **Этапы реализации:**

## **1. Регистрация событий:**

- Программа должна отслеживать и регистрировать события в системе, такие как запуск и завершение процессов, изменение файлов, сетевые операции и другие значимые действия.

## **2. Хранение данных в журнале:**

- События должны сохраняться в журнале событий, который будет хранить информацию о времени события, типе события, пользователе и процессе.

## **3. Фильтрация и поиск:**

- Необходимо реализовать возможность фильтрации и поиска по ключевым данным, таким как время события, тип события, пользователь и процесс. Пользователь должен иметь возможность легко искать и просматривать события по заданным критериям.

## **4. Оповещения:**

- В случае обнаружения важных или подозрительных событий, программа должна отправлять уведомления, например, через электронную почту или другие способы оповещений.

## **5. Создание отчетов:**

- Программа должна иметь возможность создавать отчеты на основе данных журнала, включая статистику о событиях, а также визуализировать данные в виде графиков.

## **6. Безопасность:**

- Особое внимание должно быть уделено безопасности программы, включая ограничение привилегий доступа к системным ресурсам и защите от несанкционированного доступа.

#### **7. Ротация и архивация журнала:**

- Для предотвращения переполнения диска необходимо реализовать механизмы ротации и архивации журнала событий, чтобы старые данные архивировались или удалялись.

#### **8. Конфиденциальность данных:**

- Программа должна обеспечивать конфиденциальность данных, защищая чувствительную информацию, такую как данные пользователей, от утечек или несанкционированного доступа.

#### **9. Тестирование и документация:**

- Программа должна быть протестирована на наличие ошибок, а также быть снабжена документацией, описывающей установку, настройку и использование инструмента.

### **Теоретическая часть**

Современные операционные системы, такие как Linux, предоставляют большое количество данных о происходящих событиях, которые могут быть использованы для аудита и мониторинга безопасности системы. Правильное отслеживание и анализ этих событий позволяет повысить безопасность, обеспечить соответствие политике конфиденциальности и своевременно реагировать на подозрительные действия. Разработка системного инструмента для аудита системы требует глубоких знаний о работе

операционной системы, системных вызовах и методах обработки и хранения данных.

Аудит и мониторинг системы являются неотъемлемыми частями обеспечения безопасности, поскольку они позволяют администратору отслеживать действия пользователей, обнаруживать потенциальные угрозы и анализировать влияние системных изменений. С помощью инструментов аудита можно не только отслеживать события в реальном времени, но и генерировать отчеты и проводить детальный анализ.

### **Тезисы теоретической части:**

#### **1. Роль системного аудита в безопасности:**

- Аудит является важным элементом обеспечения безопасности системы. Он позволяет фиксировать все значимые события, такие как запуск процессов, изменения в файловой системе, сетевые операции и действия пользователей, что помогает администратору выявить возможные угрозы и несанкционированный доступ.

#### **2. Типы событий, подлежащих регистрации:**

- В операционных системах Linux важно отслеживать разнообразные события:
  - Запуск и завершение процессов, которые могут быть использованы для анализа активности приложений.
  - Изменение файлов (например, создание, удаление, модификация), что может свидетельствовать о действиях пользователя или программы.

- Сетевые операции, такие как открытие портов, передача данных, что полезно для мониторинга сетевой безопасности.

### 3. Системные вызовы для аудита:

- Для регистрации событий в системе можно использовать системные вызовы и утилиты, такие как `ptrace`, `auditd` и другие инструменты, предоставляемые Linux.
  - `ptrace` позволяет отслеживать действия процессов, включая их запуск, завершение и выполнение инструкций.
  - `auditd` — это демон для аудита системы, который позволяет регистрировать события и действия в системе, такие как изменение конфигурационных файлов, действия с правами доступа и другие.

### 4. Журнал событий:

- Важной частью аудита является создание и хранение журнала событий. Журнал должен быть структурирован и включать информацию о времени события, пользователе, типе события и процессе. Это позволит быстро идентифицировать и исследовать подозрительные действия.

### 5. Фильтрация и поиск по журналу:

- Для эффективного анализа данных журнала необходимо реализовать систему фильтрации и поиска, которая позволит искать события по различным критериям, таким как тип события, пользователь или время. Это значительно ускоряет процесс обнаружения важных событий и подозрительных действий.

## **6. Оповещения и уведомления:**

- Важно иметь систему оповещений, которая информирует администратора о подозрительных событиях, таких как несанкционированный доступ или изменение важных системных файлов. Оповещения могут быть отправлены по электронной почте, через систему сообщений или другие средства уведомления.

## **7. Ротация и архивация журнала:**

- Для предотвращения переполнения диска необходимо реализовать механизмы ротации журнала, что позволит автоматически архивировать или удалять старые записи. Это позволяет эффективно управлять пространством на диске, сохраняя только актуальные данные.

## **8. Конфиденциальность и защита данных:**

- При разработке системных инструментов для аудита необходимо учитывать защиту конфиденциальности данных, включая информацию о пользователях, процессе и действиях в системе. Для этого следует использовать методы шифрования, а также ограничение доступа к данным.

## **9. Тестирование и документация:**

- Разработанный инструмент должен быть тщательно протестирован для обеспечения его корректной работы в различных условиях. Также важным элементом является создание документации, которая будет описывать способы установки,



настройки и использования программы, а также возможности фильтрации, поиска и анализа данных.

## **Основные шаги программы**

### **1. Инициализация программы:**

- Настройка необходимых параметров, таких как путь к журналу событий, параметры уведомлений и конфигурация фильтров.
- Инициализация соединений с системными сервисами и демонами (например, auditd), если это необходимо для получения данных о событиях.

### **2. Подключение к системным событиям:**

- Использование системных вызовов или утилит (например, ptrace, auditd, inotify) для отслеживания значимых событий в системе, таких как запуск процессов, изменение файлов, сетевые операции и т.д.
- Регистрация и анализ системных событий в реальном времени.

### **3. Запись событий в журнал:**

- Для каждого события необходимо собирать информацию: дата и время события, тип события, пользователь, процесс, и другая релевантная информация.
- Сохранение данных о событиях в структурированном виде (например, в текстовом файле или базе данных).

### **4. Обработка и фильтрация данных:**

- Обработка полученных данных для фильтрации событий по заданным критериям: пользователи, типы событий, временные интервалы, процессы и другие параметры.
- Реализация интерфейса или команды для фильтрации данных в журнале, что позволит администратору быстро находить интересные события.

#### **5. Оповещение о событиях:**

- Реализация механизма оповещения о важных или подозрительных событиях, например, через электронную почту или систему уведомлений.
- Настройка порогов для автоматических уведомлений о событиях, таких как несанкционированный доступ или изменение критичных системных файлов.

#### **6. Ротация и архивация журнала:**

- Реализация механизма ротации журнала событий для предотвращения переполнения диска.
- Архивация старых записей и удаление неактуальных данных, при этом обеспечивается возможность доступа к архивным данным.

#### **7. Создание отчетов и статистики:**

- Разработка функций для создания отчетов на основе данных журнала: статистика по типам событий, пользователям, процессам, временным интервалам и т.д.
- Возможность генерации отчетов в различных форматах, включая текстовый или CSV.

## **8. Обеспечение безопасности и конфиденциальности:**

- Ограничение доступа к данным журнала и конфиденциальной информации, защита от несанкционированного доступа.
- Шифрование данных в журнале для защиты от утечек конфиденциальной информации.

## **9. Периодическая проверка и тестирование программы:**

- Регулярная проверка программы на корректность работы, производительность и обработку ошибок.
- Проведение тестов для проверки работоспособности всех компонентов программы, включая запись событий, фильтрацию, создание отчетов и отправку уведомлений.

## **10. Документация и инструкции по использованию:**

- Создание документации, в которой будет подробно описан процесс установки и настройки программы, а также способы использования всех функций: фильтрации, поиска, создания отчетов и оповещений.
- Подготовка инструкций для администраторов и пользователей системы.

## **Описание программы**

Программа предназначена для обнаружения и блокировки подозрительного сетевого трафика в реальном времени. Она использует библиотеку Scapy для захвата и анализа сетевых пакетов, а также реализует механизмы обнаружения аномалий и сетевых атак на основе заранее заданных правил и сигнатур. Программа анализирует такие параметры, как IP-адреса, порты,

размер пакетов, а также частоту и типы запросов, чтобы выявить потенциальные угрозы, такие как сканирование портов или DDoS-атаки.

При обнаружении подозрительного трафика программа принимает меры по блокировке источников угроз, отправляя ICMP-сообщения о недостижимости или применяя фильтрацию на уровне сети для блокировки определенных IP-адресов или портов. Все действия логируются для дальнейшего анализа и мониторинга, что позволяет улучшать эффективность работы программы и адаптировать её под новые угрозы.

Программа обеспечивает эффективную защиту сети, своевременно выявляя и блокируя угрозы, что способствует повышению безопасности информационной инфраструктуры.

Таблица 1. main.py

Функция	Описание	Результат
main	инициализация модулей	None

Таблица 2. event\_filter.py

Функция	Описание	Результат
filter_by_user	функция фильтра	None
filter_by_event_type	функция фильтра	None

Таблица 3. notification.py

Функция	Описание	Результат
send_email	отправляет отчет на почту	None

Таблица 4. GUI.py

Функция	Описание	Результат
load_file	загрузка файлов	None
parse_logs	преобразование данных	None
display_logs	отображение логов	None
apply_filters	применение фильтров	None

sort_column	сортирует колонны	None
-------------	-------------------	------

Таблица 5. logger.py

Функция	Описание	Результат
log	создает log	None

### Рекомендации пользователя

Программа для аудита и мониторинга системы Linux предназначена для повышения безопасности системы, отслеживания системных событий и анализа действий пользователей и процессов. Чтобы эффективно использовать программу, важно настроить параметры, такие как путь к журналу событий и уведомления, а также фильтры для отслеживания нужных событий. После запуска программа будет отслеживать и записывать события в журнал, анализируя их в реальном времени. Журнал содержит данные о типах событий, пользователях и процессах, что позволяет искать аномалии.

При обнаружении подозрительных событий программа отправит уведомление, если настроен правильный адрес для оповещений. Для предотвращения переполнения диска журнал автоматически архивируется и ротруется. Генерация отчетов позволяет анализировать события по различным параметрам, таким как типы событий или пользователи. Программа также шифрует журнал для защиты данных от несанкционированного доступа, а регулярное тестирование гарантирует корректность работы. Обновляйте программу, чтобы использовать последние улучшения и устранять возможные уязвимости.

### **Рекомендации программиста**

Для эффективного использования программы, разработанной для аудита и мониторинга системы Linux, важно следовать нескольким рекомендациям. Во-первых, при настройке программы убедитесь, что правильно указаны пути к журналам и настроены фильтры для нужных типов событий. Это позволит вам отслеживать только важные события, минимизируя нагрузку на систему.

Во-вторых, важно регулярно проверять корректность работы программы и проводить тестирование ее функций, включая ротацию и архивацию журналов. Настройка уведомлений — еще одна ключевая часть работы программы, поэтому обязательно укажите верный адрес для оповещений, чтобы оперативно реагировать на подозрительные события.

Также уделите внимание безопасности: программа должна работать с минимальными привилегиями, чтобы предотвратить возможные уязвимости.



Используйте шифрование для защиты журнала событий и проверяйте права доступа.

Не забывайте о периодическом обновлении программы, чтобы обеспечить ее совместимость с последними версиями операционной системы и минимизировать риски безопасности. Обновления могут включать исправления ошибок и новые функции, которые улучшат производительность и функциональность программы.

**Исходный код программы:**

<https://github.com/hanglider/Jasur-Labs/tree/main/linux>

**Контрольный пример**

Запустите main.py (Рис. 1)

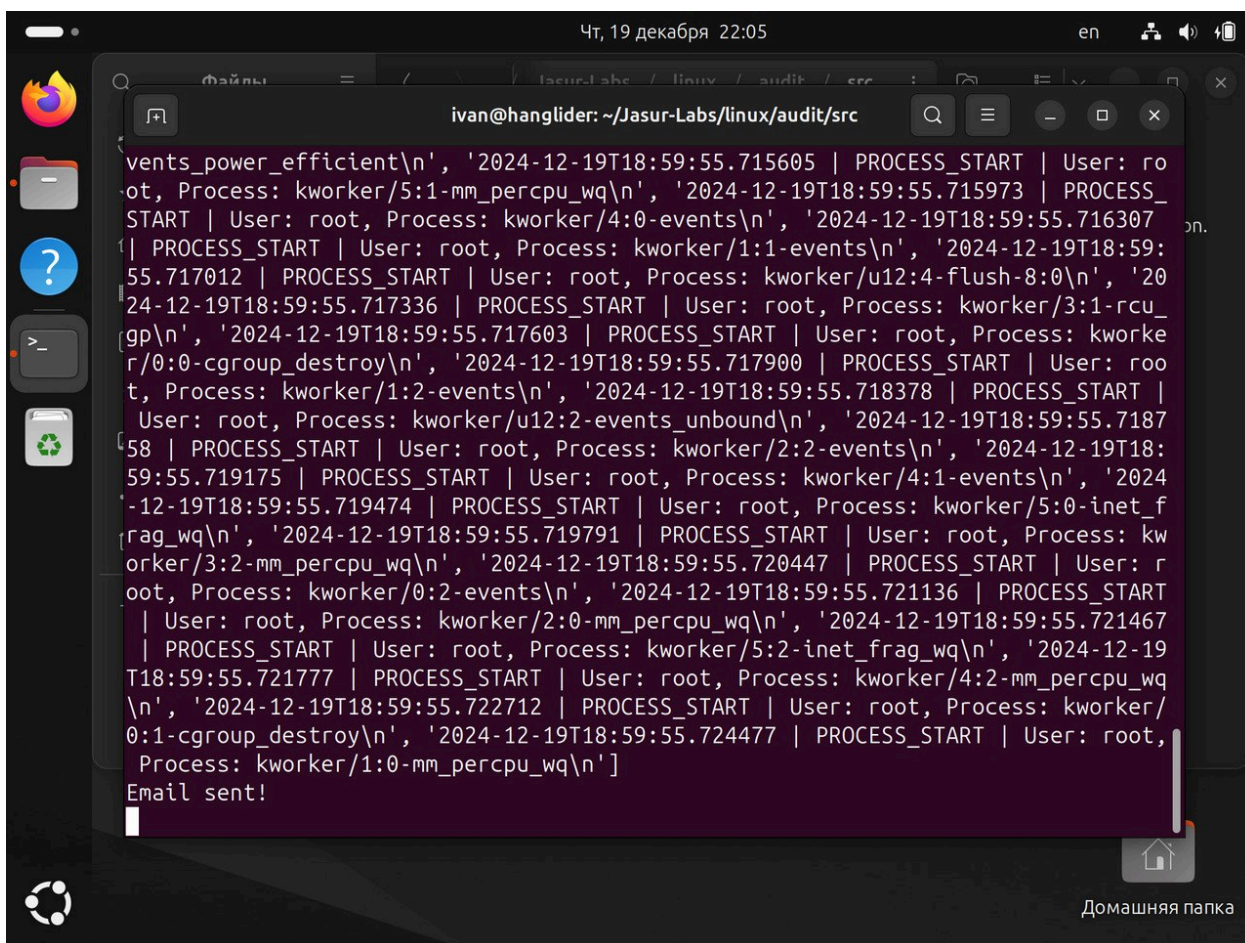


Рис.1 Рабочая область

Скрипт проведет анализ системы, результаты которого отобразятся прямо в терминале, а также в log файле. После чего отправятся на почту.

## Вывод

В ходе разработки программы для аудита и мониторинга системы Linux было создано эффективное решение для отслеживания и анализа системных событий. Программа позволяет фиксировать важные действия, такие как запуск и завершение процессов, изменения в файловой системе и сетевые операции, а также предоставляет механизмы для фильтрации, поиска и генерации отчетов на основе собранных данных. Важной частью является настройка уведомлений, позволяющая оперативно реагировать на

подозрительные события. Программа также уделяет внимание безопасности, шифруя журнал событий и ограничивая доступ к нему. Регулярное обновление и тестирование программы обеспечат ее надежную работу и защиту системы от потенциальных угроз. В результате, использование данной программы способствует улучшению безопасности и стабильности системы Linux, позволяя своевременно выявлять и устранять проблемы.

### **Источники**

Нет.