***Обнаружение аномальной сетевой активности на хосте***

**Программный макет и исходный код**

**Созданный программный макет**

Программный макет представляет собой *Python*-скрипт, который автоматически собирает сетевой трафик, обучает автоэнкодер для обнаружения аномалий и проверяет новые данные на наличие аномалий. Основные функции макета:

**Сбор трафика:** Используется библиотека *scapy* для захвата сетевых пакетов.

**Оцифровка данных:** Пакеты преобразуются в числовые признаки (IP-адреса, порты, длина пакета).

**Обучение автоэнкодера:** Используется нейронная сеть для обучения на нормальном трафике.

**Обнаружение аномалий:** Новые данные проверяются на отклонения от нормального поведения.

**Формирование отчетов:** Результаты сохраняются в файл и выводятся на консоль.

**Исходный код**

Исходный код представлен в файле *main.py*. Он включает:

Функции для сбора трафика (*continuous\_capture*).

Функции для оцифровки данных (*packet\_to\_features, pcap\_to\_dataframe*).

Функции для обучения автоэнкодера (*train\_autoencoder*).

Функции для обнаружения аномалий (*detect\_anomalies*).

Функции для формирования отчетов (*generate\_report*).

**Описание программного макета**

**Цель**

Программный макет предназначен для обнаружения аномальной сетевой активности на хосте с использованием методов машинного обучения. Он автоматически собирает трафик, обучает модель и проверяет новые данные на наличие аномалий.

**Основные функции**

1. **Сбор трафика:**

Захват пакетов на указанном сетевом интерфейсе.

Сохранение данных в формате .*pcap*.

1. **Оцифровка данных:**

Преобразование пакетов в числовые признаки (IP-адреса, порты, длина пакета).

Сохранение данных в формате *.csv*.

1. **Обучение автоэнкодера:**

Использование нейронной сети для обучения на нормальном трафике.

Сохранение обученной модели и *scaler*.

1. **Обнаружение аномалий:**

Проверка новых данных на отклонения от нормального поведения.

Формирование отчетов об обнаруженных аномалиях.

1. **Формирование отчетов:**

Сохранение результатов в файл и вывод на консоль.

**Инструкция по установке**

1. **Установка зависимостей:**  
   Убедитесь, что установлены все необходимые библиотеки. Для этого выполните команду:

*pip install scapy pandas numpy tensorflow scikit-learn joblib matplotlib pyyaml psutil*

1. **Загрузка исходного кода:**  
   Скопируйте файл *main.py* и конфигурационный файл *config.yaml* в рабочую директорию.
2. **Настройка конфигурации:**  
   Отредактируйте файл *config.yaml* для указания параметров:

*path:*

*model\_path: 'autoencoder\_model.h5'*

*scaler\_path: 'scaler.save'*

*network\_settings:*

*interface\_type: 'wi-fi' # Или "ethernet"*

*interval: 1800*

*total\_duration: 86400*

*capture\_duration: 3600*

*check\_interval: 0*

1. **Запуск программы:**  
   Запустите скрипт с помощью команды:

*python main.py*

**Инструкция по использованию**

1. **Обучение модели:**

Если модель и *scaler* отсутствуют, программа автоматически соберет обучающую выборку и обучит автоэнкодер.

Обучающая выборка сохраняется в файл *training\_traffic.pcap*.

1. **Обнаружение аномалий:**

Программа начинает непрерывный сбор и проверку данных на указанном интерфейсе.

Результаты проверки сохраняются в папке *report/anomaly\_report*.

1. **Просмотр отчетов:**

Отчеты об обнаруженных аномалиях сохраняются в текстовых файлах в папке *report/anomaly\_report*.

Пример отчета:

Detected 3 anomalies

Threshold: 0.1234

Anomaly detected at packet index 10 with error 0.5678

Anomaly detected at packet index 20 with error 0.6789

Anomaly detected at packet index 30 with error 0.7890

**Результаты тестирования**

**Тестовые данные**

Для тестирования использовались:

Нормальный трафик: Захват трафика в течение 24 часов.

Аномальный трафик: Имитация аномалий (например, сканирование портов, необычные IP-адреса).

**Примеры обнаружения аномалий**

1. **Пример 1: Сканирование портов**

Обнаружено: 5 аномалий.

Порог: 0.1234.

Пример отчета:

Detected 5 anomalies

Threshold: 0.1234

Anomaly detected at packet index 15 with error 0.5678

Anomaly detected at packet index 25 with error 0.6789

Anomaly detected at packet index 35 with error 0.7890

Anomaly detected at packet index 45 with error 0.8901

Anomaly detected at packet index 55 with error 0.9012

1. **Пример 2: Необычные IP-адреса**

Обнаружено: 3 аномалии.

Порог: 0.1234.

Пример отчета:

Detected 3 anomalies

Threshold: 0.1234

Anomaly detected at packet index 10 with error 0.5678

Anomaly detected at packet index 20 with error 0.6789

Anomaly detected at packet index 30 with error 0.7890

**Выводы**

Программа успешно обнаруживает аномалии в сетевом трафике.

Порог аномальности можно настроить в зависимости от требований.

Результаты сохраняются в удобном формате для дальнейшего анализа.

**Заключение**

Программный макет успешно решает задачу обнаружения аномальной сетевой активности. Он может быть использован для мониторинга сетевого трафика и выявления потенциальных угроз.