

## РЕФЕРАТ

**Пояснительная записка** 99 страниц, 46 рисунков, 2 таблицы, 16 источников, 4 приложения.

**Ключевые слова:**

Беспроводная сеть

Перехват сетевых пакетов

Обработка сетевых пакетов

**Обобщённые трудовые функции и трудовые функции из профессионального стандарта:** ПС 06.027, ОТФ С (С/15.6, С/31.6)

**Объект выпускной квалификационной работы** – беспроводная вычислительная сеть производственного предприятия.

**Цель выпускной квалификационной работы** – проектирование и последующее создание информационной подсистемы исследования трафика в беспроводных сетях.

**Способ реализации** – создание и описание методики, направленной на перехват, хранение, обработку и классификацию сетевых пакетов в беспроводных сетях, в зависимости от содержимого этих сетевых пакетов.

**Операционная система и устройства, обеспечивающие работу информационной подсистемы**

Сценарии на языке BASH должны исполняться в операционной системе GNU/Linux семейства Debian и CentOS.

**Язык программирования**

BASH

**Полученные результаты** – два интерактивных сценария на языке BASH направленные на настройку и сбор с трафика в беспроводных сетях. Готовые фильтры Logstash, которые отбрасывают лишние заголовки сетевых пакетов.

**Область применения** – беспроводная локальная вычислительная сеть предприятия.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
1 ПРЕДМЕТНАЯ ОБЛАСТЬ АВТОМАТИЗАЦИИ.....	6
1.1 Постановка задачи.....	6
1.2 Особенности перехвата беспроводного трафика в беспроводной локальной вычислительной сети.....	7
1.2.1 Перекрёстные помехи в беспроводных .....	7
1.2.2 Обнаружение и анализ беспроводных сигналов.....	8
1.3 Устройство и функционирование современных ИС.....	9
1.3.1 Основы современных операционных систем.....	11
1.3.2 Основы современных систем управления базами данных.....	12
1.3.2.1 Модель «Сущность-связь» .....	12
1.3.2.2 Реляционная модель.....	13
1.3.2.3 Системы NOSQL.....	14
1.3.3 Программные средства и платформы инфраструктуры информационных технологий организаций. Современные подходы и стандарты автоматизации организации .....	16
1.4 Языки современных бизнес-приложений.....	19
1.5 Инструменты и методы прототипирования пользовательского интерфейса.....	21
1.6 Современные методики тестирования разрабатываемых ИС.....	22
1.6.1 Методология тестирования «белого ящика».....	22
1.6.2 Метод тестирования базового пути.....	23
1.6.3 Способы тестирования условий.....	23
1.6.4 Тестирование ветвей и операторов отношений.....	23
1.6.5 Метод потоков данных.....	23
1.6.6 Тестирование циклов.....	24
2 ПРОЕКТИРОВАНИЕ ПОДСИСТЕМЫ АНАЛИЗА ТРАФИКА В БЕСПРОВОДНЫХ СЕТЯХ.....	25
2.1 Анализ бизнес-процессов организации .....	25
2.2 Основы информационной безопасности организации .....	28
2.3 Предложения по внедрению или модернизации программных, или аппаратных средств.....	30
2.4 Анализ существующих решений для обработки сетевых пакетов.....	30
2.5 Режимы работы адаптера беспроводной связи.....	33
2.6 Разработка прототипа ИС в соответствии с требованиями.....	34
2.6.1 Установка и настройка Elastic Stack.....	36
2.6.1.1 Настройка Elasticsearch.....	37
2.6.1.2 Настройка Kibana.....	38
2.6.1.3 Настройка Logstash.....	39
2.6.1.4 Настройка Filebeat.....	40
2.6.2 Анализ прецедентов.....	41

2.6.3 Формирование функциональных требований к сценариям на командном процессоре Bash.....	44
2.6.4 Диаграмма последовательности.....	45
2.6.5 Проектирование интерфейса пользователя.....	47
2.7 Определение необходимого уровня прав доступа, управление правами доступа к репозиторию данных о выполнении работ по созданию (модификации) и сопровождению ИС.....	51
3 РАЗРАБОТКА ПОДСИСТЕМЫ.....	53
3.1 Описание процесса разработки информационной подсистемы.....	53
3.1.1 Настройка параметров безопасности Elastic Stack.....	53
3.1.2 Разработка сценариев взаимодействия с подсистемой хранения трафика в беспроводной сети.....	59
3.1.2.1 Реализация функции проверки прав суперпользователя.....	59
3.1.2.2 Реализация функции определения операционной системы на которой запущен сценарий .....	60
3.1.2.3 Реализация функции изменения режима работы адаптера беспроводной сети.....	61
3.1.2.4 Реализация функции вызова сценария для подключения к беспроводной сети.....	61
3.1.2.5 Реализация функции сканирования установленных пакетов в ОС и автоматической установки их в ОС.....	62
3.1.2.6 Реализация функции циклической смены каналов на адаптере беспроводной сети .....	63
3.1.2.7 Реализация функции запуска сценария для сбора сетевого трафика в беспроводной сети без использования фильтров.....	63
3.1.2.8 Реализация функции интерактивного меню.....	64
3.2 Описание работы информационной подсистемы.....	65
3.3 Тестирование прототипа ИС на проверку корректности архитектурных решений. ....	71
3.4 Анализ результатов тестов. Принятие решения о пригодности архитектуры .....	76
ЗАКЛЮЧЕНИЕ .....	77
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	78
Приложение А.....	81
Приложение Б.....	85
Приложение В.....	98
Приложение Г.....	99

## **ВВЕДЕНИЕ**

Жизнь современной организации, которая занимается коммерческой деятельностью, совершенно невозможна без эффективного управления. Одним из самых важных аспектов эффективной работы организации являются системы обработки информации. Информационные системы обработки информации должны соответствовать следующим требованиям:

1. обеспечивать получение общих и детализированных отчётов в ходе и по итогам работы;
2. обеспечивать получение информации, которая является критической по времени, без задержек;
3. давать возможность быстро и эффективно вносить новые данные;
4. иметь перспективы для расширения функционала.

Целью выпускной квалификационной работы является проектирование и последующее создание информационной подсистемы исследования трафика в беспроводных сетях.

Для достижения цели необходимо решить ряд задач, в соответствии с которыми построена структура работы:

1. произвести анализ бизнес-процессов организации;
2. проанализировать существующие решения для обработки сетевых пакетов;
3. сформулировать требования к информационной подсистеме;
4. разработать прототип информационной подсистемы в соответствии с требованиями;
5. определить необходимый уровень прав доступа к репозиторию данных о выполнении работ по созданию и сопровождению информационной подсистемы.

Актуальность данной темы обусловлена непрерывным развитием технологий беспроводной связи и повсеместным использованием данных технологий в различных организациях, что в свою очередь, например, с целью

информационной защиты или выявления других аномалий в беспроводной сети обязывает иметь некую подсистему, которая выполняет функции сбора, хранения и обработки сетевых пакетов, учитывая особенности перехвата трафика в беспроводных сетях.

Первый раздел содержит техническое задание и теоретические сведения, которые необходимы для успешной реализации проектирования информационной системы.

Второй раздел содержит подробный анализ предмета разработки и описание процесса проектирования информационной подсистемы сбора и хранения для последующего анализа и обработки трафика в беспроводных сетях.

Третий раздел содержит описание процесса разработки информационной подсистемы исследования трафика в беспроводных сетях.

В приложениях находится листинг сценариев командной оболочки Shell, инструкции установки Elastic Stack, и содержание файла конфигурации фильтров Logstash.

# **1 ПРЕДМЕТНАЯ ОБЛАСТЬ АВТОМАТИЗАЦИИ**

## **1.1 Постановка задачи**

Каждый день в вычислительной сети может произойти множество разных событий: от простого заражения шпионской программой до сложной ошибки конфигурирования маршрутизатора.

Для того, чтобы разобраться в затруднениях, которые возникают в сети, необходимо перейти на уровень пакетов. Все ошибки в компьютерных сетях начинаются именно на этом уровне. Тут могут обнаружиться некорректные организации, даже в самых корректно реализованных приложениях, а заслуживающие, на первый взгляд полного доверия сетевые протоколы могут оказаться зловредными.

Целью дипломной работы является разработка информационной подсистемы анализа трафика в беспроводных, которая будет выполнять следующие функции:

1. Перехват сетевых пакетов
2. Хранение сетевых пакетов
3. Обработка и классификация сетевых пакетов в зависимости от содержимого этого пакета с целью последующего анализа для выявления аномалий сетевого трафика.

Поиск эффективных методов выявления аномальных состояний в работе сетей передачи данных в настоящее время остаётся актуальной научной задачей. Нарушения являются следствием программных сбоев, отказов аппаратуры или нарушений информационной защиты.

В отличие от традиционных проводных сетей организация беспроводных сетей несколько отличается. С одной стороны, в беспроводных сетях применяются обычные сетевые протоколы, такие как TCP и IP, однако порядок действий несколько изменяется при переходе от самых нижних уровней модели OSI. В данном случае необходимо принимать во внимание

особенности протоколов беспроводной сети, таких как 802.11, которые в отличие от протоколов Ethernet, которые не особенно изменились со временем, развиваются очень быстро.

Для понимания процесса перехвата сетевого трафика, передаваемого по беспроводной сети, необходимо рассмотреть особенности физической среды передачи данных в беспроводных сетях.

## **1.2 Особенности перехвата беспроводного трафика в беспроводной локальной вычислительной сети.**

Особенность перехвата сетевого трафика в беспроводной локальной вычислительной сети состоит в том, что частотный спектр беспроводной связи распределён в среде передачи информации по отдельным физическим радиоканалам. В отличие от проводных сетей, где каждый клиент подключается к коммутатору с помощью отдельного кабеля, среда передачи данных по беспроводным сетям является общей для всех клиентов, хотя радиус её действия ограничен. [8]

Благодаря такому разделению физического пространства общий частотный спектр удалось разбить на отдельные каналы связи, где канал – это часть частотного спектра в беспроводной связи. В РФ разрешено использовать первые 13 каналов беспроводной связи. Это важно, так как WLAN может работать одновременно только на одном канале, из чего следует, что анализировать пакеты можно только по очереди в отдельных каналах беспроводной сети.

### **1.2.1 Перекрёстные помехи в беспроводных**

При установлении беспроводной связи не всегда стоит полагаться на целостность данных, которые присутствуют в эфире. Это связано с тем, что сигналы, распространяемые по каналам беспроводной связи, имеют свойства накладываться друг на друга, тем самым создавая помехи. Далеко не все меры

для борьбы с перекрёстными помехами на практике оказываются действенными. Именно поэтому при подготовке к перехвату пакетов в беспроводной сети следует уделить пристальное внимание к окружающей среде. Например, чтобы в ней отсутствовали такие источники помех как большие отражающие поверхности, крупные металлические предметы, микроволновые электроприборы, мобильные телефоны, которые работают на частоте 2,4 ГГц, толстые стены или очень плотные поверхности. [8]

Перекрёстные помехи в каналах связи так же возможны. Учитывая, что анализ пакетов допустимо проводить только в одном канале, тем не менее это можно сделать с одной небольшой оговоркой: вследствие ограниченного диапазона частот, выделенного каналам связи, частотные спектры отдельных каналов могут незначительно перекрываться. Однако подобная проблема возникает достаточно редко, поскольку при развёртывании беспроводных сетей в одном районе, им, как правило выделяется не перекрывающиеся каналы связи 1, 6 и 11 [8].

### **1.2.2 Обнаружение и анализ беспроводных сигналов**

Следует понимать, что диагностика наложения сигналов невозможна, если будут анализироваться только сетевые пакеты. Если требуется заниматься специально диагностикой WLAN, то необходимо регулярно выявлять факт наложения сигналов. Задачу по выявлению наложения сигналов решает анализатор спектра. Это специальный прибор, который предназначен для проверки и отображения взаимных помех в определённом частотном спектре.

Профессиональные анализаторы спектра могут достигать стоимости три миллиона рублей, однако для повседневной практики имеется решение в виде устройства Wi-Spy, которое выпускает компания MetaGeek. Данное устройство подключается к USB порту компьютера и способно контролировать весь спектр сигналов стандарта 802.11. Посредством программного обеспечения, которое тоже поставляется компанией MetaGeek



это устройство позволяет выводить анализируемый частотный спектр в графическом виде. Пример такого вывода представлен на рисунке 1.

В данном примере графического вывода, который представлен на рисунке 1, наглядно показано, что спектры четырёх каналов связи равномерно распределены по всему диапазону частот, который был выделен для беспроводной связи.

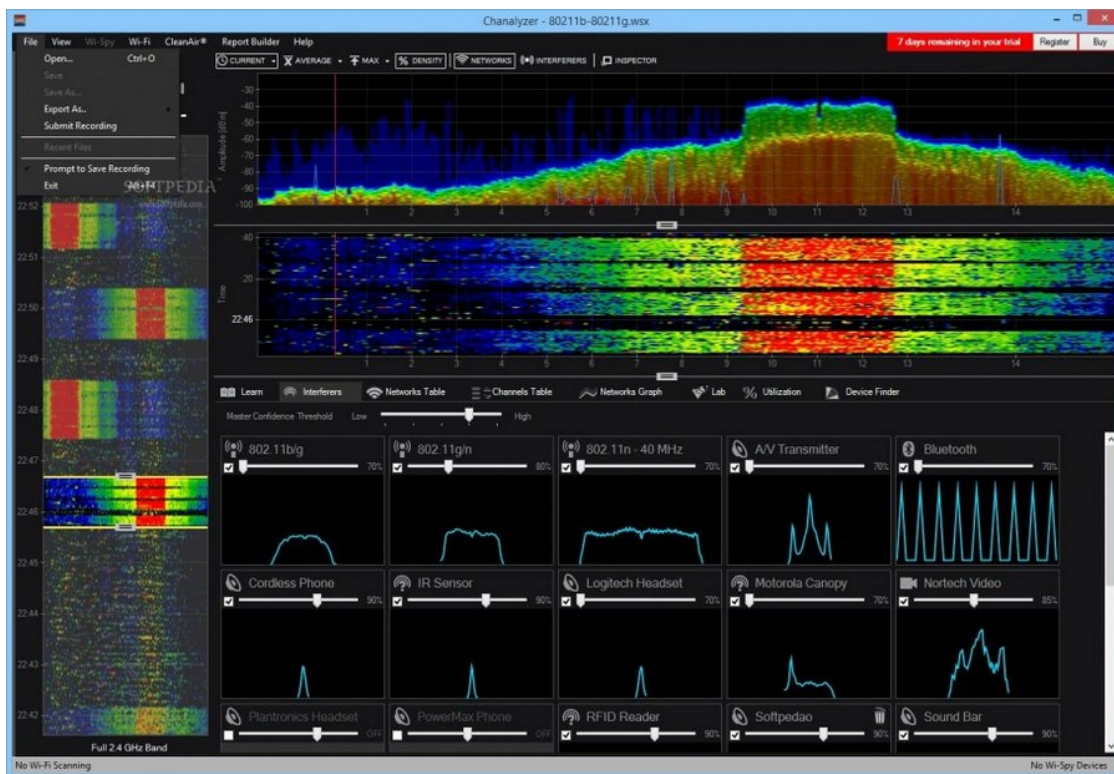


Рисунок 1 — Вывод частотного спектра в программе Cl analyzer

После того, как были рассмотрены особенности перехвата трафика в беспроводных сетях, необходимо перейти другим аспектам, связанным с разработкой информационной подсистемы.

### 1.3 Устройство и функционирование современных ИС

Системная архитектура организации определяет совокупность методологических, технологических и технических решений для обеспечения информационной поддержки деятельности организации, определяемой его

бизнес-архитектурой, и включает в себя архитектуру приложений, архитектуру данных и техническую архитектуру. [2, 25]

Архитектура приложений включается в себя:

1. прикладные системы, которые поддерживают исполнение бизнес-процессов;
2. интерфейсы взаимодействия прикладных систем между собой, внешними системами и источниками или потребителями данных;
3. средства, методы разработки и сопровождения приложений.

Архитектура данных включает в себя:

1. базы данных и хранилища данных;
2. системы управления базами данных или хранилищами данных;
3. правила и средства функционирования доступа к данным.

Техническая архитектура состоит из сетевой архитектуры и архитектуры платформ. Сетевая архитектура включает в себя:

1. локальные и территориальные вычислительные сети;
2. коммуникационные протоколы, системы адресации и сервисы, которые используются в сетях;
3. аварийные планы, направленные на обеспечение бесперебойной работы сети в условиях чрезвычайных обстоятельств.

Архитектура платформ включает в себя:

1. аппаратные части вычислительной техники, такие как серверы, рабочие станции, накопители и другое компьютерное оборудование;
2. операционные и управляющие системы, утилиты и офисные программные системы;
3. аварийные планы, направленные на обеспечение бесперебойной работы аппаратуры, в особенности серверов и баз данных в условиях чрезвычайных обстоятельств.

Значение архитектуры организации в современных условиях постоянно увеличивается за счёт обеспечения возможностей эффективного использования существующих технологий и эволюционного перехода к

новейшим технологиям. Фактически, информационная система организации является одним из главных средств управления изменениями. [2]

### **1.3.1 Основы современных операционных систем**

Операционная система — совокупность системного программного обеспечения, обеспечивающего управление аппаратной частью вычислительной системы, работу с файлами, ввод и вывод данных, а также выполнение сервисных программ пользователя [9].

Основные функции операционных систем:

1. Загрузка приложений в основную память;
2. Управление памятью вычислительной системы;
3. Управление процессами и потоками;
4. Управление доступом к внешним устройствам;
5. Защита программ и данных от злонамеренных действий пользователей или приложений и т.п. [9]

По числу одновременно выполняемых задач, операционные системы могут разделяться на два класса:

1. Однозадачные (MS-DOS);
2. Многозадачные (UNIX, Windows).

В свою очередь многозадачные операционные системы подразделяются на три типа:

1. ОС пакетной обработки (ОС ЕС);
2. ОС разделения времени (UNIX, VMS);
3. ОС реального времени (QNX, RT/11) [10].

В составе любой ОС различают супервизор, сервисы (службы) и приложения пользователя. Супервизор и сервисы ОС образуют её ядро.

Супервизор — центральная часть ОС, обеспечивающая приложениям координированный доступ к ресурсам вычислительной системы, таким как процессор, ОП и внешние устройства [9, 8].

В настоящее время различают следующие типы архитектур ядра ОС — монолитное ядро, модульное ядро, микроядро, экзоядро, наноядро и гибридное ядро [9, 9].

Монолитное ядро ОС является самым производительным ядром, так как его драйвера работают в пространстве ядра. Такое ядро понижает отказоустойчивость ОС, но повышает скорость её работы.

Модульное ядро — современная, усовершенствованная архитектура монолитного ядра. Все модули ядра работают в его адресном пространстве и могут пользоваться всеми функциями, предоставляемыми ядром [9, 10].

Микроядро — это следующий подход к размещению драйверов, при котором драйверы не работают в пространстве ядра. Драйверы работают с собственным независимым пространством и ядро производит переключение из одного пространства в другое.

Экзоядро — архитектура ядра ОС, предоставляющая лишь функции для взаимодействия между процессами и безопасного выделения и освобождения ресурсов вычислительной системы. Примером ОС с таким ядром является VM/370 [9,12].

Наноядро — архитектура ядра ОС, выполняющая лишь одну задачу — обработку аппаратных прерываний, генерируемых аппаратной частью вычислительной системы. [9, 12].

Гибридные ядра—это модифицированные микроядра, позволяющие с целью повышения быстродействия запускать «несущественные» части ОС в пространстве его ядра. Примером ОС с гибридным ядром являются Windows NT, Windows XP, Windows Vista, а также NetWare [9,13].

### **1.3.2 Основы современных систем управления базами данных**

#### **1.3.2.1 Модель «Сущность-связь»**

Модель данных «сущность-связь» очень тесно связана с реляционной моделью базы данных, используемой в различных СУБД. Для её построения используется три элемента: сущность, атрибут, связь.

Сущность — это совокупность атрибутов предметной области, которая имеет имя. Отображается на диаграмме в виде прямоугольника с именем внутри [11].

Атрибут — это поименованная характеристика сущности. Атрибут принимает значение из некоторого множества, которое называют типом данных. Множество допустимых значений атрибута называется доменом атрибута [12, 15].

Атрибуты бывают ключевыми и не ключевыми. Ключевой атрибут подчёркивается или отмечается слева (или справа) каким-то символом. На рисунке 2 это символ «#». [12]

Ключ сущности — это совокупность её ключевых атрибутов. Ключ всегда уникален.

Связь — это символ, который обозначает отношения между сущностями.

### **1.3.2.2 Реляционная модель**

Достоинствами реляционного подхода принято считать следующие свойства:

- реляционный подход основывается на небольшом числе интуитивно понятных абстракций, на основе которых возможно простое моделирование наиболее распространенных предметных областей;
- эти абстракции могут быть точно и формально определены; теоретическим базисом реляционного подхода к организации баз данных служит простой и мощный математический аппарат теории множеств и математической логики;
- реляционный подход обеспечивает возможность ненавигационного манипулирования данными без необходимости знания конкретной физической организации баз данных во внешней памяти [11].

Можно выделить следующие понятия реляционных баз данных: тип данных, домен, атрибут, кортеж, отношение, первичный ключ.

Тип данных — понятие, которое в реляционной модели данных полностью соответствует понятию типа данных в языках программирования. [11].

Обычно в современных реляционных базах данных допускается хранение символьных, числовых данных, специализированных числовых данных (валюта), а также специализированных «темпоральных» (дата, время, временной интервал). Кроме этого, в реляционных системах поддерживается возможность определения пользователями собственных типов данных [11].

Домен — понятие, которое более специфично для баз данных, хотя имеются аналогии с подтипами в некоторых языках программирования. В общем виде домен определяется путём задания некоторого базового типа данных, к которому относятся элементы домена и произвольного логического выражения, применяемому к элементу этого типа данных (ограничение домена). Элемент данных является элементом домена в том и только в том случае, если вычисление этого логического выражения дает результат истина. С каждым доменом связывается имя, уникальное имен всех доменов соответствующей базы данных [11].

Заголовком (или схемой) отношения  $r$  ( $H_r$ ) называется конечное множество упорядоченных пар вида  $\langle A, T \rangle$ , где  $A$  называется именем атрибута, а  $T$  обозначает имя некоторого базового типа или ранее определенного домена. По определению требуется, чтобы все имена атрибутов в заголовке отношения были различны [11].

Кортежем  $tr$ , соответствующим заголовку  $H_r$ , называется множество упорядоченных триплетов вида  $\langle A, T, v \rangle$ , по одному такому триплету для каждого атрибута в  $H_r$ . Третий элемент —  $v$  — триплета  $\langle A, T, v \rangle$  должен являться допустимым значением типа данных или домена  $T_m$  [11].

Телом  $B_r$  отношения  $r$  называется произвольное множество кортежей  $t$ .

### **1.3.2.3 Системы NOSQL**

Системы NOSQL — это базы данных, которые построены на парадигме распределения хранилищ  $\langle \text{ключ}, \text{значение} \rangle$ . Эти системы обеспечивают

доступ к неструктурированным данным посредством прямого чтения записей. Основные преимущества таких хранилищ — это ориентация на агрегаты данных: высокая масштабируемость, большое количество реплик, и, следовательно, высокая надёжность [12].

Эти базы являются сильно агрегатно-ориентированными, т.е. в качестве «значения» выступает агрегат. Выделяют четыре типа баз данных NoSQL:

- ключ-значение;
- документная;
- семейство столбцов;
- графовая.

В базе данных «ключ-значение» агрегат является непроницаемым для NoSQL. Чтение записи выполняется по ключу. Структуру агрегата видит только приложение. Поиск по индексу возможен для атрибутов, хранящихся в специальной заголовке записи (Riak). Примерами баз данных «ключ-значение» являются Riak, Redis, Amazon Dynamo.

В остальных базах данных агрегат является проницаемым для NoSQL, т. е. база данных видит структуру агрегата. Чтение записи выполняется по значениям полей агрегата. Поиск по индексу возможен для атрибутов, хранящихся в агрегате.

В документной базе данных в каждом агрегате есть поле уникального идентификатора, которое используется в приложении как ключ. В качестве агрегата могут выступать данные, структура которых определяется конкретной базой данных NoSQL. Например, в агрегате базы MongoDB хранится объект JSON (JavaScript Object Notation) / BSON (Binary JavaScriptObject Notation), который может содержать структуры

Базы данных «семейство столбцов» ориентированы на хранение данных по столбцам. Наборы столбцов в разных записях могут быть разными. Примерами этого типа баз данных являются BigTable, HBase, Cassandra.

В графовой базе данных данные представляются в виде графов: в виде вершин и связей между ними. Каждому узлу и связи соответствует запись

базы данных, в агрегате которой сохраняются свойства соответствующего элемента графа. Приложение читает свойства исходного узла и связи и осуществляет переход к следующей вершине. Пример базы данных - Neo4 [12].

### **1.3.3 Программные средства и платформы инфраструктуры информационных технологий организаций. Современные подходы и стандарты автоматизации организации**

В условиях цифровой трансформации ИТ-инфраструктура становится фундаментом функционирования и движущей силой развития любой организации [13].

Правильно спроектированная ИТ-инфраструктура обеспечивает основу для внедрения прикладных информационных систем, автоматизации бизнес-процессов, повышения эффективности работы организации [13].

ИТ-инфраструктура (инфраструктура информационных технологий) – единый комплекс взаимосвязанных программных, аппаратных и телекоммуникационных ресурсов организации, необходимых для обеспечения ее эффективного функционирования и выполнения имеющихся задач [13].

Процессы цифровой трансформации обуславливают ряд требований к ИТ-инфраструктуре независимо от специфики деятельности организации:

- Бесперебойная работа — отсутствие сбоев в работе ИТ-оборудования;
- Масштабируемость — чем больше в ИТ инфраструктуре автоматизировано, тем лучше её масштабируемость;
- Безопасность — чем сложнее инфраструктура и чем больше данных она включает, тем она уязвимее;
- Скорость изменений — изменение бизнес-процессов, внедрение новых технологий должно поддерживаться сервисами ИТ-инфраструктуры, которые способны реализовать изменения в кратчайшие сроки;



- Модульная структура — ИТ инфраструктура должны быть построена по модульному принципу, с возможностью изменять имеющиеся и добавлять новые сервисы и избавляться от устаревших;

- Стоимость владения;
- Прозрачность и управляемость.

Система требований к построению ИТ-инфраструктуры, управление жизненным циклом информационных систем и ИТ-продуктов регламентируется стандартами и иной нормативной технической документацией. В настоящее время наиболее известным общепризнанным стандартом в данной сфере является ITIL (IT Infrastructure Library) - Библиотека инфраструктуры информационных технологий [13].

ИТ-инфраструктура практически любой организации состоит из нескольких элементов, которые допустимо разделить на шесть групп:

- Клиентские устройства — офисное оборудование;
- Клиентские приложения и данные — рабочие приложения сотрудников организации.
- ОС и среда исполнения — основное ПО и сервисы (операционные системы серверов и рабочих станций, почтовые службы, службы печати, обновления, мониторинга, СУБД и прочее);
- Виртуализация — программное обеспечение для распределение и мониторинга виртуальных ресурсов;
- Серверы, СХД, сеть — аппаратное обеспечение центра обработки данных, коммутаторы прочее серверное оборудование;
- Размещение оборудования — серверная;

При традиционной (локальной) модели построения ИТ-инфраструктуры организация самостоятельно управляет указанными выше элементами. То есть организация приобретает оборудование, программное обеспечение, организует размещение аппаратных средств, содержит штат ИТ-специалистов для поддержки и администрирования и т.д.

Другая альтернатива, так называемая гибридная модель – это построение комплексной ИТ-инфраструктуры на базе облачных технологий. Облачные технологии обеспечивают через Интернет-соединение удаленный доступ к программно-аппаратному комплексу, включающему серверы, базы данных, хранилища и разнообразные приложения. Здесь возможны 3 варианта построения ИТ-инфраструктуры:

- IaaS (Infrastructure as a Service) – инфраструктура как услуга;
- PaaS (Platform as a Service) – платформа как услуга;
- SaaS (Software as a Service) – программное обеспечение как услуга [13];

Облачные технологии в ближайшей перспективе продолжают бурное развитие. Это связывают, прежде всего, с доминированием сервисной бизнес-модели взаимодействия с потребителем. Такие сервисные модели применяются как в частном, так и государственном секторе и идеально подходят под облачную ИТ-инфраструктуру. Одним из подходов создания сервисов на базе IaaS и PaaS является технология cloud native.

Cloud native – платформенный подход к созданию приложений на основе микросервисов.

Микросервисы представляют собой наборы небольших сервисов. Каждый такой сервис независимо от других реализует для пользователей определенную бизнес-возможность.

Каждый микросервис – отдельная программа, которую пишет группа программистов, специализирующихся на том или ином виде сервисов.

Архитектуру целостного программного приложения продумывают архитекторы-программисты на этапе проектирования.

Преимущества микросервисного подхода заключаются в микросервисной архитектуре, благодаря чему любой сервис можно развернуть, обновить, масштабировать или перезапустить независимо от других сервисов приложения. Приложения, состоящие из нескольких микросервисов, можно обновлять, не причиняя неудобств пользователям [13].

## 1.4 Языки современных бизнес-приложений

Существует несколько индексов, которые показывают популярность языков программирования. У них разные подходы к оценке, и все они не на 100 процентов объективны. Однако основные тенденции с помощью языков выявить можно.

Список IEEE Spectrum примечателен тем, что ранжирует языки в контексте интернета, предприятия, мобильных устройств и встроенных приложений. Организация создает свои рейтинги, комбинируя 11 показателей из восьми источников, включая Stack Overflow, GitHub и Twitter. Теоретически это создает ранжирование языков в соответствии как с использованием (например, количеством репозиторий GitHub), так и с обсуждением (например, количеством сообщений Reddit)[3].

На основе этого рейтинга можно выделить 15 самых популярных языков бизнес-приложений.

- Python — интерпретируемый, объектно-ориентированный высокоуровневый язык программирования с динамической семантикой. [4];
- Java — объектно-ориентированный язык, лёгкий в изучении и позволяющий создавать программы, которые могут исполняться на любой платформе без каких-либо доработок [5];
- C — является компилируемым статически типизированный язык программирования общего назначения;
- C++ — это объектно-ориентированный язык программирования, хорошо известный своей эффективностью, экономичностью и переносимостью. Указанные преимущества C++ обеспечивают высокое качество разработки программного продукта. [6];
- JavaScript — основной язык в web-разработке, так как все современные web браузеры включают в себя его интерпретатор. Развитие экосистемы языка, обеспечило ему широкую применимость, т.к. существует

масса фреймворков, позволяющих использовать JavaScript-код на различных платформах: от серверов до микроконтроллеров [7];

- C# — объектно-ориентированный язык программирования. Разработан группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework;

- R — язык для статической обработки данных, главный конкурент Python для тех, кто занимается статистикой и анализом данных. Его используют в социальных и экономических науках для поиска причинно-следственных связей, сравнения выборок, создания наглядных отчётов и графиков;

- Go — многопоточный компилируемый язык программирования, который был разработан внутри компании Google. Считает языком общего назначения;

- HTML — стандартизированный язык разметки документов для просмотра веб-страниц в браузере;

- Swift — это компилируемый язык программирования с открытым исходным кодом от компании Apple. Предназначен для разработки приложений для MacOS и iOS, реже применяется в других проектах;

- Arduino — язык программирования, который основан на C/C++ и скомпонован с библиотекой AVR Libc и позволяет использовать любые ее функции, чаще всего используется для программирования устройств на микроконтроллерах;

- Matlab — высокоуровневый интерпретируемый язык программирования с интегрируемой средой разработки и вместе с пакетом прикладных программ, выполнения инженерных и математических расчетов, работы с матричными базами данных, визуализации;

- PHP — скриптовый язык программирования, который имеет открытый исходный код. Первоначально язык был предназначен для

разработки веб-приложений, но в процессе обновления стал языком общего назначения;

- Dart — язык программирования общего назначения от компании Google, который предназначен прежде всего для разработки веб-приложений (как на стороне клиента, так и на стороне сервера) и мобильных приложений;
- SQL — это структурированный язык запросов, созданный для того, чтобы получать из базы данных необходимую информацию;

## **1.5 Инструменты и методы прототипирования пользовательского интерфейса**

Прототипирование является ключевой частью UI и UX. Прототипы возможно создавать как с высокой, так и с низкой точностью. Прототипы дают возможность опробовать функцию, сайт или приложение. Цель прототипирования — протестировать идею до того, как она будет полностью реализована [15].

Adobe XD — один из самых популярных инструментов среди UI/UX-дизайнеров. Программа работает в качестве универсальной платформы для создания вайрфреймов, дизайна сайтов, мобильных приложений, голосовых интерфейсов и т.д.

Figma — это очень простой в использовании редактор перетаскивания. Он предназначен для создания каркасов и практических прототипов. [15].

Webflow — это хорошо известный метод прототипирования для разработки веб-сайтов с zero-code. [15].

Sketch — одно из лучших приложений, доступных сегодня для создания всех форм пользовательских интерфейсов. Это как более сложная и широко поддерживаемая версия Figma. Доступно только на iOS [15].

Axure RP — это инструмент, который сочетает в себе эффективные инструменты проектирования, SVG-импорт, интеграцию Sketch и Adobe XD с прототипированием мирового уровня. [15].

## **1.6 Современные методики тестирования разрабатываемых ИС**

Тестирование программного обеспечения является одним из основных процессов жизненного цикла программного обеспечения. Процессы ЖЦ ПО определены в международном стандарте ISO/IEC 12207: 1995. [1]

Существующие на сегодняшний день методики тестирования ПО не позволяют однозначно и полностью устранить все дефекты и ошибки и установить корректность функционирования программного продукта. Поэтому, все существующие методы тестирования действуют в рамках формального процесса проверки исследуемого или разрабатываемого программного продукта.

Такой процесс формальной проверки или верификации может доказать, что дефекты отсутствуют, с точки зрения используемого метода.

Тестирование программного обеспечения — это попытка определить, выполняет ли программа то, что от неё ожидают.

Существуют следующие методики тестирования программного обеспечения

### **1.6.1 Методология тестирования «белого ящика»**

Тестирование «белого ящика» — это методология тестирования, которая учитывает внутренние механизмы системы или компонента. Так же этот метод называют структурным тестированием. [1]

При тестировании «белого ящика»:

- известна внутренняя структура программы;
- изучаются внутренние элементы программы и связи между ними.

Объектом такого тестирования является не внешнее, а внутреннее поведение программы. Проверятся корректность построение элементов программы и правильность их взаимодействия друг с другом. Обычно анализируются управляющие связи, реже — информационные.

### **1.6.2 Метод тестирования базового пути**

Тестирование базового пути — это способ, который основан на принципе «белого ящика». Автор этого метода — Том МакКейб.

Способ тестирования базового пути даёт возможность:

- получить оценку комплексной сложности программы;
- использовать эту оценку для определения необходимого количества тестовых вариантов.

Тестовые варианты разрабатываются для проверки базового множества путей (маршрутов) в программе. Они гарантируют однократное выполнение каждого оператора программы при тестировании. [1]

### **1.6.3 Способы тестирования условий**

Цель этого семейства методов тестирования – построение тестовых вариантов для проверки логических условий программы. При этом желательно обеспечить охват операторов из всех ветвей программы. [1]

### **1.6.4 Тестирование ветвей и операторов отношений**

Метод тестирования ветвей и операторов отношений обнаруживает ошибки ветвления и операторов отношения в условии, для которого выполняются следующие ограничения:

- все булевы переменные и операторы отношения входят в условие только по одному разу;
- в условии нет общих переменных.

### **1.6.5 Метод потоков данных**

В методе потоков данных учитывается информационная структура программы. Работу любой программы можно рассматривать как обработку потока данных, передаваемых от входа в программу к её выходу [1].

### **1.6.6 Тестирование циклов**

Цикл — наиболее распространённая конструкция алгоритмов, которые реализуются в программном обеспечении. Тестирование циклов производится по принципу «белого ящика», при проверке циклов, в первую очередь обращают внимание на правильность конструкций циклов. Различают четыре типа циклов: простые, объединённые, неструктурированные, вложенные.

Для проверки простых циклов с числом повторений  $n$  может использоваться один из следующих наборов тестов:

1. прогон всего цикла;
2. только один проход цикла;
3. два прохода цикла;
4.  $m$  проходов цикла, где  $m < n$ ;
5.  $n - 1$ ,  $n$ ,  $n + 1$  проходов цикла.

С увеличением уровня вложенности циклов число возможных путей резко возрастает. Это приводит к нереализуемому числу тестов. Для сокращения числа тестов применяется специальная методика, в которой используются такие понятия, как объемлющий и вложенный цикл.



## **2 ПРОЕКТИРОВАНИЕ ПОДСИСТЕМЫ АНАЛИЗА ТРАФИКА В БЕСПРОВОДНЫХ СЕТЯХ**

### **2.1 Анализ бизнес-процессов организации**

Кинотеатр «Спартак» территориально располагается в г. Воронеж, пл. Ленина д 13. Структура организации насчитывает семь основных отделов, а именно:

1. Отдел кинопоказа – структурное подразделение организации, занимается всеми возможными вопросами кинопоказа;
2. Бухгалтерия — штатно-структурное подразделение хозяйствующего субъекта, предназначенное для аккумулирования данных о его имуществе и обязательствах;
3. Рекламный отдел — структурное подразделение организации, которое занимается продвижением кинопоказа и ресторанного направления и сайта организации;
4. Отдел направления общественного питания – занимается руководством ресторанного направления организации;
5. Служба безопасности — структурное подразделение организации, предназначенное для организации и контроля над выполнением мероприятий по обеспечению защиты объекта, а также для выполнения ряда других специальных функций;
6. IT отдел — структурное подразделение организации, которое обеспечивает штатную работу и модернизацию парка компьютерной техники, сети и программного обеспечения, следят за безопасностью компьютерной сети, помимо этого сопровождают мероприятия, где требуется применение аудио оборудования;
7. Административно-хозяйственный отдел. — структурное подразделение организации, которое занимается хозяйственным

обслуживанием, содержанием в надлежащем состоянии внутренних помещений здания, созданием комфортных условий труда.

Структурная схема организации представлена на рисунке 2.

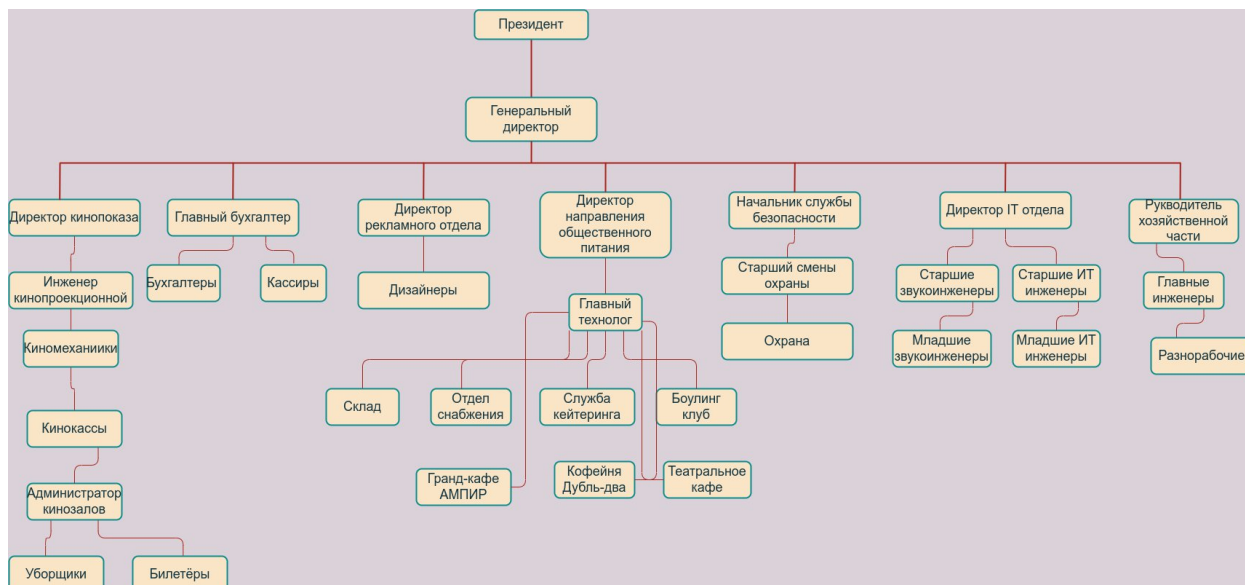


Рисунок 2 — Структурная схема организации

Бухгалтерия является самостоятельным структурным подразделением предприятия и подчиняется непосредственно генеральному директору и президенту организации.

В бухгалтерии располагается 9 персональных компьютеров, один из которых является сервером лицензий 1С. Помимо этого имеется три сетевых принтера, которые подключены к локальной сети. Беспроводная сеть в данном отделе не используется.

Отдел кинопоказа является самостоятельным структурным подразделением и подчиняется непосредственно генеральному директору и президенту организации. Отдел кинопоказа активно взаимодействует с рекламным отделом с целью обмена служебной информации о предстоящих мероприятиях, связанных с кинопоказом.

Отдел кинопоказа имеет три персональных компьютера подключенных в локальную сеть, один из них – ноутбук, который подключается посредством беспроводной сети к локальной сети организации.

Рекламный отдел является самостоятельным структурным подразделением, которое в свою очередь подчиняется генеральному директору и президенту организации. Помимо отдела кинопоказа, отдел активно взаимодействует с отделом направления общественного питания. Пример такого взаимодействия может являться разработка нового дизайна меню для «Гранд-кафе Амбир».

Отдел рекламы имеет три персональных ПК, которые используются в основном в дизайнерских целях, подключены по проводной сети.

Отдел направления общественного питания является самостоятельным структурным подразделением, которое в свою очередь подчиняется генеральному директору и президенту организации. В своём управлении имеет четыре пункта общественного питания, а именно: «Боулинг клуб», «Гранд-кафе Амбир», «Кофейня Дубль-два» и «Театральное кафе». Активно взаимодействует в рекламным и IT отделом.

В отделе направления общественного питания имеется 10 персональных компьютеров, 8 из них – моноблоки, имеющие программное обеспечение R-keeper, 7 банковских терминалов, 4 точки доступа Wi-Fi, которые взаимодействуют с локальной сетью организации.

Служба безопасности является самостоятельным структурным подразделением, которое в свою очередь подчиняется генеральному директору и президенту организации. Активно взаимодействует с IT отделом, с целью управления системой контроля доступа и по вопросам взаимодействия с системой видео наблюдения. Примером такого взаимодействия является помощь в записи с камер видео наблюдения на носитель информации с целью передачи её в правоохранительные органы.

Служба безопасности имеет 3 персональных компьютера, которые предназначены для просмотра записи с камер видео наблюдения в реальном времени.

IT является самостоятельным структурным подразделением и подчиняется непосредственно генеральному директору и президенту

организации. Обеспечивает штатную работу и модернизацию парка компьютерной техники, сети и программного обеспечения, следят за безопасностью компьютерной сети, помимо этого сопровождают мероприятия, где требуется применение аудио оборудования. В разной степени взаимодействуют со всеми отделами предприятия для оказания технической поддержки пользователей.

Служба безопасности имеет 4 персональных компьютера, один ноутбук с целью диагностики неисправностей. Так же ответственна за штатную работу сети и различных серверов организации.

Административно-хозяйственный отдел является самостоятельным структурным подразделением и подчиняется непосредственно генеральному директору и президенту организации. Отдел активно взаимодействует со всеми подразделениями организации, касаясь обслуживания, содержания в надлежащем состоянии внутренних помещений здания, созданием комфортных условий труда.

Отдел имеет три персональных компьютера. Два из них используются главными инженерами для написания разного рода отчётов, один используется в роли сервера для автоматического забора показаний со счётчиков тепла.

## **2.2 Основы информационной безопасности организации**

Информация может представлять некоторую ценность. Например, описание некоторого, нового, ещё не запатентованного изобретения будет ценной для изобретателя. Эта же информация будет ценной для организации, которые могут производить, продавать или использовать это изобретение в своих целях [14].

Защита информации — это комплекс мер, направленных на обеспечение защиты информации от неправомерного доступа, уничтожения, модифицирования, блокирования, копирования, предоставления,

распространения, а также от иных неправомерных действий в отношении такой информации; соблюдение конфиденциальности информации ограниченного доступа; реализацию права на доступ к информации [14].

Меры по обеспечению информационной безопасности

Меры по обеспечению информационной безопасности делятся на следующие основные группы:

- нормативно-правовые и научные — к данной группе мер относятся законодательные и иные нормативно-правовые акты разного уровня, определяющие понятия, определения, требования, разрешения, запреты и ответственность за их соблюдение или нарушение, действующие в области информационной безопасности и зафиксированные юридически [14];
- административные — к этой группе мер относятся создание и функционирование различных подразделений, занимающихся обеспечением информационной безопасности. Это может быть и один ИБ-специалист, занимающийся защитой информации на небольшом предприятии, и полномасштабные государственные структуры, решающие похожие задачи, только уже на другом уровне: Роскомнадзор, подразделения МВД и ФСБ, занимающиеся компьютерными преступлениями;
- организационно-технические — эта группа мер представлена решениями о том, как должны функционировать элементы защищаемой информационной системы и какие работы должны производиться в системе постоянно, периодически и в случае возникновения той или иной ситуации [14];
- программно-технические меры — к этой группе мер относится использование специальных программных и аппаратных средств, применяемых для решения задач защиты информации. Примерами таких средств можно назвать антивирусы, межсетевые экраны, генераторы паролей, программы для шифрования и дешифрования, и т.п.

## **2.3 Предложения по внедрению и модернизации программных средств**

В отделе направления общественного питания имеется 10 персональных компьютеров, 8 из них – моноблоки, имеющие программное обеспечение R-keeper, 7 банковских терминалов, 4 точки доступа Wi-Fi, которые взаимодействуют с локальной сетью организации.

В организации банковские терминалы обрабатывают транзакции путём подключения к точкам доступа Wi-Fi. С данными терминалами не редко происходят какие-либо проблемы, в частности, на этапе оплаты. Вероятно, данные проблемы связаны с беспроводной сетью. Для того, чтобы попытаться исключить проблему именно с беспроводной сетью, было предложено разработать некоторую информационную подсистему для анализа трафика в беспроводных сетях с целью выявления аномалий в беспроводной сети. Данную подсистему, в свою очередь можно связать с существующими службами мониторинга в организации или настроить отдельную систему мониторинга. С экономической точки зрения данное решение является абсолютно бесплатным, что не требует дополнительных согласований денежных затрат.

## **2.4 Анализ существующих решений для обработки сетевых пакетов**

Особенности перехвата сетевого трафика в беспроводной локальной вычислительной сети состоит в том, что частотный спектр беспроводной связи распределён в среде передачи информации по отдельным физическим радиоканалам. В отличие от проводных сетей, где каждый клиент подключается к коммутатору с помощью отдельного кабеля, среда передачи данных по беспроводным сетям является общей для всех клиентов.

Будут рассмотрены ряд средств, которые используются на сегодняшний день для анализа сетевых пакетов.

Wireshark —анализатор сетевых пакетов. Анализатор сетевых пакетов максимально подробно представляет захваченные пакетные данные [3].

tcpdump — утилита, которая входит в состав большинства Unix-систем и позволяет перехватывать и отображать сетевой трафик [2].

CloudShark — инструментальное средство, которое было разработано компанией QA Cafe, используется для хранения, индексирования и сортировки перехваченных пакетов. Лицензия является платной. Оно позволяет пометить перехваченные пакеты для быстрого нахождения и снабжать их комментариями и даже предоставляет некоторые средства для анализа пакетов, аналогичные тем, что имеются в Wireshark. Данное средство актуально, если в организации ведётся крупная библиотека образцов перехвата пакетов.

WireEdit — данное программное обеспечение может пригодится, чтобы протестировать применяемую систему обнаружения вторжений или разработку сетевого программного обеспечения. Средство позволяет редактировать значения в отдельных полях сетевых пакетов.

Cain &Abel — программное обеспечение, которое относится к одному из лучших инструментальных средств Windows для заражения ARP-кэша.

Scapy — библиотека на языке Python, с помощью которой можно создавать пакеты и манипулировать ими по сценариям, выполняемым из командной строки в своей среде. Данная библиотека является наиболее эффективной и удобной для обработки сетевых пакетов.

TraceWrangler — это набор инструментов для работы с файлами трассировки, работающий в Windows (и в Linux с использованием Wine). Поддерживает новый формат файла PCAPng, который является стандартным форматом файла, используемым Wireshark. Наиболее известный вариант использования для TraceWrangler — это простая очистка/анонимизация файлов трассировки пакетов, которая удаляет или заменяет конфиденциальные данные из состава сетевых пакетов.

TcpReplay — программное средство, которые выполняет после перехвата сетевых пакетов их повторную передачу, например, чтобы посмотреть, как на них будут реагировать сетевые устройства.

NetworkMiner — инструментальное средство, которое применяется для судебной экспертизы сетей. Помимо перехвата сетевых пакетов, NetworkMiner способен выполнять синтаксический анализ пакетов из файла перехвата. В этом случае NetworkMiner выбирает файл перехвата формата PCAP и разбирает его содержимое по типам операционных систем, обнаруживаемых в сеансах связи между хостами. NetworkMiner позволяет даже извлекать переданные файлы непосредственно из перехваченного трафика [1].

CapTipper — инструментальное средство, которое специально предназначено для специалистов по безопасности сетей, занимающихся анализом сетевого трафика по протоколу HTTP. CapTipper предоставляет богато оснащенную среду командной оболочки, которая дает пользователю возможность исследовать отдельные диалоги в интерактивном режиме в поисках переадресации, файловых объектов и зловредного содержимого. В CapTipper предоставляются также удобные средства для взаимодействия с раскрываемыми данными, включая возможность извлекать данные из архивов типа gzip и передавать хеш-суммы файлов в службу VirusTotal.

ngrep — утилита командной строки Linux, которая позволяет находить конкретные данные в перехваченных пакетах. Обычно используют, когда фильтры перехвата и отображения оказываются не пригодными или слишком сложными для выявления нужных данных.

libpcap — переносимая библиотека C/C++ для перехвата сетевого трафика. Wireshark, tcpdump и большинство других приложений для анализа пакетов в какой-то степени используют данную библиотеку.

Npcap — это библиотека из проекта под названием Nmap Project, основанная на библиотеке WinPcap/libpcap и предназначена для анализа пакетов в Windows.



hping — инструментальное средство для анализа пакетов. Служит для обработки, правки и передачи пакетов в режиме командной строки. Данное средство поддерживает различные протоколы.

Python — интерпретируемый язык программирования. По мере совершенствования в анализе пакетов непременно пользователь может столкнуться, когда для удовлетворения некоторых потребностей может не оказаться подходящих автоматизированных средств. В подобных случаях необходим язык Python, который позволяет создавать инструментальные средства, способные выполнять не мало интересных действий над пакетами.

Все перечисленные выше решения являются отдельными инструментами и не представляют из себя целостной подсистемой для анализа и хранения сетевого трафика. Их освоение требует достаточно количества времени и высокой квалификации сотрудников. Поэтому необходимо создать способ, при котором не будет необходимости запоминать сложные команды с большим количеством ключей, но при этом собирать сетевой трафик так, чтобы потом его можно было анализировать в Elastic-Stack.

## **2.5 Режимы работы адаптера беспроводной связи**

Перед тем как приступить к анализу пакетов в беспроводной сети, необходимо рассмотреть различные режимы работы адаптера беспроводной связи.

1. Управляемый режим (Managed Mode) — данный режим применяется в том случае, если клиент беспроводной сети подключается непосредственно к самой точке беспроводного доступа. В подобных случаях программный драйвер, связанный с адаптером беспроводной связи, использует точку беспроводного доступа для управления всем процессом обмена данными по беспроводной сети [8].

2. Режим прямого подключения (Ad hoc mode) — данный режим применяется в том случае, если организована беспроводная сеть, в которой

устройства подключаются непосредственно друг у другу. В этом режиме два клиента беспроводной сети, которым требуется обмениваться данными друг с другом, разделяют обязанности, которые обычно возлагаются на точку беспроводного доступа [8].

3. Ведущий режим (Master mode) — некоторые адаптеры поддерживают также ведущий режим. В этом режиме адаптеру беспроводной связи разрешается работать вместе с программным драйвером, чтобы компьютер, на котором установлен этот адаптер действовал в качестве точки беспроводного доступа для других устройств [8].

4. Режим текущего контроля (Monitor mode) — это наиболее важный режим для перехвата и анализа пакетов. Режим текущего контроля применяется в том случае, когда клиенту беспроводной сети требуется остановить передачу и приём данных и вместо этого прослушивать пакеты, распространяемые в эфире. Для перехвата пакетов в анализаторе сетевых пакетов Wireshark адаптер беспроводной связи вместе с программным драйвером должен поддерживать режим текущего контроля, называемый так же режимом RFMON, то есть режимом радиочастотного контроля [8].

Большинство пользователей применяют адаптеры беспроводной связи только в режиме прямого подключения или управляемом режиме.

## **2.6 Разработка прототипа ИС в соответствии с требованиями**

Информационная подсистема исследования трафика в беспроводных сетях согласно требованиям, должна уметь выполнять следующие действия:

1. Перехватывать сетевые пакеты
2. Хранение сетевых пакетов
3. Обработка и классификация сетевых пакетов в зависимости от содержимого этого пакета с целью последующего анализа для выявления аномалий сетевого трафика.

Для того чтобы перехватывать сетевые пакеты будет использоваться инструмент под названием Tshark. Данный инструмент является консольным клиентом Wireshark и позволяет делать всё то, что делает Wireshark только без графической оболочки.

Для обработки и классификации сетевых пакетов в зависимости от содержимого этого пакета с целью последующего анализа для выявления аномалий сетевого трафика будет использоваться Elastic Stack.

Elastic Stack — это Kibana, Logstash, Beats, X-Pack и Elasticsearch. Ядром Elastic Stack выступает поисковая система Elasticsearch, которая предоставляет возможности для хранения, поиска и обработки данных. Утилита Kibana, которую также называют окном в Elastic Stack, является отличным средством визуализации и пользовательским интерфейсом для Elastic Stack. Компоненты Logstash и Beats позволяют передавать данные в Elastic Stack.

Так как Elasticsearch умеет взаимодействовать с файлами формата JSON, а Wireshark умеет записывать перехваченные сетевые пакеты и сохранять их в файле формата JSON, было предложено сделать подсистему анализа пакетов в беспроводных сетях, которая бы автоматически опраивляла данные в ELK Stack для хранения и последующего анализа сетевого трафика в беспроводной сети. Примерная схема такого решения представлена на рисунке 3.

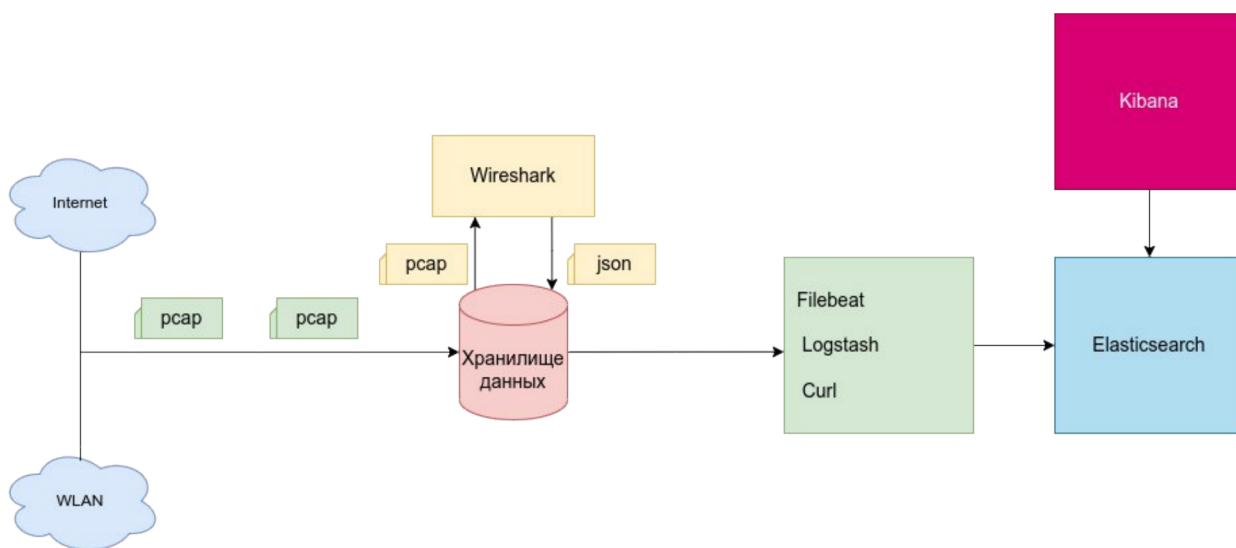


Рисунок 3 — Планируемая структурная схема проекта анализа трафика в беспроводной сети

### 2.6.1 Установка и настройка Elastic Stack

Для установки и настройки прототипа информационной подсистемы анализа трафика необходимо воспользоваться программным продуктом виртуализации для операционных систем. В данном случае таким программным продуктом будет выступать VirtualBox от компании Oracle.

Установка Elastic Stack будет производиться на операционной системе Debian 11 с кодовым именем «bullseye».

После установки операционной системы к ней следует подключиться по протоколу SSH, чтобы получить традиционный доступ к консоли. Удачное подключение к консоли по протоколу SSH представлено на рисунке 4.

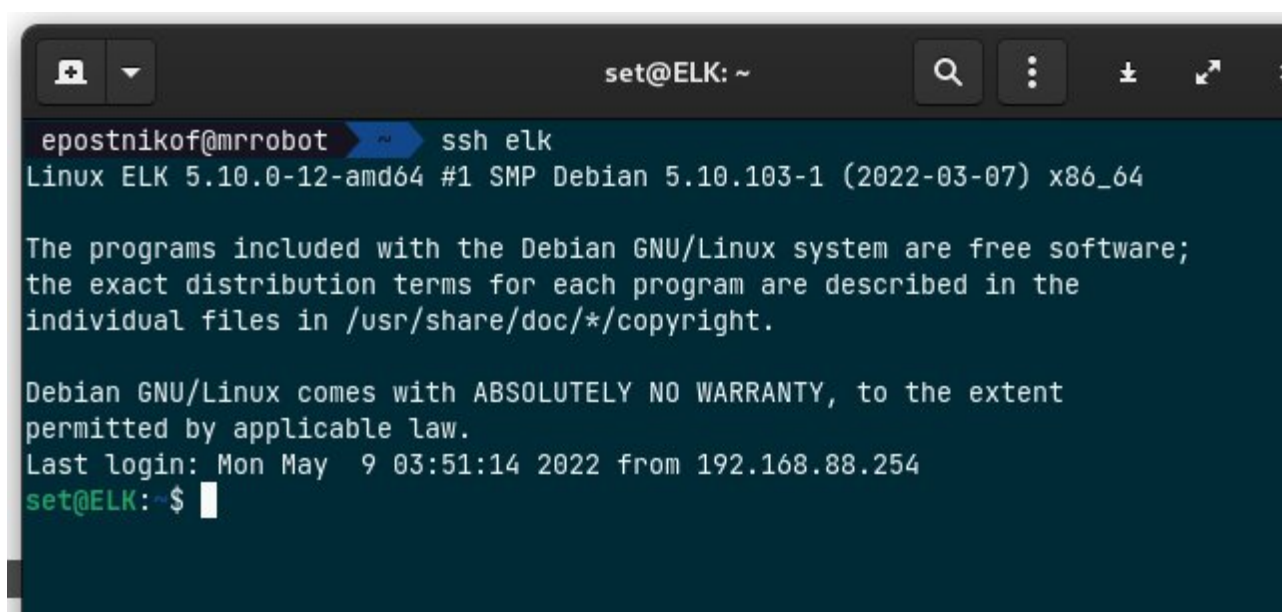


Рисунок 4 — удачное подключение по SSH протоколу

Минимальные системные требования, которые нужны для нормальной работы Elastic Stack представлены в таблице 1.

Таблица 1 — системные требования для Elastic Stack

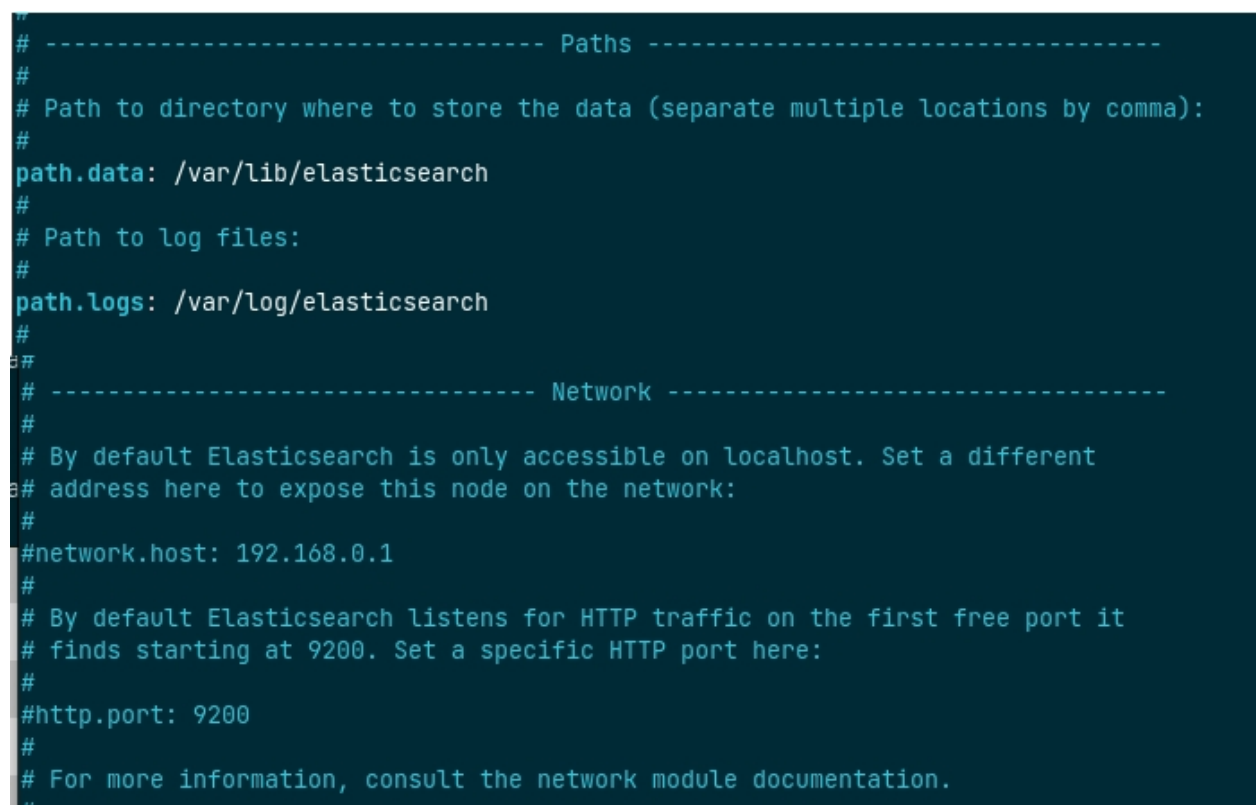
	Минимальные	Рекомендуемые
CPU	2	4+
Memory	6 Gb	8+ Gb
Disk	10 Gb	10+ Gb

Последовательность команд для установки Elastic Stack представлена в приложении В.

После совершения всех шагов установки необходимо настроить компоненты стека.

### 2.6.1.1 Настройка Elasticsearch

Для настройки Elasticsearch необходимо перейти в режим редактирования конфигурационного файла от пользователя root, который располагается по пути: «/etc/elasticsearch/elasticsearch.yml». В данном файле стоит обратить внимание на секцию «Paths» в которой указан путь сохранения лог файлов и на секцию «Network». В данной секции необходимо принять к сведению используемый сетевой порт по умолчанию или указать другой сетевой порт, на котором будет принимать запросы Elasticsearch (рисунок 5).



```
# ----- Paths -----
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /var/lib/elasticsearch
#
# Path to log files:
#
path.logs: /var/log/elasticsearch
#
# ----- Network -----
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
#network.host: 192.168.0.1
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
#http.port: 9200
#
# For more information, consult the network module documentation.
#
```

Рисунок 5 — Внешний вид файла конфигурации Elasticsearch

В случае редактирования файла конфигурации, необходимо перезапустить Elasticsearch с помощью команды:

```
«sudo systemctl restart elasticsearch»
```

После перезапуска нужно убедиться, что служба запущена корректно с помощью команды:

```
«sudo systemctl status elasticsearch»
```

После настройки elasticsearch можно приступить к настройке Kibana

### **2.6.1.2 Настройка Kibana**

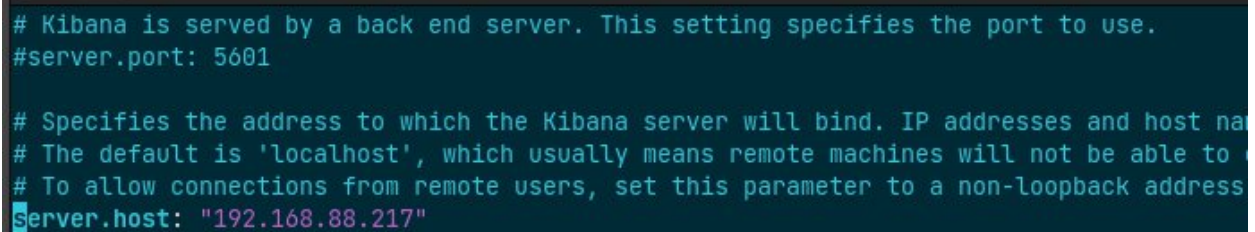
Для настройки Elasticsearch необходимо перейти в режим редактирования конфигурационного файла от пользователя root, который располагается по пути: «/etc/kibana/kibana.yml».

В данном файле (рисунок 6) стоит обратить внимание, что Kibana принимает запросы на порту 5601, и что сам веб сервис будет располагаться по IP адресу: «192.168.88.217». IP адрес необходимо сменить на реальный адрес машины. После того, как конфигурация выставлена верно необходимо перезапустить Kibana командой:

```
sudo systemctl restart kibana
```

После перезапуска проверить доступность сервиса с помощью команды:

```
sudo systemctl restart kibana
```



```
# Kibana is served by a back end server. This setting specifies the port to use.
#server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host na
# The default is 'localhost', which usually means remote machines will not be able to
# To allow connections from remote users, set this parameter to a non-loopback address
Server.host: "192.168.88.217"
```

Рисунок 6 — Внешний вид файла конфигурации Kibana

После того как был настроен Kibana, необходимо приступить к настройке Logstash.

### **2.6.1.3 Настройка Logstash**

Так как Logstash представляет из себя программу, которая фильтрует поступающие в нее записи, то конфигурация этих фильтров для каждого

отдельного случая будет индивидуальной. Файлы конфигурации Logstash располагаются в по пути: «/etc/logstash/conf.d». Конфигурация Logstash состоит из трех разделов:

1. Input (рисунок 7);
2. Filter (рисунок 8);
3. Output (рисунок 9).

```
input {
  beats {
    port => 5400
    ssl => true
    ssl_certificate_authorities => ["/etc/elk-certs/elk-ssl.crt"]
    ssl_certificate => "/etc/elk-certs/elk-ssl.crt"
    ssl_key => "/etc/elk-certs/elk-ssl.key"
    ssl_verify_mode => "force_peer"
  }
}
```

Рисунок 7 — Раздел «Input» файла конфигурации Logstash

Конфигурация на рисунке 7 обозначается, что:

- данные ожидаются на порту 5400,
- используется ssl;
- местоположение корневого сертификата;
- путь к сертификату и ключу,
- ssl обязателен.

```
filter {
  # Drop Elasticsearch Bulk API control lines
  if ([message] =~ "{\\"index") {
    drop {}
  }

  json {
    source => "message"
    remove_field => "message"
  }
}
```

Рисунок 8 — Раздел «Filter» в конфигурационном файле Logstash

У раздела «Filter» присутствует достаточно большое количество модификаторов, которые позволяют собирать данные из различных источников, преобразовывать их и отправлять в нужное место назначения.

```
output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "wireless-packets-%{+YYYY.MM.dd}"
    document_type => "pcap_file"
    manage_template => false
  }
}
```

Рисунок 9 — секция «Output» конфигурационного файла Logstash

Секция Output содержит в себе информацию о том где расположен Elasticsearch и индексы, с которыми эти данные будут выгружаться. С полным конфигурационным файлом, который направлен на обработку данных из сетевых пакетов, можно ознакомиться в приложении Г.

#### 2.6.1.4 Настройка Filebeat

Настройка Filebeat производится на компьютере, с которого планируется собирать данные. В данном случае это будет компьютер, на котором будут анализироваться пакеты. Команды для установки Filebeat находятся в приложении В. Файл конфигурации находится по пути: «/etc/Filebeat/filebeat.yml». Содержимое файла конфигурации для сбора данных сетевых пакетов представлено на рисунке 10.

В конфигурационном файле Filebeat есть две секции:

1. filebeat.inputs
2. output.logstash

В секции «filebeat.inputs» находятся данные о местоположении логов, которые должен считывать filebeat. В данном случае задано местоположение логов nginx и файлов с пакетами формата json.



В секции «output.logstash» описаны настройки для подключения logstash. В данном случае подключение происходит посредством SSL сертификата, который был предварительно сгенерирован для Elastic Stack.

```
filebeat.inputs:
- type: log
  paths:
    - /var/log/nginx/*.log
  exclude_files: ['\*.gz$']

- type: log
  paths:
    - /var/files/packets/*.json
  document_type: "pcap_file"
  json.keys_under_root: true

processors:
- drop_event:
  when:
    equals:
      index._type: "pcap_file"

output.logstash:
  hosts: ["logstash-server.local:5400"]
  ssl.certificate_authorities: ["/etc/elk-certs/elk-ssl.crt"]
  ssl.certificate: "/etc/elk-certs/elk-ssl.crt"
  ssl.key: "/etc/elk-certs/elk-ssl.key"
```

Рисунок 10 — Конфигурация Filebeat

### 2.6.2 Анализ прецедентов

Для описания поведения и определения функциональных требований разрабатываемой подсистемы существует диаграмма прецедентов, состоящая из актёров, которые взаимодействуют с системой посредством вариантов использования.

Диаграмма прецедентов представлена на рисунке 11.

В таблице 2 представлено описание действующих лиц, а в таблице 3 приложено краткое описание вариантов использования.

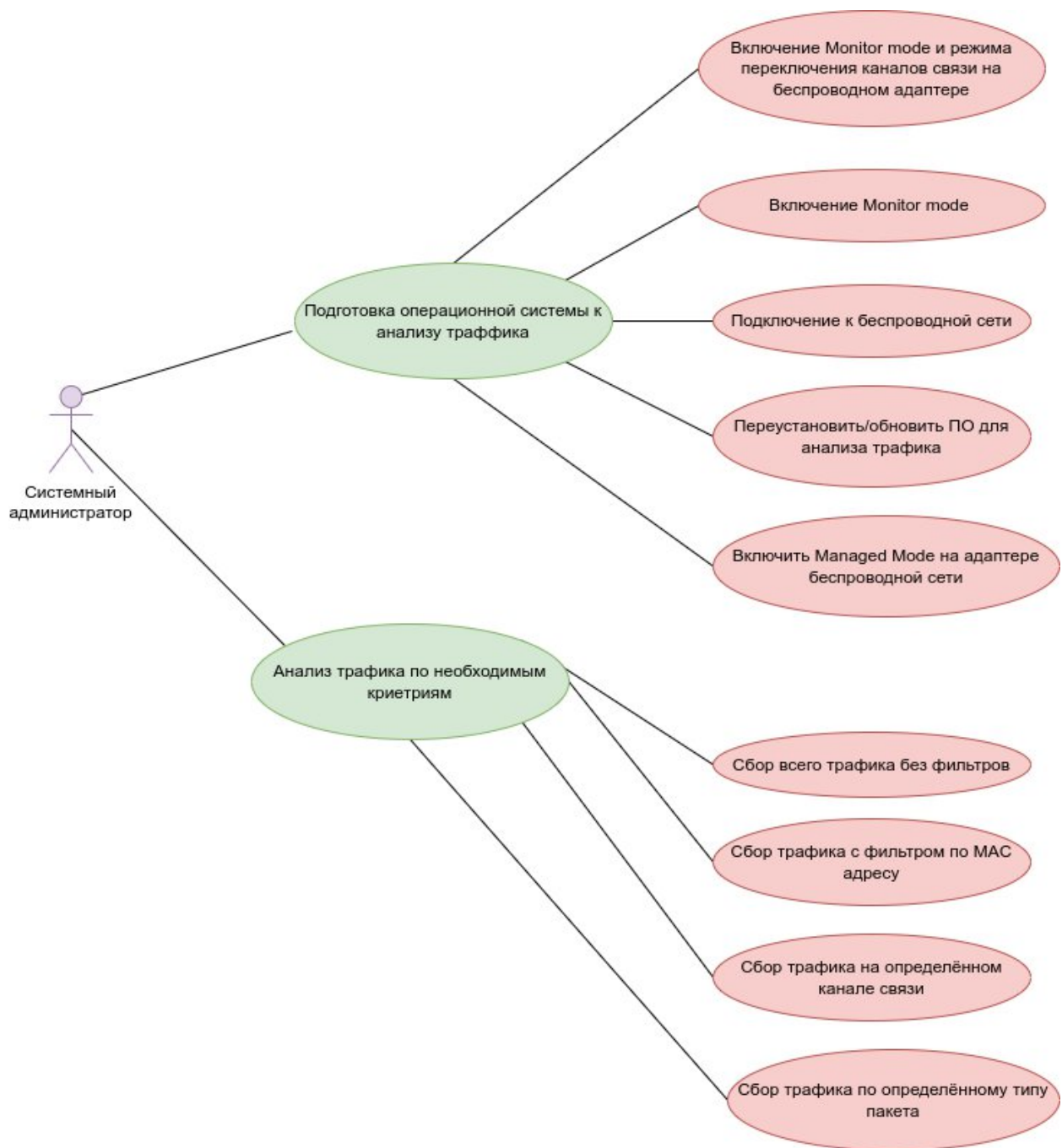


Рисунок 11 — Диаграмма прецедентов

Таблица 2 — Описание действующих лиц

Название	Профиль, подготовка и навыки
Системный администратор	Пользователь, имеет профильное образование, знаком как с текстовым, так и графическим интерфейсом

Таблица 3 — Описание вариантов использования

Действующее лицо	Цель	Краткое описание
------------------	------	------------------

Системный администратор	Подготовка операционной системы к анализу трафика	Установка программного обеспечения для анализа трафика, подготовка сетевого оборудования к анализу трафика в беспроводной сети с учётом особенности сбора трафика в беспроводной сети. Требуются права суперпользователя.
Системный администратор	Анализ трафика по необходимым критериям	Анализ беспроводного трафика без использования фильтров, с использованием фильтров по MAC адресу, номеру канала и типу беспроводного сигнала.

При выполнении действий по сбору трафика в беспроводной сети системный администратор работает в двух основных направлениях:

- подготовка операционной системы к анализу трафика;
- анализ трафика по необходимым критериям

Подготовка операционной системы к анализу трафика заключается в том, чтобы системный администратор подготовил целевое устройство на базе операционной системы семейства GNU/Linux к процедуре анализа трафика в беспроводной сети и настроил режим работы беспроводного адаптера.

Анализ трафика по необходимым критериям позволяет производить действия, направленные на сбор трафика в беспроводной сети, его фильтрацию и запись в конечный файл, который принимает для обработки Filebeat.

### **2.6.3 Формирование функциональных требований к сценариям на командном процессоре Bash**

Предполагается создать два сценария на командном процессоре Bash с названиями «start.sh» и «begin-analis (no root)».

Сценарий «start.sh» должен выполнять следующие функции:

1. Определять, запущен ли скрипт от имени суперпользователя;
2. Определять версию GNU/Linux на которой он запущен;
3. Менять режим работы адаптера беспроводной сети на Monitor и Managed mode;
4. Подключать беспроводной адаптер к нужной Wi-Fi сети;
5. Устанавливать недостающие пакеты для анализа трафика в автоматическом режиме;
6. Циклично, с заданным интервалом переключать каналы беспроводного адаптера;
7. Иметь интерактивное меню.

Сценарий «begin-analis (no root)» должен выполнять следующие функции:

1. Собирать весь трафик в беспроводной сети;
2. Иметь интерактивное меню;
3. Собирать весь трафик, фильтруя его по MAC адресу
4. Проверять на какой версии GNU/Linux запущен сценарий;
5. Проверять наличие необходимых для анализа сетевых пакетов в ОС, и в случае ошибки выводить запрос с просьбой запустить сценарий «start.sh» для исправления этих ошибок.
6. Собирать весь трафик беспроводной сети, с фильтруя его по отдельному каналу беспроводной сети;
7. Собирать весь трафик беспроводной сети, фильтруя его по следующим типам пакетов беспроводной сети:
  - 7.1. Фрейм управления;
  - 7.2. Фрейм контроля;
  - 7.3. Фрейм данных;
  - 7.4. Запрос на установку связи;
  - 7.5. Ответ на установку связи;
  - 7.6. Запрос на повторную установку связи;
  - 7.7. Ответ на повторную установку связи;

- 7.8. Запрос на зондирование;
- 7.9. Ответ на зондирование;
- 7.10. Сигнальный пакет;
- 7.11. Разрыв связи;
- 7.12. Аутентификация;
- 7.13. Отказ в аутентификации;
- 7.14. Фрейм действия;
- 7.15. Запросы на подтверждение блокировки;
- 7.16. Подтверждение блокировки;
- 7.17. Опрос энергосбережения;
- 7.18. Запрос на передачу;
- 7.19. Готовность к приёму;
- 7.20. Подтверждение приёма;
- 7.21. Окончание бесконфликтного периода;
- 7.22. Пустые данные (NULL data);
- 7.23. Данные о качестве обслуживания;
- 7.24. Пустые данные о качестве обслуживания.

#### **2.6.4 Диаграмма последовательности**

Диаграмма последовательности — это диаграмма, на которой показан жизненный цикл какого-либо объекта на единой временной оси, в рамках какого-либо временного прецедента. Для разрабатываемой подсистемы диаграмма последовательности построена для прецедента «Сканирование сетевого трафика». Диаграмма последовательности представлена на рисунке 12.

Объект «Системный администратор» направляет запрос на анализ трафика по какому-либо критерию, заполняя имя сетевого интерфейса, на котором необходимо сканировать данные и путь, по которому данные будут сохранены. После старта анализа трафика Filebeat обнаруживает файлы, содержащие данные о пакетах, передаёт их в свою очередь в Logstash. Logstash

обрабатывает эти данные согласно фильтрам, в заданной конфигурации, Kibana считывает обработанные данные и передаёт их пользователю в графическом представлении.

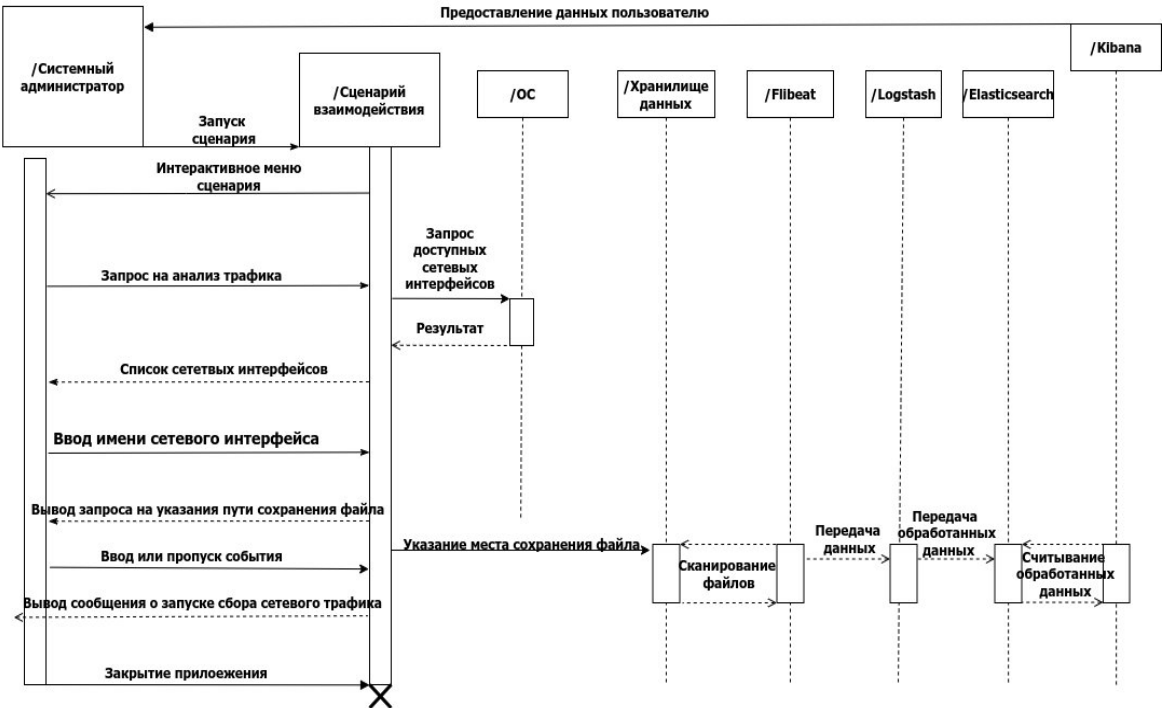
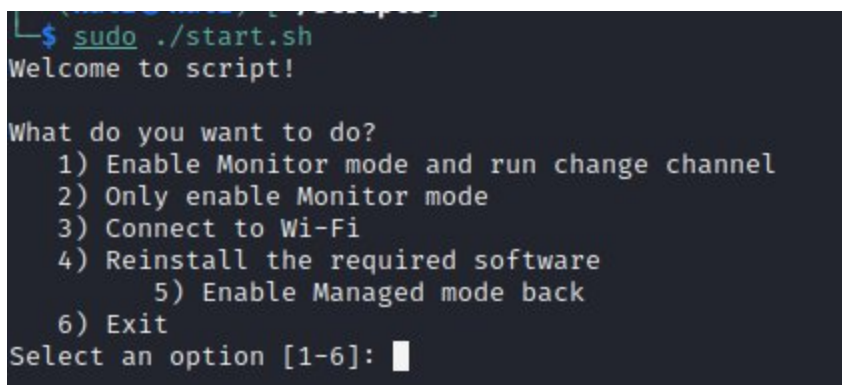


Рисунок 12 — Диаграмма последовательности для прецедента «Сканирование сетевого трафика»

## 2.6.5 Проектирование интерфейса пользователя

Интерфейс пользователя — это совокупность аппаратных и программных средств, которые обеспечивают взаимодействие пользователя с компьютером. Диалоги являются основой такого взаимодействия. Под диалогом понимают регламентированный обмен информацией между пользователем и компьютером, который осуществляется в реальном времени и направленный на решение конкретной задачи. Диалог должен состоять из отдельных процессов ввода-вывода, которые физически обеспечивают связь пользователя и компьютера. Обмен информацией осуществляется передачей сообщения.

Главное меню пользовательского интерфейса сценария «start.sh» представлено на рисунке 13.



```
$ sudo ./start.sh
Welcome to script!

What do you want to do?
 1) Enable Monitor mode and run change channel
 2) Only enable Monitor mode
 3) Connect to Wi-Fi
 4) Reinstall the required software
 5) Enable Managed mode back
 6) Exit
Select an option [1-6]:
```

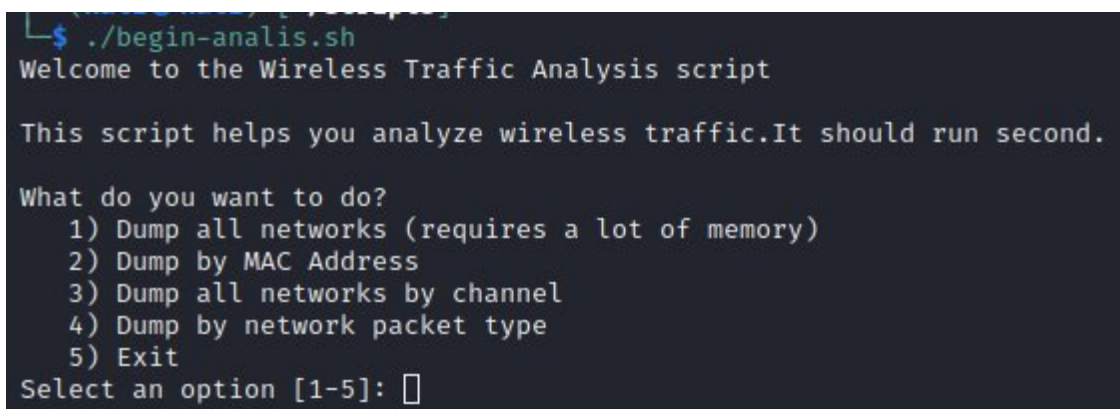
Рисунок 13 — Главное меню пользовательского интерфейса сценария «start.sh»

Главное меню состоит из шести опций, которые вызывают позволяют вызывать необходимые функции в сценарии:

— опция «Enable Monitor mode and run change channel» запускает сценарий включения режима монитора на беспроводном адаптере, помимо этого включает смену каналов у адаптера беспроводной сети (важно понимать, что при запуске данной функции нельзя закрывать окно терминала, в противном случае переключение каналов работать не будет);

- опция «Only enable Monitor mode» запускает сценарий включения режима монитора на беспроводном адаптере без дополнительных опций;
- опция «Connect to Wi-Fi» запускает сценарий подключения к точке доступа беспроводной сети (важно понимать, что данный сценарий не будет работать в режиме монитора из-за особенности этого режима);
- опция «Reinstall the required software» запускает сценарий дополнительной проверки уже существующего программного обеспечения, например, с целью обновления (важно понимать, что сценарий автоматически устанавливает необходимые компоненты, если не может найти их в системе, однако не будет проверять обновления при каждом запуске)
- опция «Enable Managed mode back» позволяет переключить адаптер беспроводной сети в управляемый режим (в котором он работает изначально);
- опция «Exit» закрывает сценарий.

Главное меню пользовательского интерфейса «begin-analis (no root)» представлено на рисунке 14.



```
└─$ ./begin-analis.sh
Welcome to the Wireless Traffic Analysis script

This script helps you analyze wireless traffic.It should run second.

What do you want to do?
  1) Dump all networks (requires a lot of memory)
  2) Dump by MAC Address
  3) Dump all networks by channel
  4) Dump by network packet type
  5) Exit
Select an option [1-5]: █
```

Рисунок 14 — Главное меню пользовательского интерфейса «begin-analis (no root)»

Главное меню состоит из шести опций, которые вызывают позволяют вызывать необходимые функции в сценарии:

- опция «Dump all networks (requires a lot of memory)» запускает сценарий, который позволяет захватить и записать в файл формата JSON весь



трафик в беспроводной сети без каких-либо фильтров (важно понимать, что в режиме монитора захват сетевого трафика может занимать значительное количество памяти);

— опция «Dump by MAC Address» запускает сценарий, который позволяет захватить и записать в файл формата JSON трафик беспроводной сети, фильтруемый на этапе сбора трафика по MAC адресу;

— опция «Dump all networks by channel» запускает сценарий, который позволяет захватить и записать в файл формата JSON трафик беспроводной сети, фильтруемый на этапе сбора трафика по каналу беспроводной связи;

— Опция «Dump by network packet type» открывает меню (рисунок 15) в котором предлагает выбрать опцию по которой будет фильтроваться, захватываемый беспроводной трафик по заданному типу сетевого пакета в беспроводной сети;

— опция «Exit» позволяет выйти из сценария.

```
Select an option [1-5]: 4
Welcome to the Wireless Traffic Analysis script

This script helps you analyze wireless traffic.It should run second.
Select the type (subtype) of the wireless signal:

1) Control frame (0)                14) Action frame (0x0D)
2) Control frame (1)                15) Block confirmation requests (0x18)
3) Data frame (2)                   16) Lock confirmation (0x19)
4) Communication request (0x00)      17) Energy saving poll (0x1A)
5) Connection setup response (0x01)  18) Transfer Request (0x1B)
6) Reconnect Request (0x02)          19) Ready to receive (0x1C)
7) Reconnect response (0x03)         20) Reception confirmation (0x1D)
8) Probing Request (0x04)            21) End of conflict-free period (0x1E)
9) Response to probing (0x05)        22) NULL data (0x24)
10) Signal packet (0x08)              23) Quality of Service Data (0x28)
11) Disconnect (0x0A)                24) Empty quality of service data (0x2C)
12) Authentication (0x0B)             25) Exit
13) Authentication Denied (0x0C)
#? █
```

Рисунок 15 — меню «Dump by network packet type»

Меню «Dump by network packet type» состоит из 25 опций, каждая из которой представляет фильтр типа сетевого пакета в беспроводной сети, по

которой будет фильтроваться захватываемый сетевой трафик. В таблице 4 приведены названия и значения каждого из фильтров.

Таблица 4 — Назначение фильтров по типу сетевого пакета

Название фильтра	Тип сетевого пакета
1) Control frame (0)	Фрейм управления (0)
2) Control frame (1)	Фрейм контроля (1)
3) Data frame (2)	Фрейм данных (2)
4) Communication request (0x00)	Запрос на установку связи (0x00)
5) Connection setup response (0x01)	Ответ на установку связи (0x01)
6) Reconnect Request (0x02)	Запрос на повторную установку связи (0x02)
7) Reconnect response (0x03)	Ответ на повторную установку связи (0x03)
8) Probing Request (0x04)	Запрос на зондирование (0x04)
9) Response to probing (0x05)	Ответ на зондирование (0x05)
10) Signal packet (0x08)	Сигнальный пакет(0x08)
11) Disconnect (0x0A)	Разрыв связи (0x0A)
12) Authentication (0x0B)	Аутентификация (0x0B)
13) Authentication Denied (0x0C)	Отказ в аутентификации (0x0C)
14) Action frame (0x0D)	Фрейм действия (0x0D)
15) Block confirmation requests (0x18)	Запросы на подтверждение блокировки(0x18)
16) Lock confirmation (0x19)	Подтверждение блокировки (0x19)
17) Energy saving poll (0x1A)	Опрос энергосбережения (0x1A)
18) Transfer Request (0x1B)	Запрос на передачу (0x1B)
19) Ready to receive (0x1C)	Готовность к приёму (0x1C)
20) Reception confirmation (0x1D)	Подтверждение приёма (0x1D)
21) End of conflict-free period (0x1E)	Окончание бесконфликтного периода(0x1E)
22) NULL data (0x24)	Пустые данные (NULL data) (0x24)
23) Quality of Service Data (0x28)	Данные о качестве обслуживания (0x28)
24) Empty quality of service data (0x2C)	Пустые данные о качестве обслуживания (0x2C)
25) Exit	Выход из сценария

## **2.7 Определение необходимого уровня прав доступа, управление правами доступа к репозиторию данных о выполнении работ по созданию (модификации) и сопровождению ИС**

В реальных информационных системах разные пользователи должны иметь разные возможности. Как минимум существует разделение на обычных пользователей, которые занимаются решением своих личных или профессиональных задач, используя информационную систему в качестве инструмента и на пользователей технических специалистов, которые занимаются настройкой, модернизацией, восстановлением и обслуживанием той же информационной системы [14].

Из идеи неравноправия пользователей вытекает задача разграничения прав доступа (привилегий): одним разрешены те или иные действия в информационной системе, другим — нет. Но разрешения и запреты должны действовать в отношении конкретных людей или их групп. Решение задач идентификации и аутентификации необходимо именно для реализации механизмов разграничения прав доступа [14].

Чтобы обеспечить защиту объектов, нужно определить, какому субъекту какие из возможных действий разрешение, а какие — нет. При этом стоит отметить, что общие действия, как правило, встречаются во всех системах единообразно. А специфические сильно зависят от конкретной системы. [14].

Разграничение прав доступа, то есть фиксации разрешений на выполнение действий с объектами, может быть основано на идее домена защиты. Каждая пара в домене определяет объект и список операций, которые могут быть применены к этому объекту. Права обычно обозначаются латинскими буквами: R—чтение, W—запись, X—выполнение [14].

Достоинствами подхода доменной защиты являются:

- 1) простота понимания, реализации и использования;
- 2) гибкость — возможность изменения прав для каждого конкретного пользователя на каждый конкретный объект. Вторым подходом является

формирование перечней возможностей. Данный подход отличается от списков управления доступом способом сопоставления субъектов, объектов и прав. Если список управления состоит из записей вида «объект — (субъект 1 — права), (субъект 2 — права)», то перечень возможностей состоит из записей вида «субъект — (объект 1 — права), (объект 2 — права)» [14].

После анализа бизнес процессов организации, анализа существующих решений, инсталляции оборудования для реализации хранения и обработки, собираемых сетевых пакетов, анализа прецедентов, построения диаграммы последовательности и описания процесса определения необходимого уровня прав доступа необходимо приступать к процессу разработки.

## **3 РАЗРАБОТКА ПОДСИСТЕМЫ**

### **3.1 Описание процесса разработки информационной подсистемы**

#### **3.1.1 Настройка параметров безопасности Elastic Stack**

После ознакомления с существующими на данный момент системами управления базами данных и спецификой структуры сетевых пакетов было принято решение для обработки, хранения и дальнейшего удобного графического представления использовать NoSQL решение Elastic Stack. У данного программного продукта есть коммерческая версия, но для целей индексирования информации сетевых пакетов будет достаточно бесплатной версии программного решения.

Основные преимущества Elastic Stack:

— горизонтально масштабируемый поиск — данная подсистема может работать в любой сфере, где необходим сбор данных среднего объема, например, можно собирать данные от 1000 до 30000 таких устройств как холодильники, мобильные устройства, персональные компьютеры, интерактивных панелей и т.п.;

— многопоточность — возможность одновременно принимать данные с большого количества устройств;

— отказоустойчивость — в случае сбоя кластерных узлов данные не потеряются, а будут перераспределены, и поисковая система сама продолжит работу. Операционная стабильность достигается ведением логов на каждое изменение данных в хранилище сразу на нескольких узлах кластера [16];

— поисковые индексы возможно делить на сегменты, каждый сегмент может иметь несколько реплик;

— хорошо подходит для работы с большим объёмом данных (2–10 терабайт в год, 20–30 миллиардов документов в индексах);

— легковесные агенты (например, Filebeat) позволяют собирать необходимые данные с конкретного устройства (рабочей станции или сервера).

К недостаткам Elastic Stack следует относить следующие пункты:

— относительно высокое потребление ресурсов из-за использования JVM. На практике, при высоких нагрузках может не хватить производительности, однако показатель производительности в данном случае является субъективным фактором;

— сложность внутреннего языка запросов к базам данных QueryDSL;

— информационная безопасность — по умолчанию Elasticsearch не имеет встроенной системы ограничения прав доступа и системы авторизации, после установки движок открывает порт 9200 для всех доступных интерфейсов, что открывает доступ к базе данных — поэтому необходимо принимать комплекс мер связанный со сторонними средствами, такими как Firewall, помимо этого, для разграничения доступа к веб-интерфейсу Kibana рекомендуется настроить идентификацию по логину и паролю средствами nginx.

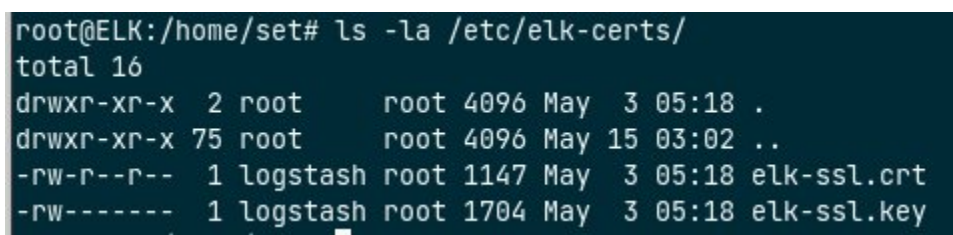
Оказывая особенное внимание на недостаток связанный с информационной безопасностью Elastic Stack, необходимо предложить следующий комплекс мер по обеспечению информационной безопасности данных:

1. Сгенерировать SSL ключ и сертификат, который будет являться идентификатором для передачи данных в Logstash;
2. Открыть порт 9200 только для localhost, установить Logstash на тот же сервер, что и Elasticsearch, чтобы данные передавались только по localhost;
3. Настроить идентификацию в Kibana по логину и паролю средствами nginx.

Для генерации SSL ключа и сертификата необходимо подключиться к серверу, на котором располагается Elastic Stack и выполнить следующие действия:

1. Создать папку от имени суперпользователя, в которой будут храниться сертификаты, для удобства будет создана папка в директории настроек по пути «/etc/elk-certs». Команда для создания папки: «mkdir /etc/elk-certs»;
2. Перейти в созданную папку с помощью команды: «cd /etc/elk-certs»;
3. Создать SSL сертификаты с помощью команды: «openssl req -subj '/CN=logstash-server.local/' -x509 -days 365 -batch -nodes -newkey rsa:2048 -keyout elk-ssl.key -out elk-ssl.crt». Важно понимать, что сертификаты создаются всего на год и для доменного имени в локальной сети. По истечении срока действия сертификатов их необходимо создавать заново;
4. Дать права для созданных сертификатов пользователю logstash с помощью команд: «chown logstash elk-ssl.crt» и «chown logstash elk-ssl.key».

При просмотре получившихся файлов с помощью команды «ls -la», результат должен соответствовать рисунку 16.



```
root@ELK:/home/set# ls -la /etc/elk-certs/
total 16
drwxr-xr-x  2 root    root  4096 May  3 05:18 .
drwxr-xr-x 75 root    root  4096 May 15 03:02 ..
-rw-r--r--  1 logstash root  1147 May  3 05:18 elk-ssl.crt
-rw-----  1 logstash root  1704 May  3 05:18 elk-ssl.key
```

Рисунок 16 — Корректно созданные ключи, с корректно выданными правами

Доступ к порту 9200 только для localhost настраивается в конфигурационном файле Elasticsearch, который располагается в «/etc/elasticsearch/elasticsearch.yml», в секции «Network». Результат такой настройки отображён на рисунке 17.

```
# ----- Network -----
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
#network.host: 127.0.0.1
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
#http.port: 9200
#
# For more information, consult the network module documentation.
#
```

Рисунок 17 — Конфигурация доступа сети для Elasticsearch

Принимая тот факт, что Kibana по умолчанию примет запросы на открытом порту 5601, сетевой адрес веб интерфейса можно узнать с помощью команды: «netstat -tulpn | grep 5601». Результаты команды представлены на рисунке 18

```
root@ELK:/home/set# netstat -tulpn | grep 5601
tcp        0      0 192.168.88.217:5601  0.0.0.0:*        LISTEN      425/node
root@ELK:/home/set#
```

Рисунок 18 — Адрес веб-интерфейса Kibana

Для удобства будет добавлена А запись для DNS сервера которая будет сопоставлять IP адрес 192.168.88.217 с доменным именем «elk.lan».

При вводе в адресную строку браузера адреса: «<http://elk.lan/>:5601» откроется интерфейс Kibana, как это показано на рисунке 19. Проблема безопасности заключается в том, что в этот интерфейс может попасть любой пользователь, что делает данную систему не безопасной. Средства идентификации Elastic Stack поставляются только с платной лицензией, именно поэтому необходимо воспользоваться сторонними бесплатными решениями. Одним из таких сторонних бесплатных решений является nginx.



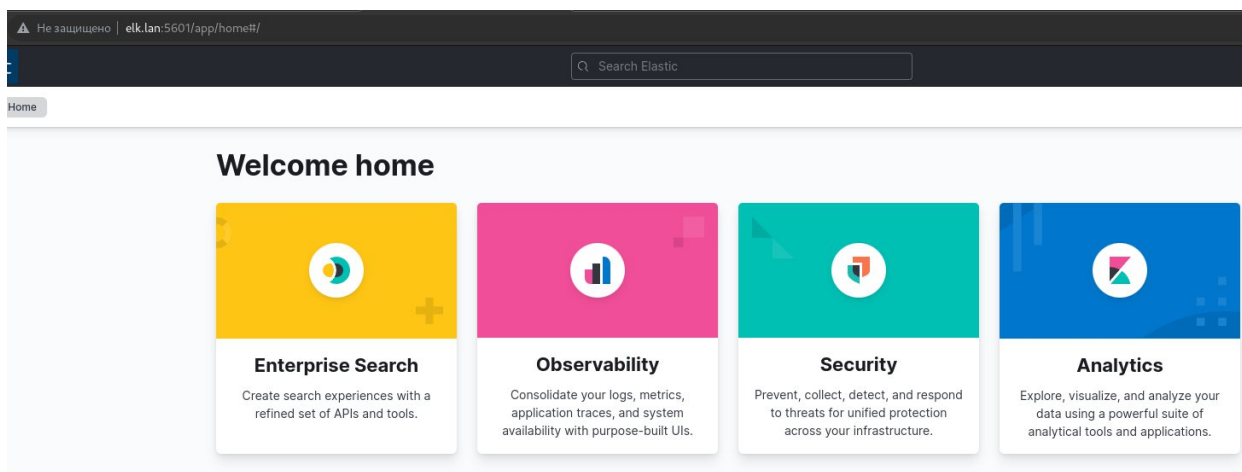


Рисунок 19 — Стартовая страница Kibana

Для установки `nginx` необходимо воспользоваться командой: «`sudo apt install nginx`». Данный пакет есть в репозиториях многих GNU/Linux систем. После установки необходимо добавить демон `nginx` в автозагрузку, после чего запустить этот демон с помощью команды: «`systemctl enable nginx && systemctl start nginx`».

Для удобства помимо авторизации по логину и паролю, в качестве дополнительной меры безопасности нужно настроить обратный прокси-сервер, который будет перенаправлять запросы на «`localhost:5601`» и дать доменному имени самоподписанный SSL сертификат.

Самоподписанный SSL сертификат генерируется командой: «`sudo openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout kibana.key -out kibana.crt`». После чего пусть в данном сертификате необходимо указать в файле конфигурации `nginx`.

Для настройки обратного прокси сервера необходимо в папке: «`/etc/nginx/sites-enabled`» создать файл конфигурации с названием «`elk`». После чего этот файл наполнить содержимым согласно рисунку 20.

```
server {
    #listen 80 default_server;
    listen 443 ssl;

    server_name elk.lan;
    ssl_certificate /etc/nginx/kibana.crt;
    ssl_certificate_key /etc/nginx/kibana.key;

    location / {
        auth_basic "Restricted Access";
        auth_basic_user_file /etc/nginx/.htpasswd;
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;

    }
}
```

Рисунок 20 — Содержание файла настроек обратного прокси сервера и авторизации nginx для сайта elk.lan

В секции «auth\_basic\_user\_file /etc/nginx/.htpasswd» файла настроек (рисунок 20) присутствует информация о том, где необходимо создать файл с логинами и паролями для доступа к веб-интерфейсу Kibana. Для создания такого файла и создания пользователя необходимо воспользоваться командой: «sudo sh -c "echo -n 'kibanauser:'>>/etc/nginx/.htpasswd"», а для задания пароля пользователю: «sudo sh -c "openssl passwd -apr1 >> /etc/nginx/.htpasswd"». Помимо этого, в конфигурационном файле Kibana необходимо изменить сетевой адрес принятия запросов на localhost (рисунок 21).

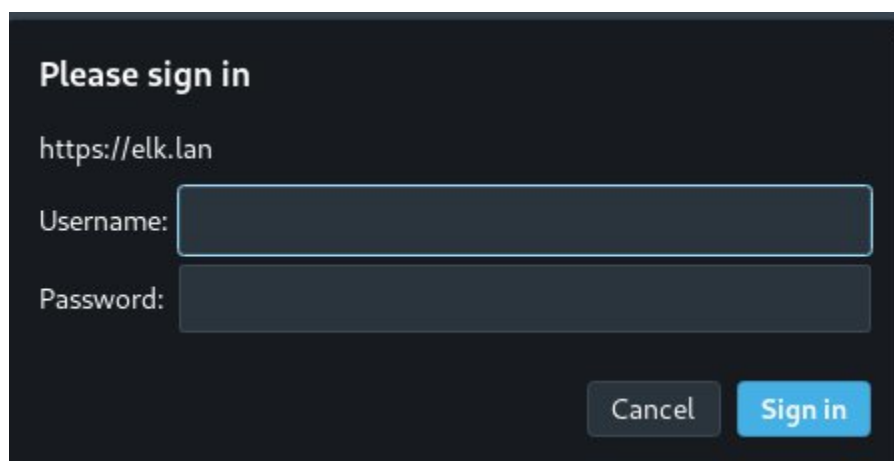
После создания пользователя и присвоения ему пароля, необходимо перейти в веб-интерфейс Kibana для проверки. В результате перехода по адресу «elk.lan» появляется форма авторизации (рисунок 22), введя в которой учётные данные пользователь попадёт в веб-интерфейс Kibana.

Следует считать, что после выполнения действий, направленных на повышения безопасности при взаимодействии и передачи данных в Elastic Stack можно приступить к разработке сценариев, выполняющих вспомогательные функции для захвата сетевого трафика.

```
# Kibana is served by a back end server. This s
#server.port: 5601

# Specifies the address to which the Kibana ser
# The default is 'localhost', which usually mea
# To allow connections from remote users, set t
#server.host: "192.168.88.217"
server.host: "127.0.0.1"
```

Рисунок 21 — Обновлённая конфигурация Kibana



Please sign in

https://elk.lan

Username:

Password:

Cancel Sign in

Рисунок 22 — окно авторизации локального сайта «elk.lan»

### **3.1.2 Разработка сценариев взаимодействия с подсистемой хранения трафика в беспроводной сети**

Исходя из функциональных требований к проектируемой программной системе, с целью обеспечения возможности масштабирования, сценарии командной оболочки необходимо разделить на функции, которые в свою очередь вызывать по мере необходимости с помощью интерактивного меню.

#### **3.1.2.1 Реализация функции проверки прав суперпользователя**

Реализация функции проверки прав суперпользователя представлена на рисунке 23.

```
#Check ROOT
if [ "$(id -u)" ≠ 0 ]
then
    echo root permission required >&2
    exit 1
fi
```

Рисунок 23 — Реализация функции проверки прав суперпользователя для сценарий командной оболочки Shell

Данная функция запрашивает идентификатор пользователя системы, и если его идентификатор не равен 0, то сценарий прекращает свою работу и выдаёт ошибку.

### 3.1.2.2 Реализация функции определения операционной системы на которой запущен сценарий

Реализация функции для определения операционной системы на которой запущен сценарий частично продемонстрирована на рисунке 24.

```
#CheckOS
function checkOS() {
    if [[ -e /etc/debian_version ]]; then
        OS="debian"
        source /etc/os-release

        if [[ $ID = "debian" || $ID = "raspbian" ]]; then
            if [[ $VERSION_ID -lt 9 ]]; then
                echo "Your version of Debian is not supported."
                echo ""
                echo "However, if you're using Debian ≥ 9 or unstable/testing then you can continue, at your own risk."
                echo ""
                until [[ $CONTINUE =~ (y|n) ]]; do
                    read -rp "Continue? [y/n]: " -e CONTINUE
                done
                if [[ $CONTINUE = "n" ]]; then
                    exit 1
                fi
            fi
        fi
    fi
}
```

Рисунок 24 — Реализация функция определения версии ОС GNU/Linux

В данном случае функция сверяет версию ОС Debian, если версия меньше чем 9, то выводится предупреждение о том, что сценарий может отработать некорректно. С полным текстом функции можно ознакомиться в приложении А.

### 3.1.2.3 Реализация функции изменения режима работы адаптера беспроводной сети

Реализация функции изменения режима работы адаптера беспроводной сети представлена на рисунке 25.

<pre>function onlyMonitor() {     echo "Available network interfaces:"     iwconfig     read -p "Enter name WLAN interface: " wlan     iwconfig \$wlan mode monitor     echo "Monitor mode is enabled"     echo ""     echo "Check:"     echo ""     iwconfig                                #Enable monitor     exit 0 }</pre>	<pre>function onlyManaged() {     echo "Available network interfaces:"     iwconfig     read -p "Enter name WLAN interface: " wlan     iwconfig \$wlan mode managed     echo "Managed mode is enabled"     echo ""     echo "Check:"     echo ""     iwconfig                                #Enable managed     exit 0 }</pre>
---	---

Рисунок 25 — Реализация функции изменения режима работы адаптера беспроводной сети

Данная функция вызывает сценарий, который при взаимодействии с ним позволяет изменить режим работы беспроводного адаптера.

### 3.1.2.4 Реализация функции вызова сценария для подключения к беспроводной сети

Реализация функции вызова сценария для подключения к беспроводной сети представлена на рисунке 26.

```
function connectWifi() {
    echo "Plese, connect to Wi-fi"
    nmcli radio wifi on
    nmcli dev wifi list
    read -p "Enter SSID: " SSID
    nmcli --ask dev wifi connect $SSID
}
```

Рисунок 26 — Реализация функции вызова сценария для подключения к беспроводной сети

Данная функция выводит список доступных беспроводных сетей, запрашивает ввести название необходимой сети, после чего подключается к этой сети.

### 3.1.2.5 Реализация функции сканирования установленных пакетов в ОС и автоматической установки их в ОС

Реализация функции сканирования установленных пакетов в ОС и автоматической установки их в ОС представлена на рисунке 27

```
function configOS() {
    ##Setting OS
    ##Install packages and other
    checkOS
    if [ $OS = "ubuntu" ] || [ $OS = "debian" ]; then
        add-apt-repository ppa:wireshark-dev/stable -y
        apt update
        apt install wireshark tshark network-manager -y
        usermod -aG wireshark $(whoami)
    fi

    if [ $OS = "fedora" ] || [ $OS = "centos" ]; then
        dnf install wireshark-qt tshark network-manager -y
        usermod -aG wireshark $(whoami)
    fi
}
```

Рисунок 27 — Реализация функции сканирования установленных пакетов в ОС и автоматической установки их в ОС

Данная функция устанавливает необходимые пакеты, если они отсутствуют, в зависимости от обнаруженной ОС.

### 3.1.2.6 Реализация функции цикличной смены каналов на адаптере беспроводной сети

Реализация функции цикличной смены каналов на адаптере беспроводной сети представлена на рисунке 28.

```
function loopChannels() {  
    ##function switches channels of the wireless network  
    echo "Available network interfaces:"  
    iwconfig  
    read -p "Enter name WLAN interface: " wlan  
    iwconfig $wlan mode monitor  
    echo "Monitor mode is enabled"  
    echo "Channel change started"  
    while [[ true ]]; do  
        #statements  
        for channels in {1..13}  
        do  
            sleep 2  
            iwconfig $wlan channel $channels  
        done  
    done  
}
```

Рисунок 28 — Реализация функции цикличной смены каналом на адаптере беспроводной сети представлена

Данная функция запускает диалог, при котором необходимо включить режим монитора, запускает бесконечный цикл, который переключает каналы на беспроводном адаптере.

### 3.1.2.7 Реализация функции запуска сценария для сбора сетевого трафика в беспроводной сети без использования фильтров.

Реализация функции запуска сценария для сбора сетевого трафика в беспроводной сети без использования фильтров представлена на рисунке 29.

Данная функция запускает сценарий для сбора трафика, в реальном времени записывает захваченные пакеты в файлы формата JSON, при этом даёт название этим файлам по времени запуска сбора сетевых пакетов.



```

function tshark_all() {
    #This function is useful when working in monitore mode
    echo "List interfaces: "
    echo "Please wait..."
    tshark -D
    read -p "Enter interface name: " IFname
    read -p "Enter the path to save the dump (default is '/home/$(whoami)):"
    " path_d
    if [ -n $path_d ]
    then
        message_Stop
        tshark -i $IFname -T ek > $HOME/$(date +%d.%m.%Y_%H.%M').json
    else
        message_Stop
        tshark -i $IFname -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
    fi
}

```

Рисунок 29 — Реализация функции запуска сценария для сбора сетевого трафика в беспроводной сети без использования фильтров

### 3.1.2.8 Реализация функции интерактивного меню

Реализация функции интерактивного меню представлена на рисунке 30.

В данной функции описан текстовый интерфейс, с которым будет взаимодействовать пользователь. Меню взаимодействия состоит из пяти пунктов за каждым пунктом закреплена вызываемая функция. При вызове любой из функций, запускается сценарий этой функции.

Все остальные функции представлены в листинге сценариев командной оболочки приложения А и приложения Б.

После представления реализаций основных функций, необходимо приступить к описанию работы информационной подсистемы анализа трафика в беспроводных сетях.



```

function startMenu() {
    echo "Welcome to the Wireless Traffic Analysis script"
    echo ""
    echo "This script helps you analyze wireless traffic.It should run second"
    echo ""
    echo "What do you want to do?"
    echo "  1) Dump all networks (requires a lot of memory)"
    echo "  2) Dump by MAC Address"
    echo "  3) Dump all networks by channel"
    echo "  4) Dump by network packet type"
    echo "  5) Exit"
    until [[ $MENU_OPTION =~ ^[1-5]$ ]]; do
        read -rp "Select an option [1-5]: " MENU_OPTION
    done

    case $MENU_OPTION in
        1)
            tshark_all
            ;;
        2)
            tshark_filter_MAC
            ;;
        3)
            filter_Channel
            ;;
        4)
            filter_TypePacketsMenu
            ;;
        5)
            exit 0
            ;;
    esac
}

```

---

Рисунок 30 — Реализация функции интерактивного меню

### 3.2 Описание работы информационной подсистемы

Для наглядного описания работы информационной подсистемы анализа трафика в беспроводных сетях была разработана структурная схема данного решения. Структурная схема представлена на рисунке 31.

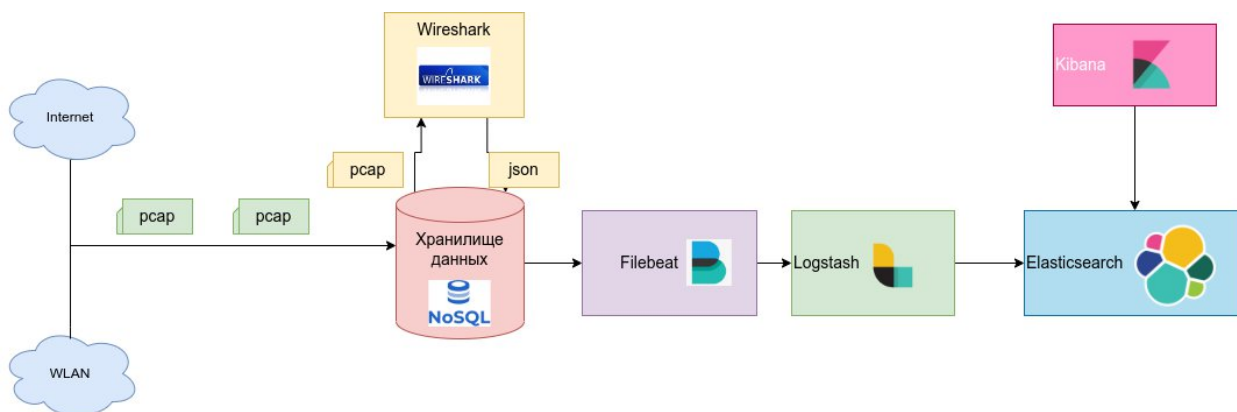


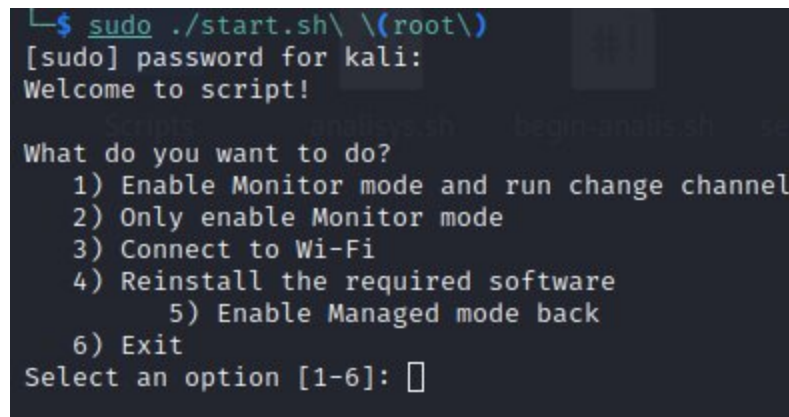
Рисунок 31 — Структурная схема взаимодействия элементов в подсистеме исследования трафика

Исходя из структурной схемы, которая представлена на рисунке допустимо сказать, что принцип работы данной структурной схемы будет следующим:

1. Программное обеспечение для захвата сетевого трафика Wireshark будет перехватывать сетевой трафик в беспроводной сети;
2. Перехваченный сетевой трафик будет сохраняться в каталоге, о котором будет знать Filebeat, в формате, который понимает Filebeat;
3. Filebeat будет индексировать файлы в каталоге, в который планируется сохранять перехваченные сетевые пакеты после чего передавать эти данные в Logstash;
4. Logstash в свою очередь будет принимать данные, отсеивать данные по заданным в конфигурации Logstash фильтрам и передавать эти данные в Elasticsearch;
5. ElasticSearch будет хранить эти данные;
6. В свою очередь Kibana будет обращаться к этим данным с целью дальнейшего представления этих данных.

Для корректного запуска процедуры анализа трафика в беспроводных сетях присутствуют два сценария с интерактивными меню. Перед тем как запускать сценарии рекомендуется установить и настроить Filebeat. Первым

необходимо запускать сценарий «start.sh» от имени суперпользователя. Запущенный сценарий представлен на рисунке 32.



```
$ sudo ./start.sh\ \(\root\  
[sudo] password for kali:  
Welcome to script!  
What do you want to do?  
1) Enable Monitor mode and run change channel  
2) Only enable Monitor mode  
3) Connect to Wi-Fi  
4) Reinstall the required software  
5) Enable Managed mode back  
6) Exit  
Select an option [1-6]:
```

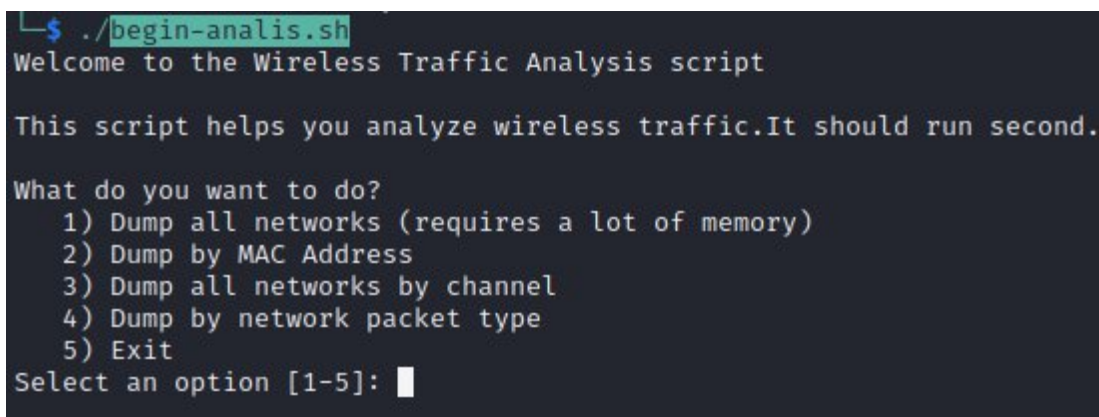
Рисунок 32 — Запущенный сценарий «start.sh»

После запуска сценарий, перед тем как выдать интерактивное меню, в фоне проверил наличие пакетов в системе, которые необходимы для захвата беспроводного трафика.

Интерактивное меню предлагает пользователю следующие сценарии взаимодействия:

1. Включить режим монитора и запустить бесконечный цикл смены каналов беспроводной связи для того, чтобы было удобно собирать пакеты на всех каналах;
2. Использовать только режим монитора (каналы связи переключаться не будут и по умолчанию трафик будет собираться с первого канала беспроводной связи);
3. Подключиться к конкретной беспроводной сети;
4. Переустановить/обновить необходимое программное обеспечение;
5. Установить адаптер беспроводной связи в управляемый режим;
6. Выйти из сценария.

После взаимодействия со сценарием `start.sh` необходимо запустить сценарий «`begin-analis.sh`» для непосредственного перехвата сетевого трафика и его сохранения в файл. Запущенный сценарий представлен на рисунке 33.



```
└─$ ./begin-analis.sh
Welcome to the Wireless Traffic Analysis script

This script helps you analyze wireless traffic.It should run second.

What do you want to do?
  1) Dump all networks (requires a lot of memory)
  2) Dump by MAC Address
  3) Dump all networks by channel
  4) Dump by network packet type
  5) Exit
Select an option [1-5]: █
```

Рисунок 33 — Запущенный сценарий «`begin-analis.sh`»

Данный сценарий в фоновом режиме сканирует необходимые для его работы пакеты в системе и если их нет, то будет выдавать сообщение о необходимости запустить в первую очередь сценарий «`start.sh`». На выбор сценарий предоставляет интерактивное меню, в котором можно выбрать из следующих типов захвата:

1. Захватывать весь сетевой трафик (может потребоваться большой объём памяти);
2. Захват сетевых пакетов, которые имеют в заголовках определённый MAC адрес;
3. Захват сетевые пакетов, которые имеют в заголовках определённый канал связи;
4. Захват определённого типа пакетов.

При выборе захвата по MAC адресу, откроется меню, где будет предложено ввести название сетевого интерфейса из доступных для сканирования, предложенных сценарием, затем необходимый MAC адрес и затем указать место, где будет сохранён файл с сетевыми пакетами (рисунок 34).

```

What do you want to do? (1) begin analysis (2) setting up sh (3) start sh (root)
  1) Dump all networks (requires a lot of memory)
  2) Dump by MAC Address
  3) Dump all networks by channel
  4) Dump by network packet type
  5) Exit
Select an option [1-5]: 2

Process started
Press Ctrl + C to STOP

1. eth0
2. wlan0
3. any
4. lo (Loopback)
5. bluetooth-monitor
6. nflog
7. nfqueue
8. dbus-system
9. dbus-session
10. ciscodump (Cisco remote capture)
11. dpauxmon (DisplayPort AUX channel monitor capture)
12. randpkt (Random packet generator)
13. sdjournal (systemd Journal Export)
14. sshdump (SSH remote capture)
15. udpdump (UDP Listener remote capture)
Enter interface name: wlan0
Enter MAC address: 6C:3B:6B:F3:AA:BD
Enter the path to save the dump (default is '/home/kali'): /var/files/packets

Process started
Press Ctrl + C to STOP

Capturing on 'wlan0'
** (tshark:9281) 09:43:42.444646 [Main MESSAGE] -- Capture started.
** (tshark:9281) 09:43:42.444769 [Main MESSAGE] -- File: "/tmp/wireshark_wlan0LSRCM1.pcapng"

```

Рисунок 34 — Взаимодействие со сценарием захвата сетевых пакетов, которые имеют в заголовках определённый MAC адрес

При взаимодействии со сценарием захвата сетевые пакеты, которые имеют в заголовках определённый канал связи откроется меню, где будет необходимо ввести название сетевого интерфейса из доступных для сканирования, предложенных сценарием, затем номер канала беспроводной связи и указать путь, куда будет сохранён собранный дамп (рисунок 35).

```

Select an option [1-5]: 3

Welcome!
To analyze packets on a specific channel, enter a number between 1 and 13, where the number will be the channel number.
Please stop the 'start.sh' script if it is currently looping through the wireless channels, put the adapter into Monitor mode

Available network interfaces:
1. eth0
2. any
3. lo (Loopback)
4. wlan0
5. bluetooth-monitor
6. nflog
7. nfqueue
8. dbus-system
9. dbus-session
10. ciscodump (Cisco remote capture)
11. dpauemon (DisplayPort AUX channel monitor capture)
12. randpkt (Random packet generator)
13. sdjournal (systemd Journal Export)
14. sshdump (SSH remote capture)
15. udpdump (UDP Listener remote capture)

Enter Name Interface: wlan0
Enter number channel : 1

Enter the path to save the dump (default is '/home/kali'):

Process started
Press Ctrl + C to STOP

Capturing on 'wlan0'
** (tshark:11298) 09:51:43.460605 [Main MESSAGE] -- Capture started.
** (tshark:11298) 09:51:43.460689 [Main MESSAGE] -- File: "/tmp/wireshark_wlan0M7QDM1.pcapng"
```

Рисунок 35 — Взаимодействие со сценарием захвата сетевых пакетов, которые имеют в заголовках определённый канал связи

При взаимодействии со сценарием захвата сетевых пакетов, которые имеют в заголовках определённый канал связи открывается меню, где необходимо выбрать тип захватываемых пакетов, после выбора необходимого пункта необходимо ввести название сетевого интерфейса из предложенных сценарием и доступных в системе, после чего указать место для сохранения сетевых пакетов.

После сбора сетевых пакетов и сохранения их в директорию, с которой их считывает Filebeat, необходимо перейти в веб-интерфейс Kibana, где перейти в раздел Discover, после чего увидеть полученные данные (рисунок 36).



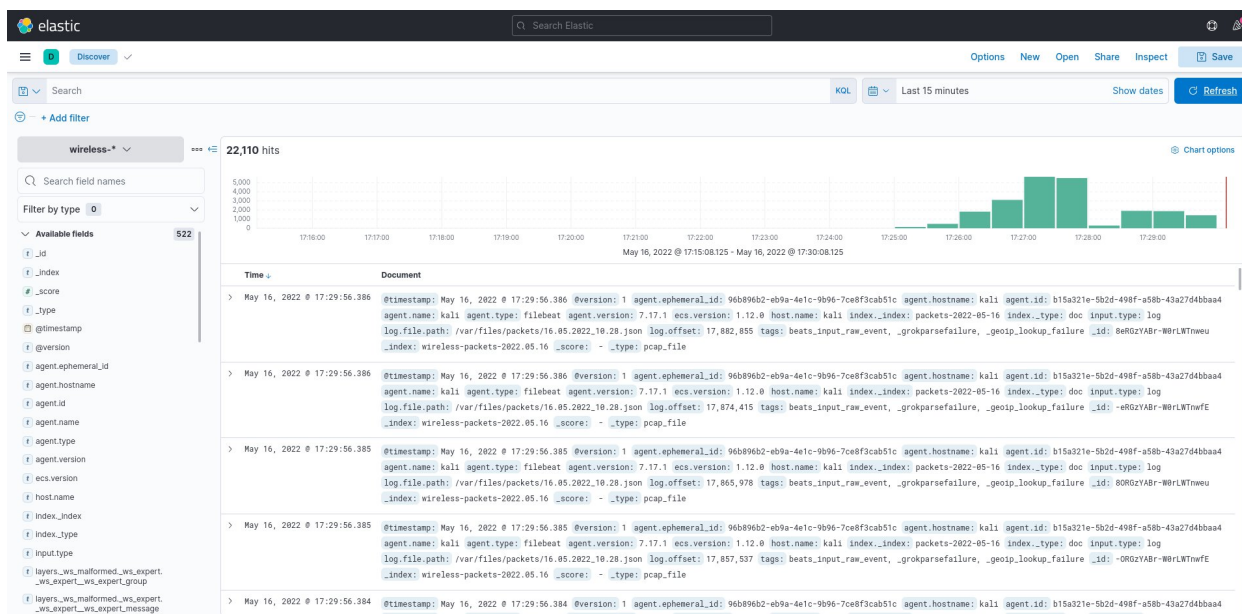


Рисунок 36 — окно Discover с полученными данными

Примером манипуляции с полученными данными может быть построение гистограммы на дашборде на которой отображена статистика занимаемых каналов связи точками доступа за текущий день (рисунок 37).

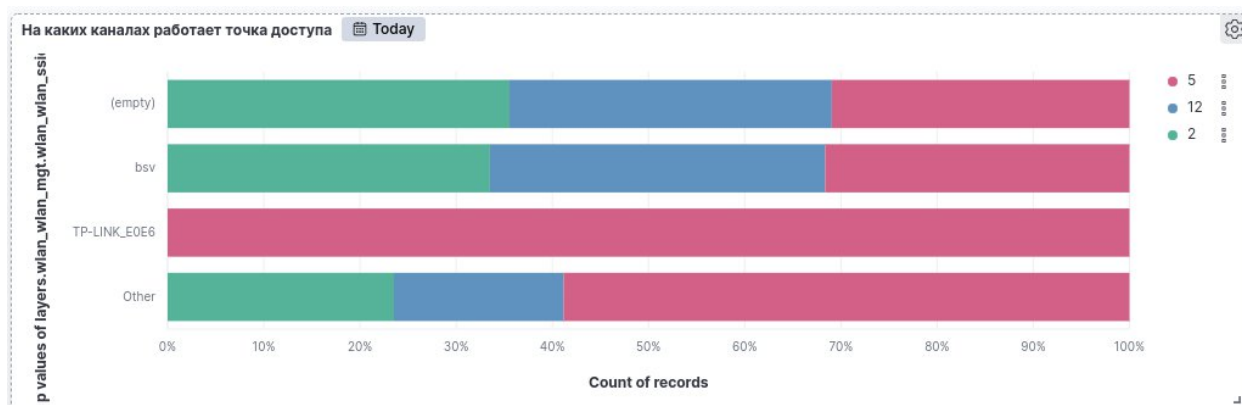


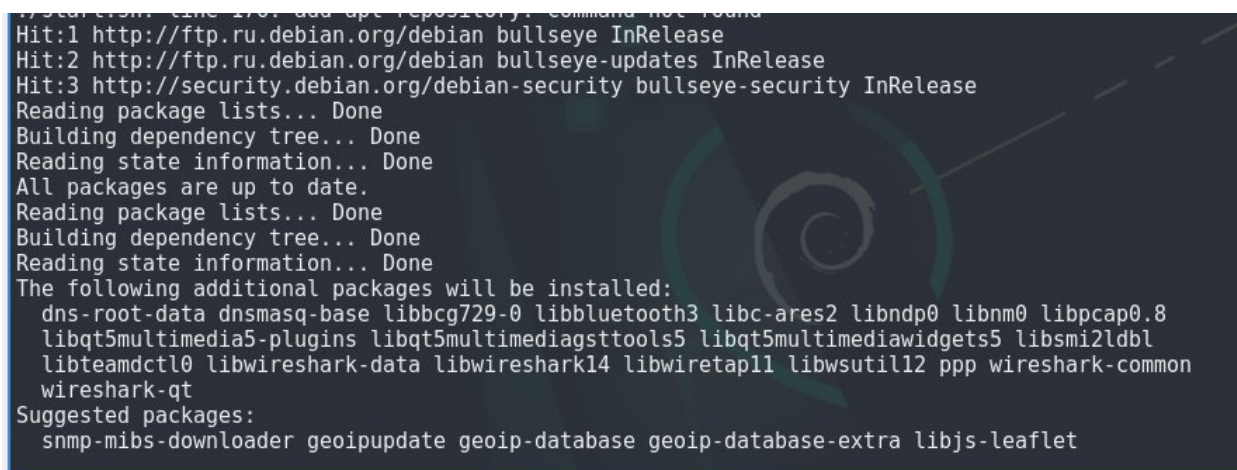
Рисунок 37 — Статистика занимаемых каналов связи точками доступа за текущий день

### 3.3 Тестирование прототипа ИС на проверку корректности архитектурных решений

Для проверки сценариев командной оборочки будет правильными применить функциональный подход. Средой для тестирования сценариев

будет являться установленная на виртуальную машину без каких-либо настроек ОС Debian 11.


После запуска сценария «start.sh», сценарий ожидаемо не обнаружил необходимых для анализа сетевого трафика программных пакетов и начал автоматическую установку недостающих пакетов (рисунок 38).



```
Hit:1 http://ftp.ru.debian.org/debian bullseye InRelease
Hit:2 http://ftp.ru.debian.org/debian bullseye-updates InRelease
Hit:3 http://security.debian.org/debian-security bullseye-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dns-root-data dnsmasq-base libbcb729-0 libbluetooth3 libc-ares2 libndp0 libnm0 libpcap0.8
  libqt5multimedia5-plugins libqt5multimediagsttools5 libqt5multimediawidgets5 libsmi2ldbl
  libteamdctl0 libwireshark-data libwireshark14 libwiretap11 libwsutil12 ppp wireshark-common
  wireshark-qt
Suggested packages:
  snmp-mibs-downloader geoipupdate geoip-database geoip-database-extra libjs-leaflet
```

Рисунок 38 — Процесс установки недостающих пакетов

Если «start.sh» запустить без прав суперпользователя, то ожидаемо появится сообщение об ошибке, которая представлена на рисунке 39.



```
e-postnikof@nb-it:~/Scripts$ ./start.sh
root permission required
```

Рисунок 39 — Ошибка отсутствия прав суперпользователя при запуске сценария «start.sh»

После установки необходимых пакетов сценарий ожидаемо откроет интерактивное меню. Следующей будет тестироваться функция переключения адаптера беспроводной сети в режим монитора и включения сценария циклической смены каналов на беспроводной сети. Для этого в появившемся меню необходимо нажать цифру «1», а затем клавишу «Ввод».



После нажатия цифры 1 и клавиши ввода сценарий ожидаемо показывает список доступных сетевых интерфейсов и предлагает ввести название необходимого интерфейса (рисунок 40).

```
by exit
Select an option [1-6]: 1
Available network interfaces:
lo          no wireless extensions.

eth0        no wireless extensions.

wlan0       IEEE 802.11bgn ESSID:"Mikrotik_Guest" Nickname:"<WIFI@REALTEK>"
Mode:Managed Frequency:2.412 GHz Access Point: 6E:3B:6B:F3:AA:C1
Bit Rate:300 Mb/s Sensitivity:0/0
Retry:off RTS thr:off Fragment thr:off
Encryption key:****-****-****-****-****-****-****-**** Security mode:open
Power Management:off
Link Quality=57/100 Signal level=-69 dBm Noise level=0 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

Enter name WLAN interface: █
```

Рисунок 40 — Тестирование взаимодействия со сценарием

При вводе не верного имени интерфейса в сценарии происходит ожидаемая ошибка (рисунок 41).

```
TX EXCESSIVE RETRIES:0 INVALID MISC:0

Enter name WLAN interface: 0
Error for wireless request "Set Mode" (8B06) :
    SET failed on device 0 ; No such device.
Monitor mode is enabled
Channel change started
Error for wireless request "Set Frequency" (8B04) :
    SET failed on device 0 ; No such device.
```

Рисунок 41 — Ошибка при вводе неверного имени интерфейса

После ввода верного имени интерфейса сценарий ожидаемо выдаст сообщение об успешном переходе в режим монитора и о старте переключения каналов. Выполнение данного сценария не предусмотрено в фоне, поэтому сценарий закрывать нельзя. Для проверки режима монитора в отдельном окне будет выполнена команда, как на рисунке 42.

```

└─$ iwconfig
lo          no wireless extensions.

eth0        no wireless extensions.

wlan0       unassociated  ESSID:""  Nickname:"<WIFI@REALTEK>"
            Mode:Monitor  Frequency=2.417 GHz  Access Point: Not-Associated

```

Рисунок 42 — Успешная проверка режима монитора

Для проверки сценария «begin-analis(noroot).sh» необходимо запустить сценарий. Для тестирования будет использоваться функция сбора пакетов с фильтром по типу пакета. После запуска сценария, необходимо нажать цифру 4 и клавишу ввода, чтобы появилось интерактивное меню выбора фильтра по типу пакета (рисунок 43).

```

5) Exit
Select an option [1-5]: 4
Welcome to the Wireless Traffic Analysis script

This script helps you analyze wireless traffic.It should
Select the type (subtype) of the wireless signal:

1) Control frame (0)
2) Control frame (1)
3) Data frame (2)
4) Communication request (0x00)
5) Connection setup response (0x01)
6) Reconnect Request (0x02)
7) Reconnect response (0x03)
8) Probing Request (0x04)
9) Response to probing (0x05)
10) Signal packet (0x08)
11) Disconnect (0x0A)
12) Authentication (0x0B)
13) Authentication Denied (0x0C)
14) Action frame (0x0D)
15) Block confirmation requests (0x18)
16) Lock confirmation (0x19)
17) Energy saving poll (0x1A)
18) Transfer Request (0x1B)
19) Ready to receive (0x1C)
20) Reception confirmation (0x1D)
21) End of conflict-free period (0x1E)
22) NULL data (0x24)
23) Quality of Service Data (0x28)
24) Empty quality of service data (0x2C)
25) Exit

```

Рисунок 43 — Меню выбора фильтра типа пакетов

Для примера будет протестирован захват пакетов с пустыми данными. Для этого необходимо ввести цифру «22» и нажать клавишу ввода, чтобы ожидаемо получить список доступных для сканирования сетевых интерфейсов и приглашение ввести название сетевого интерфейса (рисунок 44).

```
#? 22
List interfaces:
Please wait ...
1. eth0
2. any
3. lo (Loopback)
4. wlan0
5. bluetooth-monitor
6. nflog
7. nfqueue
8. dbus-system
9. dbus-session
10. ciscodump (Cisco remote capture)
11. dpauxmon (DisplayPort AUX channel monitor capture)
12. randpkt (Random packet generator)
13. sdjournal (systemd Journal Export)
14. sshdump (SSH remote capture)
15. udpdump (UDP Listener remote capture)
Enter interface name: wlan0
```

Рисунок 44 — Тестирование функции захвата сетевых пакетов по фильтру пустых данных внутри пакета

После ввода имени сетевого интерфейса необходимо указать путь, где будет храниться сетевой дамп. После указания пути (рекомендуется указывать путь к папке, которую индексирует Filebeat) ожидаемо запустится процесс захвата пакетов (Рисунок 45).

```
Capturing on 'wlan0'
** (tshark:14204) 13:14:38.982372 [Main MESSAGE] -- Capture started.
** (tshark:14204) 13:14:38.982482 [Main MESSAGE] -- File: "/tmp/wireshark_wlan07ITNM1.pcapng"
11
```

Рисунок 45 — Процесс сбора пакетов в беспроводной сети с пустыми данными

После помещения дампа необходимо проверить данные в Kibana в раздел «Discover», выставить фильтры по сбору данных за последние 15 минут и убедиться, что данные успешно поступают на хранение (рисунок 46).

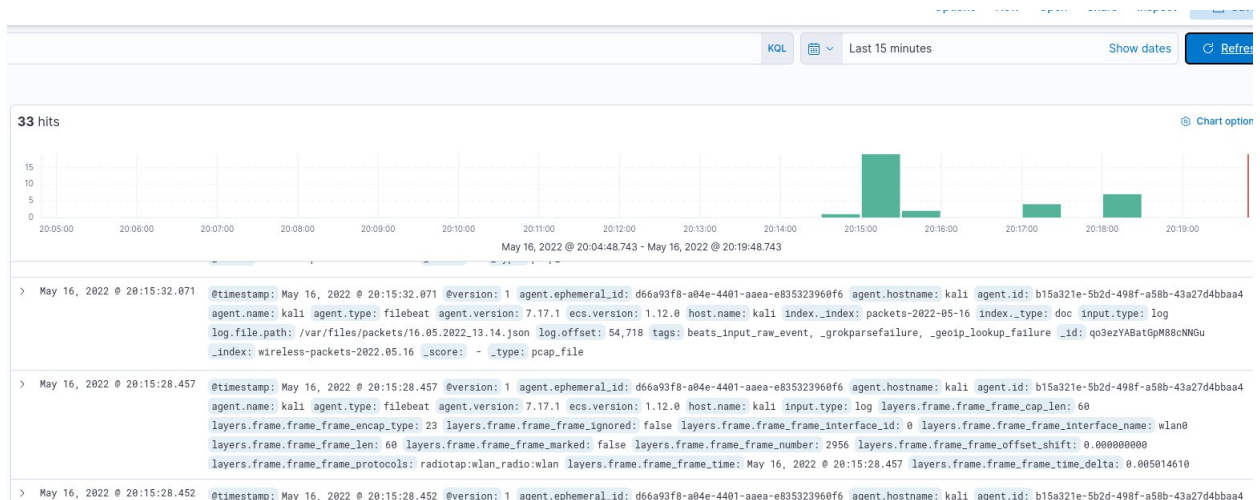


Рисунок 46 — Проверка поступивших данных на сервер Elastic Stack

### 3.4 Анализ результатов тестов. Принятие решения о пригодности архитектуры

Было проведено тестирование прототипа информационной подсистемы исследования трафика в беспроводных сетях, в результате которого выявлено, что решение может не работать на старых версиях операционных систем семейства GNU/Linux, например, Debian 8, CentOS7, Fedora 33, а также исполнение сценариев не поддерживается на дистрибутивах семейства Arch и Oracle Linux.

Алгоритмы захвата, передачи, хранения и распознавания полей в JSON файлах, содержащих информацию о сетевых пакетах, показали себя достаточно хорошо, что позволяет использовать разные сценарии для манипуляции получаемыми данными.

Таким образом прототип информационной подсистемы исследования трафика в беспроводных сетях беспрепятственно выполняет свои функции и является пригодным решением для использования в организации системными администраторами.

## ЗАКЛЮЧЕНИЕ

В результате проделанной работы была разработана информационная подсистема исследования трафика в беспроводных сетях, которая выполняет следующие функции:

1. Перехват сетевых пакетов в беспроводных сетях, с учётом особенностей беспроводных сетей;
2. Хранение перехваченных сетевых пакетов с целью последующего анализа;
3. Обработка и классификация сетевых пакетов в зависимости от содержимого этого пакета.

Интерфейс взаимодействия со сценариями сбора сетевых пакетов не имеет графической составляющей, так как предполагает удобство использования в сочетании с протоколом удалённого доступа SSH.

В перспективе, учитывая модульную структуру сценариев командной оболочки, большого функционала программного обеспечения Wireshark и гибкой системы фильтров Logstash возможна модернизация данной подсистемы с учётом потребностей сбора и анализа трафика отделом системных администраторов. Помимо этого, возможно построить большое количество дашбордов в веб-интерфейсе Kibana для визуализации получаемых в результате взаимодействия с подсистемой данных.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Карпович, Е. Е. Методы тестирования и отладки программного обеспечения : учебник / Е. Е. Карпович. — Москва : Издательский Дом МИСиС, 2020. — 136 с. — ISBN 978-5-907226-64-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/106722.html> (дата обращения: 23.04.2022). — Режим доступа: для авторизир. пользователей
2. Васильев, Р. Б. Управление развитием информационных систем : учебник / Р. Б. Васильев, Г. Н. Калянов, Г. А. Левочкина. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2022. — 507 с. — ISBN 978-5-4497-1654-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/120490.html> (дата обращения: 25.04.2022). — Режим доступа: для авторизир. пользователей
3. Top Programming Languages 2021 (<https://spectrum.ieee.org/top-programming-languages/>) (дата обращения: 26.04.2022)
4. Сузи, Р. А. Язык программирования Python : учебное пособие / Р. А. Сузи. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 350 с. — ISBN 978-5-4497-0705-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/97589.html> (дата обращения: 26.04.2022). — Режим доступа: для авторизир. пользователей
5. Вязовик, Н. А. Программирование на Java : учебное пособие / Н. А. Вязовик. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 601 с. — ISBN 978-5-4497-0852-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102048.html> (дата обращения: 26.04.2022). — Режим доступа: для авторизир. пользователей



6. Золин, А. Г. Программирование на C++ : учебное пособие для СПО / А. Г. Золин, А. Е. Колоденкова, Е. А. Халикова. — Саратов : Профобразование, 2022. — 126 с. — ISBN 978-5-4488-1439-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/116283.html> (дата обращения: 26.04.2022). — Режим доступа: для авторизир. пользователей

7. Рындин, Н. А. Технологии разработки клиентских WEB-приложений на языке JavaScript : учебное пособие / Н. А. Рындин. — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2020. — 54 с. — ISBN 978-5-7731-0888-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/108188.html> (дата обращения: 26.04.2022). — Режим доступа: для авторизир. пользователей

8. Сандерс К. Анализ пакетов: практическое руководство по использованию Wireshark и tcpdump для решения реальных проблем в локальных сетях, 3-е изд. :Пер. с англ. - СПб. : ООО "Диалектика", 2019 - 448 с.ISBN 978-5-6040723-0-1 : ил. - Парал. тит. англ.

9. Операционные системы : учебное пособие для бакалавров / составители И. В. Винокуров. — Москва : Ай Пи Ар Медиа, 2022. — 133 с. — ISBN 978-5-4497-1406-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/115696.html> (дата обращения: 11.05.2022). — Режим доступа: для авторизир. пользователей

10. Операционные системы : учебное пособие для СПО / составители И. В. Винокуров. — Саратов, Москва : Профобразование, Ай Пи Ар Медиа, 2022. — 127 с. — ISBN 978-5-4488-1441-9, 978-5-4497-1444-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/115697.html> (дата обращения: 12.05.2022). — Режим доступа: для авторизир. пользователей

11. Кузнецов, С. Д. Введение в реляционные базы данных : учебное пособие / С. Д. Кузнецов. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 247 с. — ISBN 978-5-4497-0902-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102002.html> (дата обращения: 12.05.2022). — Режим доступа: для авторизир. пользователей
12. Григорьев, Ю. А. Реляционные базы данных и системы NoSQL : учебное пособие / Ю. А. Григорьев, А. Д. Плутенко, О. Ю. Плужникова. — Благовещенск : Амурский государственный университет, 2019. — 425 с. — ISBN 978-5-93493-308-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/103912.html> (дата обращения: 12.05.2022). — Режим доступа: для авторизир. пользователей
13. «Единый комплекс взаимосвязанных программных, аппаратных и телекоммуникационных ресурсов организации» — (<https://cdto.wiki/@4204>)
14. Фомин, Д. В. Информационная безопасность : учебное пособие для СПО / Д. В. Фомин. — Саратов, Москва : Профобразование, Ай Пи Ар Медиа, 2022. — 218 с. — ISBN 978-5-4488-1351-1, 978-5-4497-1565-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/118458.html> (дата обращения: 13.05.2022). — Режим доступа: для авторизир. пользователей. - DOI: <https://doi.org/10.23682/118458>
15. Courage Egbude «Prototyping tools in 2021» (<https://uxplanet.org/prototyping-tools-in-2021-48bb0160ecc3>)
16. Анна Вичугова «5 ключевых достоинств и 3 главных недостатка ELK-стека: разбираемся с Elasticsearch, Logstash и Kibana» — (<https://www.bigdataschool.ru/blog/elk-stack-key-features.html>)



Листинг скрипта start.sh

```
#!/usr/bin/env bash

#Check ROOT
if [ "$(id -u)" != 0 ]
then
    echo root permission required >&2
    exit 1
fi

function loopChannels() {
    ##function switches channels of the wireless network
    echo "Available network interfaces:"
    iwconfig
    read -p "Enter name WLAN interface: " wlan
    iwconfig $wlan mode monitor
    echo "Monitor mode is enabled"
    echo "Channel change started"
    while [[ true ]]; do
        #statements
        for channels in {1..13}
        do
            sleep 2
            iwconfig $wlan channel $channels
        done
    done
}

function onlyMonitor() {
    echo "Available network interfaces:"
    iwconfig
    read -p "Enter name WLAN interface: " wlan
    iwconfig $wlan mode monitor
    echo "Monitor mode is enabled"
    echo ""
    echo "Check:"
    echo ""
    iwconfig
    exit 0
}

function connectWifi() {
    echo "Plese, connect to Wi-fi"
    nmcli radio wifi on
    nmcli dev wifi list
    read -p "Enter SSID: " SSID
    nmcli --ask dev wifi connect $SSID
}

function onlyManaged() {
    echo "Available network interfaces:"
    iwconfig
    read -p "Enter name WLAN interface: " wlan
    iwconfig $wlan mode managed
    echo "Managed mode is enabled"
    echo ""
    echo "Check:"
}
```

```

    echo ""
    iwconfig
    exit 0
}

function manageMenu() {
    echo "Welcome to script!"
    echo ""
    echo "What do you want to do?"
    echo "    1) Enable Monitor mode and run change channel"
    echo "    2) Only enable Monitor mode"
    echo "    3) Connect to Wi-Fi"
    echo "    4) Reinstall the required software"
    echo "    5) Enable Managed mode back"
    echo "    6) Exit"
    until [[ $MENU_OPTION =~ ^[1-6]$ ]]; do
        read -rp "Select an option [1-6]: " MENU_OPTION
    done

    case $MENU_OPTION in
        1)
            loopChannels
            ;;
        2)
            onlyMonitor
            ;;
        3)
            connectWifi
            ;;
        4)
            configOS
            ;;
        5)
            onlyManaged
            ;;
        6)
            exit 0
            ;;
    esac
}

#CheckOS
function checkOS() {
    if [[ -e /etc/debian_version ]]; then
        OS="debian"
        source /etc/os-release

        if [[ $ID == "debian" || $ID == "raspbian" ]]; then
            if [[ $VERSION_ID -lt 9 ]]; then
                echo "Your version of Debian is not
supported."
                echo ""
                echo "However, if you're using Debian >= 9 or
unstable/testing then you can continue, at your own risk."
                echo ""
                until [[ $CONTINUE =~ (y|n) ]]; do
                    read -rp "Continue? [y/n]: " -e CONTINUE
                done
                if [[ $CONTINUE == "n" ]]; then

```

```

                                exit 1
                                fi
                                fi
                                elif [[ $ID == "ubuntu" ]]; then
                                    OS="ubuntu"
                                    MAJOR_UBUNTU_VERSION=$(echo "$VERSION_ID" | cut -
d '.' -f1)
                                    if [[ $MAJOR_UBUNTU_VERSION -lt 16 ]]; then
                                        echo "Your version of Ubuntu is not
supported."
                                        echo ""
                                        echo "However, if you're using Ubuntu >=
16.04 or beta, then you can continue, at your own risk."
                                        echo ""
                                        until [[ $CONTINUE =~ (y|n) ]]; do
                                            read -rp "Continue? [y/n]: " -e CONTINUE
                                        done
                                        if [[ $CONTINUE == "n" ]]; then
                                            exit 1
                                        fi
                                    fi
                                fi
                                elif [[ -e /etc/system-release ]]; then
                                    source /etc/os-release
                                    if [[ $ID == "fedora" || $ID_LIKE == "fedora" ]]; then
                                        OS="fedora"
                                    fi
                                    if [[ $ID == "centos" || $ID == "rocky" || $ID ==
"almalinux" ]]; then
                                        OS="centos"
                                        if [[ ! $VERSION_ID =~ (7|8) ]]; then
                                            echo "Your version of CentOS is not
supported."
                                            echo ""
                                            echo "The script only support CentOS 7 and
CentOS 8."
                                            echo ""
                                            exit 1
                                        fi
                                    fi
                                    if [[ $ID == "ol" ]]; then
                                        OS="oracle"
                                        if [[ ! $OS = "oracle" ]]; then
                                            echo "Oracle Linux is not supported."
                                            echo ""
                                            exit 1
                                        fi
                                    fi
                                    if [[ $ID == "amzn" ]]; then
                                        OS="amzn"
                                        if [[ $VERSION_ID == "2" ]] || [[ $VERSION_ID != "2" ]]; then
                                            echo "Amazon Linux is not supported."
                                            echo ""
                                            exit 1
                                        fi
                                    fi
                                elif [[ -e /etc/arch-release ]]; then
                                    OS=arch

```

```

        echo "Arch Linux is not supported"
        exit 1
    fi
}
function configOS() {
    ##Setting OS
    ##Install packages and other
    checkOS
    if [ $OS = "ubuntu" ] || [ $OS = "debian" ]; then
        add-apt-repository ppa:wireshark-dev/stable -y
        apt update
        apt install wireshark tshark network-manager net-tools wireless-
tools -y
        usermod -aG wireshark $(whoami)
    fi
    if [ $OS = "fedora" ] || [ $OS = "centos" ]; then
        dnf install network-manager wireshark-qt tshark network-manager
net-tools wireless-tools-y
        usermod -aG wireshark $(whoami)
    fi
}
##Check Programs
checkOS
if [ $OS = "ubuntu" ] || [ $OS = "debian" ]; then
#Check tshark
    if [ "$(dpkg -l | grep tshark | awk '{print $2}')" != tshark ]
    then
        configOS
        manageMenu
    else
        manageMenu
    fi
fi
checkOS
if [ $OS = "fedora" ] || [ $OS = "centos" ]; then
#Check tshark
    if [ "$(rpm -q wireshark)" = 0 ]
    then
        configOS
    else
        manageMenu
    fi
fi
###If configOS
#echo "Do you need Monitor mode?(y or n)"
#read answer
#if [ $answer == 'y' ] || [ $answer == 'yes' ]; then
#loopChannels ## to analyze traffic, you need to change channels
#fi

#if [ $answer == 'n' ] || [ $answer == 'no' ]; then
#connectWifi
#fi

```

Листинг скрипта begin-analis(no root).sh

```
#!/usr/bin/env bash

#CheckOS
function checkOS() {
    if [[ -e /etc/debian_version ]]; then
        OS="debian"
        source /etc/os-release

        if [[ $ID == "debian" || $ID == "raspbian" ]]; then
            if [[ $VERSION_ID -lt 9 ]]; then
                echo "Your version of Debian is not
supported."

                echo ""
                echo "However, if you're using Debian >= 9 or
unstable/testing then you can continue, at your own risk."
                echo ""
                until [[ $CONTINUE =~ (y|n) ]]; do
                    read -rp "Continue? [y/n]: " -e CONTINUE
                done
                if [[ $CONTINUE == "n" ]]; then
                    exit 1
                fi
            fi
        elif [[ $ID == "ubuntu" ]]; then
            OS="ubuntu"
            MAJOR_UBUNTU_VERSION=$(echo "$VERSION_ID" | cut -
d '.' -f1)

            if [[ $MAJOR_UBUNTU_VERSION -lt 16 ]]; then
                echo "Your version of Ubuntu is not
supported."

                echo ""
                echo "However, if you're using Ubuntu >=
16.04 or beta, then you can continue, at your own risk."
                echo ""
                until [[ $CONTINUE =~ (y|n) ]]; do
                    read -rp "Continue? [y/n]: " -e CONTINUE
                done
                if [[ $CONTINUE == "n" ]]; then
                    exit 1
                fi
            fi
        fi
    elif [[ -e /etc/system-release ]]; then
        source /etc/os-release
        if [[ $ID == "fedora" || $ID_LIKE == "fedora" ]]; then
            OS="fedora"
        fi
        if [[ $ID == "centos" || $ID == "rocky" || $ID ==
"almalinux" ]]; then
            OS="centos"
            if [[ ! $VERSION_ID =~ (7|8) ]]; then
                echo "Your version of CentOS is not
supported."

                echo ""
```

```

        echo "The script only support CentOS 7 and
CentOS 8."

        echo ""
        exit 1
    fi
fi
if [[ $ID == "ol" ]]; then
    OS="oracle"
    if [[ ! $OS = "oracle" ]]; then
        echo "Oracle Linux is not supported."
        echo ""
        exit 1
    fi
fi
if [[ $ID == "amzn" ]]; then
    OS="amzn"
    if [[ $VERSION_ID == "2" ]] || [[ $VERSION_ID != "2" ]]; then
        echo "Amazon Linux is not supported."
        echo ""
        exit 1
    fi
fi
elif [[ -e /etc/arch-release ]]; then
    OS=arch
    echo "Arch Linux is not supported"
    exit 1
fi
}

function message_Stop() {
    #STOP message
    echo ""
    echo "Process started"
    echo "Press Ctrl + C to STOP"
    echo ""
}

function tshark_all() {
    #This function is useful when working in monitore mode
    echo "List interfaces: "
    echo "Please wait..."
    tshark -D
    read -p "Enter interface name: " IFname
    read -p "Enter the path to save the dump (default is
'/home/$(whoami): " path_d
    if [ -z $path_d ]
    then
        message_Stop
        tshark -i $IFname -T ek > $HOME/$(date +%d.%m.%Y_%H.%M').json
    else
        message_Stop
        tshark -i $IFname -T ek > $path_d/$(date
+%d.%m.%Y_%H.%M').json
    fi
}

function tshark_filter_MAC() {
    #This function filters on MAC address

```

```

        message_Stop
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter MAC address: " MAC_addr
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.bssid == $(echo $MAC_addr | tr "[A-
Z]" "[a-z]")" -T ek > $HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.bssid == $(echo $MAC_addr | tr "[A-
Z]" "[a-z]")" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }

function filter_Channel() {
    #Only traffic on a specific channel
    echo ""
    echo "Welcome!"
    echo "To analyze packets on a specific channel, enter a number
between 1 and 13, where the number will be the channel number."
    echo ""
    echo "Please stop the 'start.sh' script if it is currently
looping through the wireless channels, put the adapter into Monitor mode
"

    echo ""
    echo "Available network interfaces: "
    tshark -D
    echo ""
    read -p "Enter Name Interface: " IFname
    read -p "Enter number channel : " num
    echo ""

    if
        [ $num -lt 1 ] || [ $num -gt 13 ] ;
    then
        echo "ERROR! Enter number between 1 and 13"
        echo "The script will run again..."
        sleep 3
        filter_Channel
    fi

    read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
    if [ -z $path_d ]
    then
        message_Stop
        tshark -i $IFname -Y "wlan_radio.channel == $num" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
    else
        message_Stop
        tshark -i $IFname -Y "wlan_radio.channel == $num" -T ek >
$path_d/$(date +%d.%m.%Y_%H.%M').json
    fi
}

function startMenu() {

```

```

        echo "Welcome to the Wireless Traffic Analysis script"
        echo ""
        echo "This script helps you analyze wireless traffic.It should
run second. "
        echo ""
        echo "What do you want to do?"
        echo "    1) Dump all networks (requires a lot of memory)"
        echo "    2) Dump by MAC Address"
        echo "    3) Dump all networks by channel"
        echo "    4) Dump by network packet type"
        echo "    5) Exit"
        until [[ $MENU_OPTION =~ ^[1-5]$ ]]; do
            read -rp "Select an option [1-5]: " MENU_OPTION
        done

        case $MENU_OPTION in
            1)
                tshark_all
                ;;
            2)
                tshark_filter_MAC
                ;;
            3)
                filter_Channel
                ;;
            4)
                filter_TypePacketsMenu
                ;;
            5)
                exit 0
                ;;
        esac
    }

    function fcType0 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type == 0" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type == 0" -T ek >
$path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }

    function fcType1 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D

```



```

        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type == 1" -T ek > $HOME/$(date
+'%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type == 1" -T ek >
$path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcType2 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type == 2" -T ek > $HOME/$(date
+'%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type == 2" -T ek >
$path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubType0 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x00" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x00" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubType01 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname

```

```

        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x01" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x01" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype02 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x02" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x02" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype03 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x03" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x03" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype04 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname

```

```

        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x04" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x04" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype05 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x05" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x05" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype08 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x08" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x08" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype00A () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname

```

```

        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x0A" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x0A" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype00B () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x0B" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x0B" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype00C () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x0C" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x0C" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype00D () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname

```

```

        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x0D" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x0D" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype018 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x18" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x18" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype019 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x19" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x19" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype01A () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname

```

```

        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x1A" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x1A" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype01B () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x1B" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x1B" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype01C () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x1C" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x1C" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype01D () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname

```

```

        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x1D" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x1D" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype01E () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x1E" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x1E" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype024 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x24" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x24" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype028 () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname

```

```

        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x24" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x24" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }
    function fcSubtype02C () {
        #This function is useful when working in monitore mode
        echo "List interfaces: "
        echo "Please wait..."
        tshark -D
        read -p "Enter interface name: " IFname
        read -p "Enter the path to save the dump (default is
'/home/$(whoami)): " path_d
        if [ -z $path_d ]
        then
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype == 0x2C" -T ek >
$HOME/$(date +%d.%m.%Y_%H.%M').json
        else
            message_Stop
            tshark -i $IFname -Y "wlan.fc.type_subtype ==
0x2C" -T ek > $path_d/$(date +%d.%m.%Y_%H.%M').json
        fi
    }

    function filter_TypePacketsMenu() {
        echo "Welcome to the Wireless Traffic Analysis script"
        echo ""
        echo "This script helps you analyze wireless traffic.It should
run second. "
        echo ""
        echo "Select the type (subtype) of the wireless signal:"
        echo ""
        select numbers in "Control frame (0)" "Control frame (1)"
"Data frame (2)" "Communication request (0x00)" "Connection setup
response (0x01)" "Reconnect Request (0x02)" "Reconnect response (0x03)"
"Probing Request (0x04)" "Response to probing (0x05)" "Signal packet
(0x08)" "Disconnect (0x0A)" "Authentication (0x0B)" "Authentication
Denied (0x0C)" "Action frame (0x0D)" "Block confirmation requests
(0x18)" "Lock confirmation (0x19)" "Energy saving poll (0x1A)" "Transfer
Request (0x1B)" "Ready to receive (0x1C)" "Reception confirmation
(0x1D)" "End of conflict-free period (0x1E)" "NULL data (0x24)" "Quality
of Service Data (0x28)" "Empty quality of service data (0x2C)" "Exit"
        do
            case $numbers in
                "Control frame (0)" )          fcType0 ;;
                "Control frame (1)" )          fcType1 ;;
                "Data frame (2)" )              fcType2 ;;
                "Communication request (0x00)" ) fcSubType0 ;;
                "Connection setup response (0x01)" ) fcSubType01 ;;
            esac
        done
    }

```



```

        "Reconnect Request (0x02)" ) fcSubtype02 ;;
        "Reconnect response (0x03)"      fcSubtype03 ;;
        "Probing Request (0x04)"      fcSubtype04 ;;
        "Response to probing (0x05)"    fcSubtype05 ;;
        "Signal packet (0x08)"      fcSubtype08 ;;
        "Disconnect (0x0A)"    fcSubtype00A ;;
        "Authentication (0x0B)" fcSubtype00B ;;
        "Authentication Denied (0x0C)" fcSubtype00C ;;
        "Action frame (0x0D)"      fcSubtype00D ;;
        "Block confirmation requests (0x18)" fcSubtype018 ;;
        "Lock confirmation (0x19)"      fcSubtype019 ;;
        "Energy saving poll (0x1A)"      fcSubtype01A ;;
        "Transfer Request (0x1B)" fcSubtype01B ;;
        "Ready to receive (0x1C)" fcSubtype01C ;;
        "Reception confirmation (0x1D)" fcSubtype01D ;;
        "End of conflict-free period (0x1E)" fcSubtype01E ;;
        "NULL data (0x24)"      fcSubtype024 ;;
        "Quality of Service Data (0x28)"      fcSubtype028 ;;
        "Empty quality of service data (0x2C)" fcSubtype02C ;;
        "Exit"      exit 0 ;;
    esac
done
}

###END FUNCTIONS###

##Check Programs
checkOS
if [ $OS = "ubuntu" ] || [ $OS = "debian" ]; then
#Check tshark
    if [ "$(dpkg -l | grep tshark | awk '{print $2}')" != tshark ]
    then
        echo "First run start.sh"
        exit 1
    else
        startMenu
        echo 0
    fi
fi
checkOS
if [ $OS = "fedora" ] || [ $OS = "centos" ]; then
#Check tshark
    if [ "$(rpm -q wireshark)" = 0 ]
    then
        echo "First run start.sh"
        exit 1
    else
        echo 0
        startMenu
    fi
fi
fi

```

### Порядок действий для установки Elastic Stack

1. Копируем публичный ключ репозитория с помощью команды:  
`wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -`
2. Если нет пакета `apt-transport-https`, то надо установить с помощью команды:  
`apt install apt-transport-https`
3. Добавляем репозиторий Elasticsearch в систему с помощью команды:  
`echo "deb [trusted=yes] https://mirror.yandex.ru/mirrors/elastic/7/ stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list`
4. Устанавливаем Elasticsearch на Debian или Ubuntu с помощью команды:  
`apt update && apt install elasticsearch`
5. После установки необходимо добавить `elasticsearch` в автозагрузку с помощью команды:  
`systemctl daemon-reload`  
`systemctl enable elasticsearch.service`  
`systemctl start elasticsearch.service`
6. Проверяем, запустился ли Elasticsearch с помощью команд:  
`systemctl status elasticsearch.service`
7. Запускаем установку Kibana:  
`apt update && apt install kibana`
8. Добавляем Кибана в автозагрузку и запускаем с помощью команд:  
`systemctl daemon-reload`  
`systemctl enable kibana.service`  
`systemctl start kibana.service`
9. Установка Filebeat для отправки логов в Logstash с помощью команды:  
`apt install filebeat`

## Содержание конфигурационного файла «logstash-wlan-analysis-es.conf»

```

input {
  beats {
    port => 5400
    ssl => true
    ssl_certificate_authorities => ["/etc/elk-certs/elk-
ssl.crt"]
    ssl_certificate => "/etc/elk-certs/elk-ssl.crt"
    ssl_key => "/etc/elk-certs/elk-ssl.key"
    ssl_verify_mode => "force_peer"
  }
}
filter {
  # Drop Elasticsearch Bulk API control lines
  if ([message] =~ "{\"index\"}) {
    drop {}
  }
  json {
    source => "message"
    remove_field => "message"
  }
  # Extract innermost network protocol
  grok {
    match => {
      "[layers][frame][frame_frame_protocols]" =>
"%{WORD:protocol}$"
    }
  }
  #White List
  prune {
    whitelist_names => [ '"^hostname$"', 'id',
'[type]', '[interface_name]', '[frame_time]',
'[radiotap_channel_flags]', '[present_word]',
'[wlan_radio_channel]', '[htex_capabilities_transtime]',
'[ssid]', '[wlan_addr]', '[wlan_addr_resolved]', '[bssid]',
'[wlan_da]', '[wlan_fc_ds]', '[fc_type]', '[fcs_status]',
'[protocol]', '[timestamp]' ]
  }
  date {
    match => [ "timestamp", "UNIX_MS" ]
  }
}
output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "wireless-packets-%{+YYYY.MM.dd}"
    document_type => "pcap_file"
    manage_template => false
  }
}

```