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

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Организация и обеспечение аудита настроек средств защиты информации»

**ОТЧЕТ ПО МОДУЛЮ №6**

«Методическая ЗИ с помощью специализированных программных средств»

**Выполнил:**

Чу Ван Доан, студент группы N3347

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

(подпись)

**Проверил:**

Пенин Андрей Семенович

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

(отметка о выполнении)

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

(подпись)

# Содержание

[**Содержание 2**](#_heading=h.3jbepr30actf)

[**Введение 3**](#_heading=h.gjdgxs)

[**Задание 4**](#_heading=h.1fob9te)

[1. Этап 1: Установка и настройка DLP-системы 6](#_heading=h.k303886ij2nw)

[2. Этап 2: Установка и настройка SIEM-системы 9](#_heading=h.2uwt70x11w3s)

[3. Этап 3: Установка и настройка системы резервного копирования 12](#_heading=h.26lbr3chzcp)

[**Заключение 15**](#_heading=h.jhow0r2henwe)

# Введение

В современном мире информационной безопасности особое значение приобретает способность организаций своевременно обнаруживать и реагировать на подозрительную активность в своих информационных системах. Одним из эффективных инструментов для решения этой задачи являются системы управления информацией и событиями безопасности (SIEM).

В рамках данной лабораторной работы была произведена установка и настройка SIEM-системы Wazuh, которая представляет собой современное и бесплатное решение для мониторинга безопасности. Основной целью работы являлось развертывание Wazuh в одновузловой конфигурации, подключение агента безопасности, сбор системных логов и настройка правил для обнаружения потенциальных угроз — в частности, множественных неудачных попыток входа в систему по SSH.

Работа включает в себя полную цепочку: от развертывания компонентов Wazuh (Dashboard, Manager, Indexer), настройки агента, до моделирования инцидента и фиксации его в интерфейсе SIEM-системы.

# Задание

Необходимо установить и настроить систему предотвращения утечек данных (DLP), систему управления информацией и событиями безопасности (SIEM), а также систему резервного копирования данных. Работа выполняется на виртуальной машине (рекомендуется использовать ту же, что и в предыдущих ПР) и включает три этапа. Результаты оформляются в отчёт.

Этап 1. Установка и настройка DLP-системы (до 3 баллов)

- Установите программное обеспечение для предотвращения утечек данных (например, OpenDLP, MyDLP или аналоги).

- Настройте параметры мониторинга: определите защищаемые данные (например, текстовые файлы с заданным шаблоном, такие как номера кредитных карт или паспортов); установите правила блокировки или журналирования попыток копирования данных (например, на USB или через сеть).

- Проведите тестирование: попробуйте скопировать защищённые данные на внешний носитель или передать через сеть; зафиксируйте срабатывание DLP (логи, уведомления).

Этап 2. Установка и настройка SIEM-системы (до 4 баллов)

- Установите программное обеспечение для SIEM (например, ELK Stack с Elastic SIEM, Splunk Free, Graylog или аналоги).

- Настройте сбор и анализ событий: подключите источник логов (например, системные логи ОС или приложения); создайте правило обнаружения подозрительной активности (например, многократные неудачные попытки входа).

- Проведите тестирование: сгенерируйте тестовые события (например, несколько раз введите неверный пароль); продемонстрируйте, что SIEM зафиксировала событие и выдала уведомление или отчёт.

Этап 3. Установка и настройка системы резервного копирования (до 3 баллов)

- Установите программное обеспечение для резервного копирования (например, Bacula, Duplicati, rsync или встроенные средства Windows).

- Настройте параметры: задайте расписание автоматического создания резервных копий (например, ежедневно); выберите каталог для копирования (например, /important или C:\Data) и место хранения копий.

- Проведите тестирование: создайте резервную копию тестовых данных; удалите или измените исходные файлы и восстановите их из копии.

Отчёт должен содержать описание действий, скриншоты и выводы по каждому этапу.

Кроме того быть готовым к демонстрации ВМ на защите работы.

**Ход работы**

## Этап 1: Установка и настройка DLP-системы

Я написал программу на Python с аналогичным функционалом:

* Мониторинг каталога
* Обнаружение конфиденциальных данных (номера кредитных карт, паспортов)
* Ведение журнала при обнаружении

Цели:

* Создание системы мониторинга конфиденциальных данных.
* Обнаружение утечек данных при копировании пользователем файлов, содержащих важную информацию.
* Регистрация всех обнаруженных событий в журнале для последующей проверки и анализа.

**Листинг А.1 – Код файла dlp\_monitor.py**

import os

import re

import time

from datetime import datetime

MONITORED\_DIR = os.path.expanduser("~/Desktop/Task\_1/dlp\_test")

LOG\_FILE = os.path.expanduser("~/Desktop/Task\_1/dlp\_alerts.log")

PATTERNS = {

"Credit Card": re.compile(r"\b(?:\d[ -]\*?){13,16}\b"),

"Passport Number": re.compile(r"\b[A-Z]{2}\d{7}\b")

}

def write\_log(message):

try:

with open(LOG\_FILE, "a", encoding="utf-8") as logf:

logf.write(message + "\n")

print(f"[LOG] {message}")

except Exception as e:

print(f"[ERROR] Failed to write log: {e}")

def scan\_file(filepath):

try:

with open(filepath, "r", encoding="utf-8", errors="ignore") as f:

content = f.read()

for label, pattern in PATTERNS.items():

if pattern.search(content):

log = f"[{datetime.now()}] {label} detected in {filepath}"

write\_log(log)

except Exception as e:

print(f"[ERROR] Cannot read file {filepath}: {e}")

def monitor():

print(f"Monitoring folder: {MONITORED\_DIR}")

while True:

for root, \_, files in os.walk(MONITORED\_DIR):

for name in files:

path = os.path.join(root, name)

scan\_file(path)

time.sleep(3)

if \_\_name\_\_ == "\_\_main\_\_":

if not os.path.exists(MONITORED\_DIR):

os.makedirs(MONITORED\_DIR)

print(f"Created folder: {MONITORED\_DIR}")

monitor()

Правила обнаружения:

* Каждые 3 секунды выполняется сканирование всей директории.
* Если содержимое любого файла соответствует следующим шаблонам:
  + Номер кредитной карты: от 13 до 16 цифр подряд
  + Номер паспорта: 2 буквы + 7 цифр

→ тогда событие фиксируется в журнале (log-файле).

Создание файла с конфиденциальными данными:

echo "Credit Card: 4111-1111-1111-1111" > ~/Desktop/Task\_1/dlp\_test/test.txt

echo "Passport Number: AB1234567" >> ~/Desktop/Task\_1/dlp\_test/test.txt

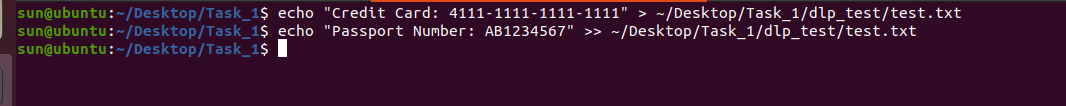


Рисунок 1 - Создание файла

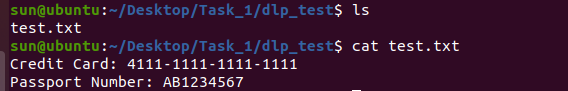


Рисунок 2 - Содержание файла

Моделирование поведения копирования файла:

mkdir -p ~/Desktop/usb\_fake

cp ~/Desktop/Task\_1/dlp\_test/test.txt ~/Desktop/usb\_fake/

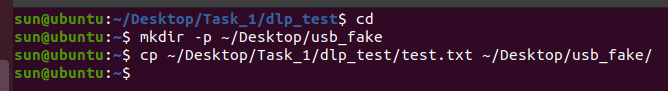


Рисунок 3 - Моделирование поведения копирования файла

Проверка журнала событий:

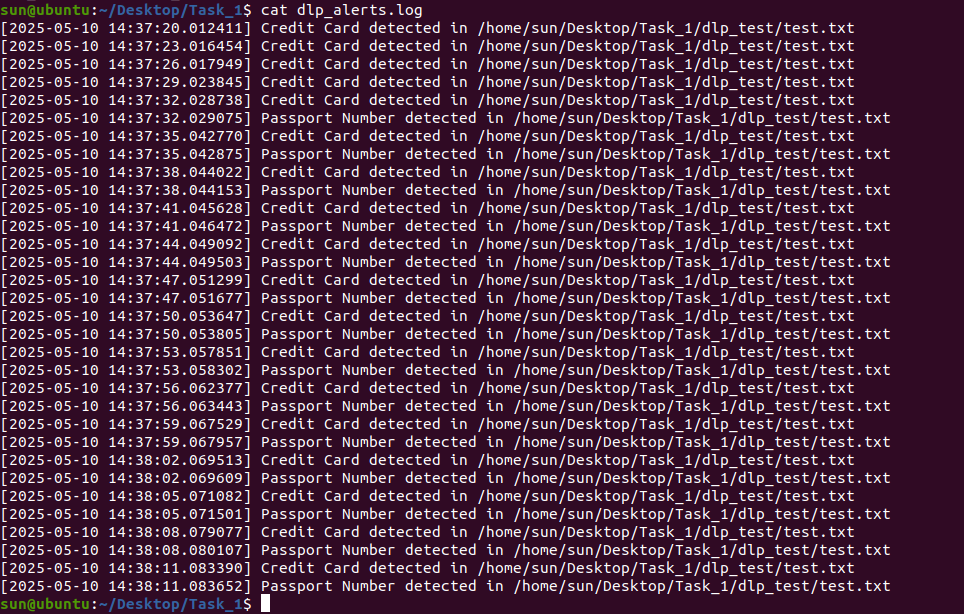


Рисунок 3 - Проверка файла log

Вывод:  
 Система DLP работает в соответствии с требованиями задания:

* Автоматически обнаруживает конфиденциальные данные
* Регистрирует события утечки в журнал
* Осуществляет мониторинг каталога в режиме реального времени

## Этап 2: Установка и настройка SIEM-системы

Установить Docker и Docker Compose

sudo apt update

sudo apt install -y docker.io docker-compose

sudo systemctl enable docker

sudo systemctl start docker

Развертывание Wazuh с помощью Docker

Загрузка исходного кода Docker Compose:

git clone https://github.com/wazuh/wazuh-docker.git -b v4.12.0

cd wazuh-docker/single-node

Создание SSL-сертификатов:

docker-compose -f generate-indexer-certs.yml run --rm generator

Эта команда создаст сертификаты в каталоге config/wazuh\_indexer\_ssl\_certs.

Запуск системы Wazuh:

docker-compose up -d

Доступ к панели управления:

Откройте браузер и перейдите по адресу: http://127.0.1:5601

Используйте следующие учетные данные для входа:

* Имя пользователя: admin
* Пароль: SecretPassword

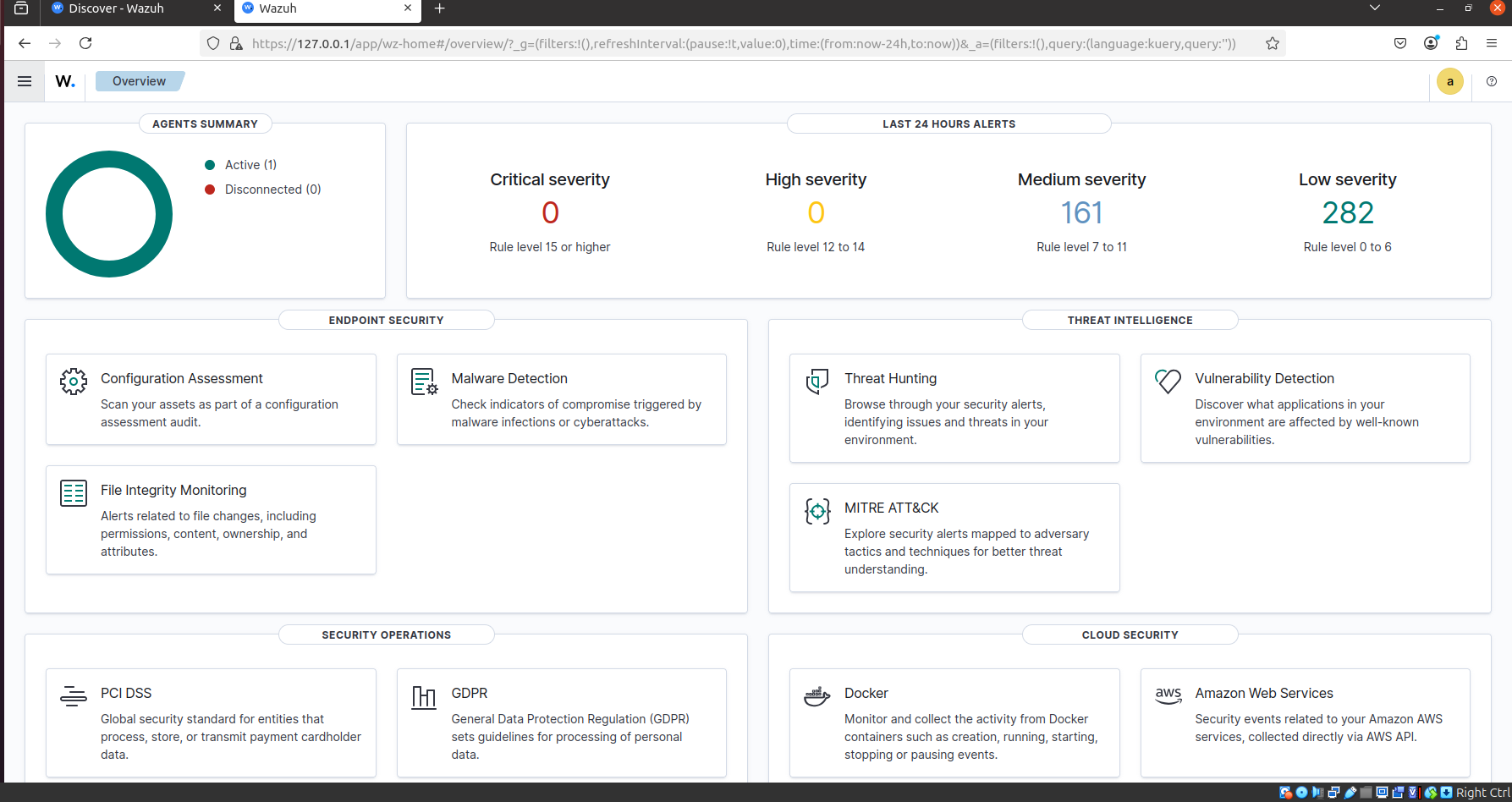


Рисунок 4 - вход

Установка Wazuh Agent на Ubuntu

Установка Wazuh Agent

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/wazuh.gpg

echo "deb https://packages.wazuh.com/4.x/apt stable main" | sudo tee /etc/apt/sources.list.d/wazuh.list

sudo apt update

sudo apt install wazuh-agent -y

Настройка отправки логов на Wazuh Manager

Открыть файл конфигурации:

sudo nano /var/ossec/etc/ossec.conf

Найти блок <server> и изменить его на следующий:

<server>

<address>127.0.0.1</address>

</server>

Запуск и проверка состояния агента

sudo systemctl enable wazuh-agent

sudo systemctl start wazuh-agent

sudo systemctl status wazuh-agent

Установка и активация SSH-сервера на Ubuntu

Установка OpenSSH-сервера:

sudo apt update

sudo apt install openssh-server -y

Запуск и активация SSH-сервиса:

sudo systemctl enable ssh

sudo systemctl start ssh

(Необязательно) Создание тестового пользователя:

sudo adduser testuser

→ Введите пароль и любые данные по запросу.

Проверка работы и генерация события:

Подключитесь по SSH к localhost:

ssh testuser@127.0.0.1

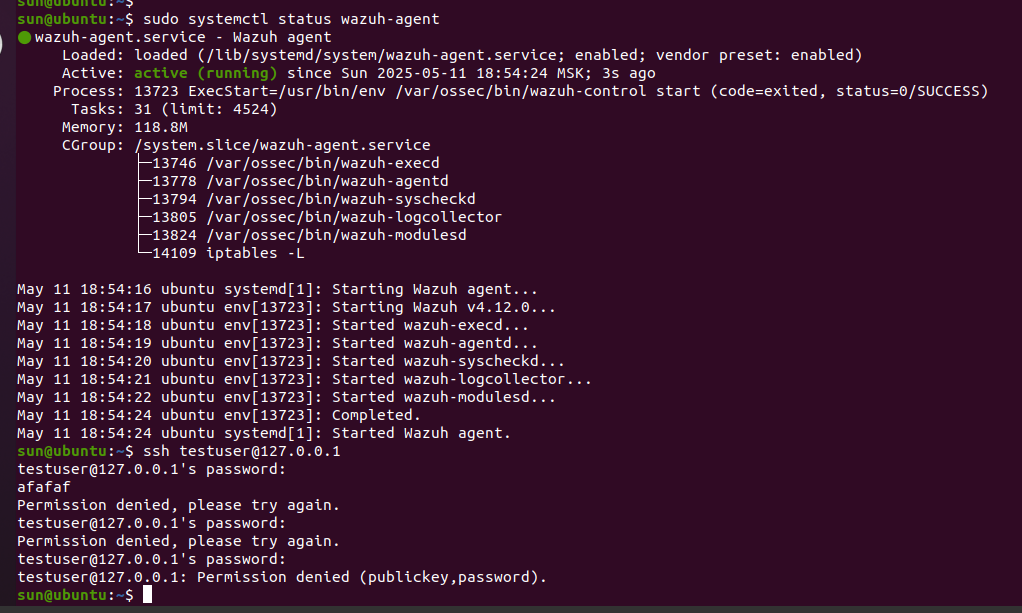


Рисунок 5 - Подключитесь по SSH к localhost

→ Введите несколько раз неправильный пароль, чтобы сгенерировать предупреждение "Failed password".

Проверка события в Wazuh Dashboard:

Перейдите в Security Events и отфильтруйте по ключевым словам:

sshd

authentication failure

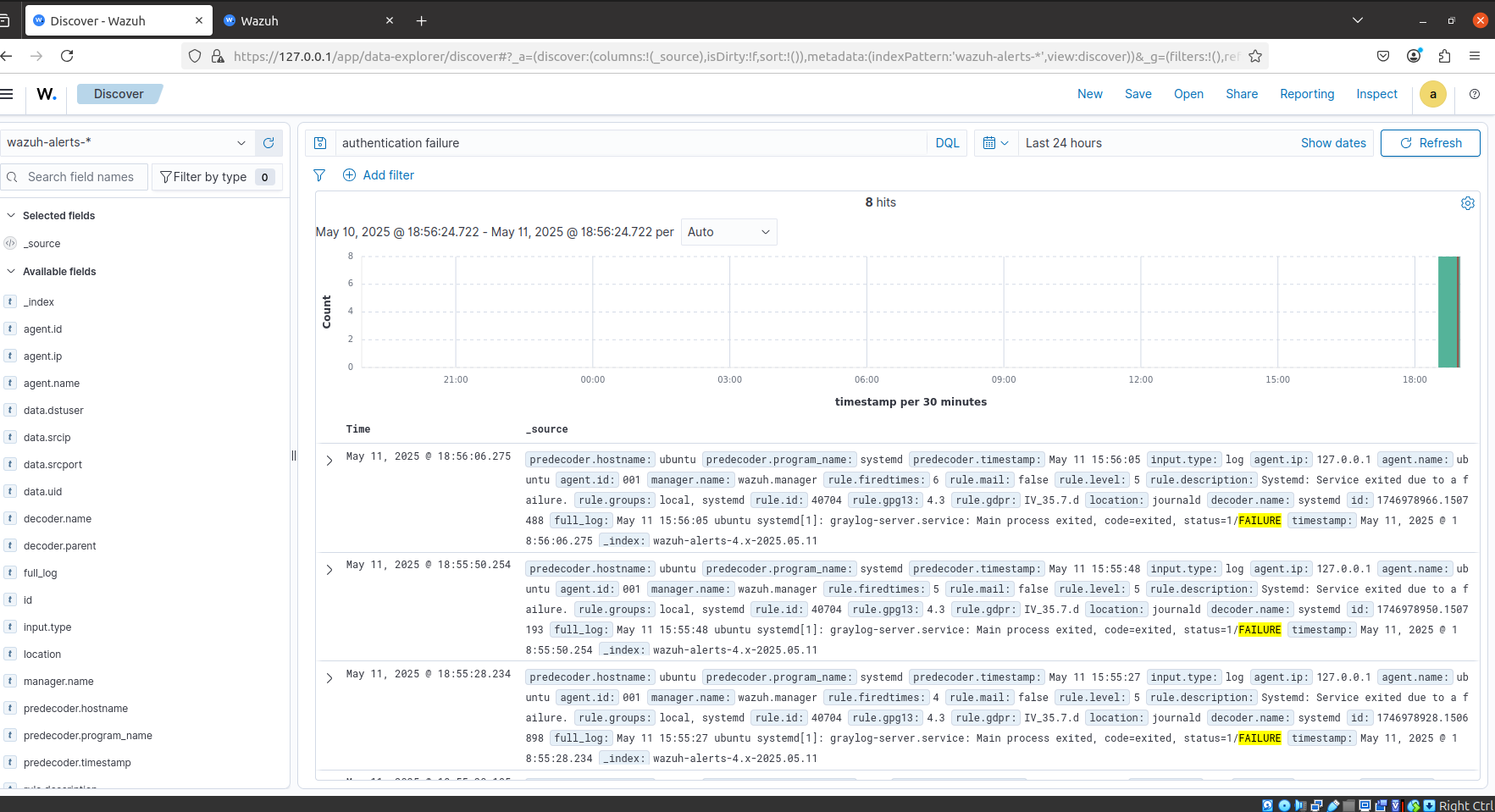


Рисунок 6 - Результат

## Этап 3: Установка и настройка системы резервного копирования

Установка инструмента rsync

sudo apt update

sudo apt install -y rsync

Создание исходной директории и директории для резервных копий

mkdir -p ~/important

mkdir -p ~/backup

Добавление тестовых данных:

echo "This is critical data." > ~/important/important\_file.txt

Написание скрипта автоматического резервного копирования

Создание файла скрипта:

nano ~/backup.sh

Содержимое скрипта:

#!/bin/bash

rsync -av --delete ~/important/ ~/backup/

Сохранение и предоставление прав на выполнение:

chmod +x ~/backup.sh

Команда rsync -av --delete синхронизирует все содержимое из ~/important в ~/backup. Если файл будет удалён из исходной папки, резервная копия также будет обновлена соответствующим образом.

Настройка ежедневного резервного копирования с помощью cron

Открыть планировщик:

crontab -e

Добавить строку для запуска скрипта каждый день в 2 часа ночи:

0 2 \* \* \* /home/<имя\_пользователя>/backup.sh

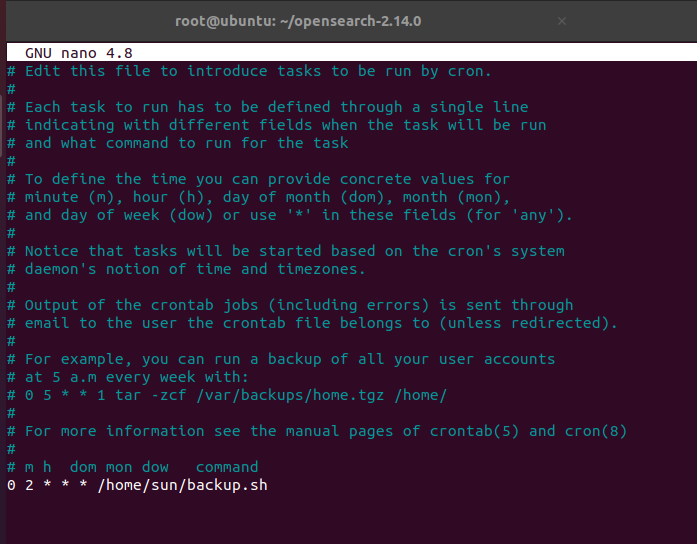


Рисунок 7 - Добавить строку для запуска скрипта каждый день в 2 часа ночи

Тестирование резервного копирования

Выполнить резервное копирование вручную:

./backup.sh

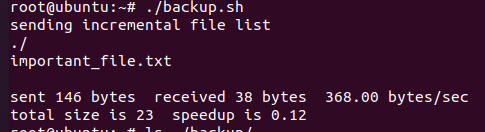


Рисунок 8- Выполнить резервное копирование вручную

Проверить содержимое резервной копии:

ls ~/backup/

cat ~/backup/important\_file.txt

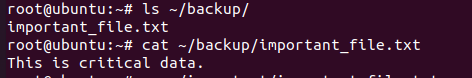


Рисунок 9 - Проверить содержимое резервной копии

Удаление исходного файла

rm ~/important/important\_file.txt

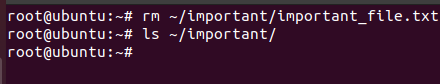


Рисунок 10 - Удаление исходного файла

Восстановление из резервной копии

cp ~/backup/important\_file.txt ~/important/

Проверка:

cat ~/important/important\_file.txt

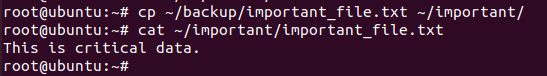


Рисунок 10 - Восстановление прошло успешно

# Заключение

В ходе выполнения лабораторной работы была успешно установлена и настроена SIEM-система Wazuh в одновузловой конфигурации с использованием Docker. Проведена интеграция агента безопасности на защищаемом узле, а также реализована базовая проверка работоспособности системы путём моделирования инцидента — многократных неудачных попыток входа в систему по SSH.

Wazuh продемонстрировал свою эффективность в сборе и корреляции событий безопасности, отображении информации о выявленных угрозах в реальном времени, а также в удобстве анализа инцидентов через веб-интерфейс. Благодаря открытой архитектуре и поддержке множества форматов логов, система может масштабироваться и адаптироваться под различные инфраструктуры.

Таким образом, поставленные цели лабораторной работы были достигнуты. Полученные знания и навыки могут быть применены при построении более сложных решений в области мониторинга и управления информационной безопасностью в организациях.

**ПРИЛОЖЕНИЕ А**

**Листинг А.1 – Код файла main.py**