Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра «Вычислительная техника»

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О.В. Непомнящий

«___» _____ 2022 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Программная среда подготовки данных для анализа трафика
тема

Руководитель _____ доцент, канд.техн.наук Ф.А. Казаков
подпись, дата

Выпускник _____ С.С. Овсянников
подпись, дата

Нормоконтролер _____ доцент, канд.техн.наук Ф.А. Казаков
подпись, дата

Красноярск 2022

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра «Вычислительная техника»

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О.В. Непомнящий

«___» _____ 2022 г.

ЗАДАНИЕ

НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

в форме бакалаврской работы

Студенту Овсянникову Сергей Сергеевичу
фамилия, имя, отчество

Группа КИ18-06Б Направление (специальность) 09.03.01
номер код

Информатика и вычислительная техника
полное наименование

Тема выпускной квалификационной работы Программная среда подготовки данных для анализа трафика

Утверждена приказом по университету № _____ от _____

Руководитель ВКР Ф.А. Казаков преподаватель кафедры ВТ
инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР: Провести анализ известного программного обеспечения для анализа трафика сети. Определить возможные способы получения данных трафика сети. Разработать приложение для подготовки данных к дальнейшему анализу.

Перечень разделов ВКР: Анализ задания для выполнения и анализ существующих аналогов, структура программной среды подготовки данных трафика, программная реализация, пример работы.

Перечень графического материала: презентация в формате PowerPoint.

Руководитель ВКР _____ Ф.А. Казаков
подпись инициалы и фамилия

Задание принял к исполнению С.С. Овсянников
подпись, инициалы и фамилия студента

« ___ » _____ 2022 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Программная среда подготовки данных для анализа трафика» содержит 47 страниц текстового документа, 23 иллюстрации, 9 использованных источников.

Цель работы: написание приложения для упрощения процедуры анализа трафика локальной сети.

При выполнении данной работы был произведен обзор предметной области, задания на выпускную квалификационную работу, изучены существующие аналоги и сформированы требования, предъявляемые к приложению.

Объект работы – приложение, позволяющее внести данные полученные на различных участках локальной сети путем записи трафика с помощью агентов. Анализ полученных данных в нашем приложение.

Задачи:

- осуществить выбор программных средств моделирования и разработки приложения;
- выполнить моделирование разрабатываемого приложения;
- выполнить программную реализацию приложения для платформ Windows;
- проанализировать полученные результаты работы.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В настоящее время часто возникает необходимость по обработки данных сетевого трафика для удобной оценки нагрузки сети. Разрабатываемое программное обеспечение позволит с удобством управлять пакетами данных сети и производить анализ трафика.

Актуальностью данной темы является то, что другие сервисы не предоставляют возможность анализировать трафик на всем его сетевом пути.

Целью данной работы является разработка программного обеспечения для упрощения подготовки данных за счет снижения объемов (снижение размерности) и выделения временных и неявных зависимостей для дальнейшего анализа трафика.

1 Анализ задания для выполнения

Актуальность данной темы обусловлена тем, что на текущий момент времени активно разрабатываются, применяются и используются различные методы по обнаружению произошедших и предотвращению будущих вторжений, но они далеко не всегда являются эффективными на практике. В последствие этого всего все технологии защиты постоянно изучаются и улучшаются.

В моей работе необходимо разработать приложение для обработки сетевых пакетов данных которые будет получать пользователь с сетевых устройств. Агенты в критических точках локальной сети должны прослушивать трафик и записывать в файлы для дальнейшей передачи и обработки. Приложение должно будет сравнивать характер, основные параметры и отдельные пакеты с различных точек локальной сети и помогать пользователю в дельнейшем проводить анализ с целью поиска скрытых зависимостей, сетевых потерь, возможных уязвимостей и других.

Первой функцией приложения должна быть возможность захвата пакетов данных с сетевых устройств по маршруту с помощью нашего приложения или другого похожего с таким функционалом, путем зеркалирования трафика с устройства на которым установлены агенты по захвату данных сети, которую мы хотим анализировать.

После получения необходимого количества данных агентами, пользователь собирает эту информацию и помещает в нашу основную программу, где мы уже сможем упростить процесс поиска несанкционированного доступа, использования, раскрытия, искажения, изменения, исследования, записи или уничтожения информации. После того как пользователь получит все необходимую информацию о том, что происходило в сети, он сможет предотвратить будущие попытки несанкционированного доступа в сеть путем расследования произошедших инцидентов, тем самым сможет защитить данные своей локальной сети.

1.2 Цель создания приложения

Основной целью создания приложения является разработка прототипа распределенной системы для анализа структуры сетевого трафика и оценки эффективности использования компьютерных сетей.

- разработка методов подготовки данных сети к дальнейшему анализу для определения структуры сетевого трафика, эффективности использования ресурсов;
- разработка агентов сбора и алгоритмов предварительной обработки сетевого трафика.

Современные компоненты сетевой инфраструктуры используют большое количество управляющих протоколов и сервисов, полной настройке которых не уделяется должного внимания. Это приводит к неэффективной загрузке сетевых ресурсов и снижению информационной безопасности.

Актуально создание системы, которая позволила бы оценить структуру трафика, наличие не продуктивной активности и другие параметры в сегментах сети и выработать рекомендации для повышения эффективности использования сетей и снижению риска угроз.

Система будет ориентирована на предприятия, специализирующиеся на аудите и удаленном сопровождении информационной и телекоммуникационной составляющей бизнеса, интернет-сервис провайдеров, а также предприятия большого и среднего бизнеса имеющие компьютерные сети.

1.3 Анализ существующих аналогов

На данный момент существует несколько различных приложений, предоставляющих функции по анализу сетевого трафика, поэтому требуется рассмотреть существующие решения, чтобы выделить их преимущества и недостатки.

1.3.1 Анализатор трафика WireShark

WireShark это относительно новый инструмент в области решений для сетевой диагностики и анализа, но несмотря на то, что данный продукт еще молод это не помешало получить себе признание и уважение со стороны ИТ-профессионалов.

WireShark превосходно справляется с анализом трафика, прекрасно выполняя для нас необходимую работу. Администраторы сетей смогли получить великолепный продукт, имеющий середину между работой с исходными данными и визуальным представлением этих данных в имеющимся интерфейсе, поэтому в WireShark мы не сможем увидеть перекосов в ту или другую сторону, которые можно увидеть в большей части других подобных решений для анализа и подготовки к анализу сетевого трафика. WireShark достаточно прост, совместим со многими системами и портативен. Пользователи WireShark, получают именно тот необходимый функционал, что и хотят, и получают это быстро.

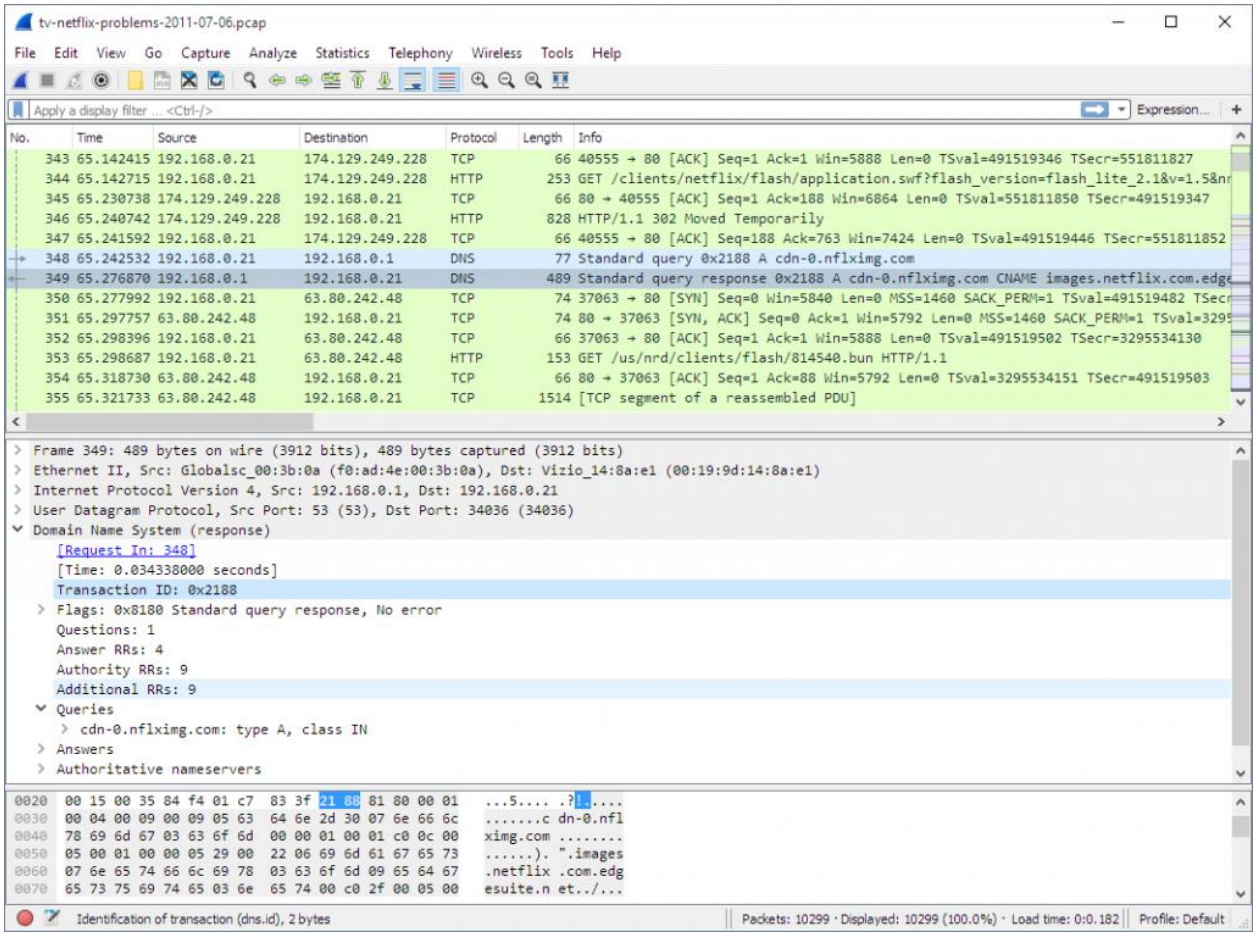


Рисунок 1 – Анализатор трафика Wireshark

WireShark обладает вполне удобным пользовательским интерфейсом, достаточно большим количеством опций и функционалом для фильтрации и сортировки, но не для глубокого анализа трафика. Большинство пользователей данной программы могут оценить данную программу, анализ трафика WireShark хорошо работает со всеми популярными операционными системами, например: — *NIX, Windows и macOS.

Отличие данной технологии от нашей в том, что мы можем проводить анализ сетевых данных лишь на одном конкретном участке сети, а не на всем маршруте данных.

1.3.2 Анализатор трафика tcpdump

Внешне анализатор трафика tcpdump представляет из себя некий инструмент, который использовался десятилетие назад, и по правде говоря, если рассматривать его с точки зрения функциональных возможностей работает он также как достаточно старое программное обеспечение. Невзирая на то, что с поставленными он справляется, при этом используя для своих возможностей минимальное количество ресурсов системы, сводя к минимуму нагрузку на системы насколько на сколько это вообще возможно в условия работы программного обеспечения.

Большей части современных пользователей будет очень сложно разобраться в большом разнообразии таблиц с данными сетевого трафика имеющим далеко не всю нужную информацию. Но все же бывают ситуации, когда подобное программное обеспечение может помочь в решение какой-либо проблемы, использование облегченных и не требовательным к ресурсам решений сможет быть полезно на практике. В некоторых средах или на достаточно слабом по техническим характеристикам ПК минимализм может оказаться единственным возможным к работе вариантом.

Программное обеспечение tcpdump было разработано для среды *NIX, но на текущей момент времени оно также хорошо работает с некоторыми другими портами Windows. Оно имеет весь базовый функционал, который вы можете увидеть в любом другом анализаторе трафика — захват, запись и т.д.

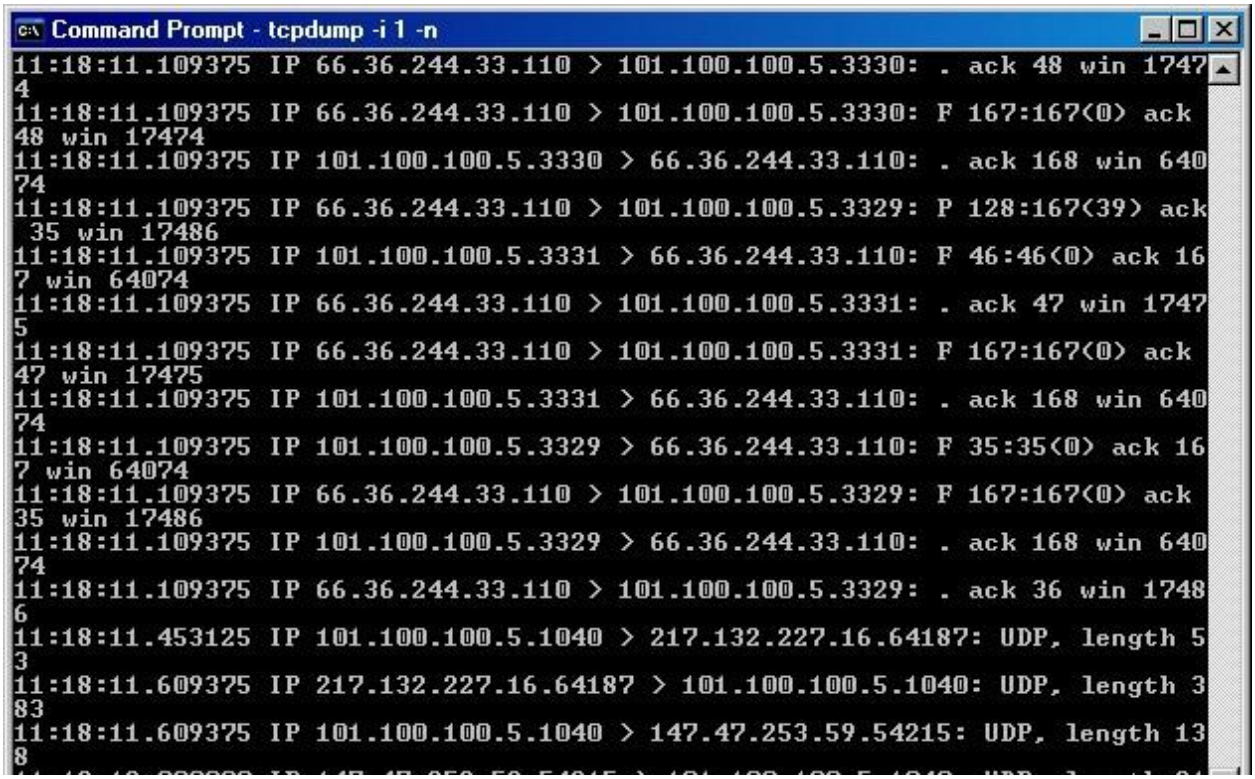


Рисунок 2 – Анализатор трафика tcpdump

В данном случае были выявлен недостаток в том, что программа обладает не удобным интерфейсом и отсутствием захвата пакетов в реальном времени.

1.3.3 Анализатор трафика Kismet

Kismet — это дополнительный пример программной среды с открытым исходным кодом, созданной для решения конкретных задач, возникающих в нашей сети. Kismet не ограничивается на анализе сетевого трафика, он дает нам куда более расширенные функциональные возможности. Например, данное программное обеспечение способно осуществлять анализ трафика скрытых и беспроводных сетей, которые не транслируют свои идентификаторы SSID. Этот инструмент будет сильно полезен, когда в нашей беспроводной сети есть что-то, что может вызывать проблемы, но быстро обнаружить их источник у нас не получается. Kismet точно сможет помочь нам обнаружить неавторизированную сеть или точку доступа, которая существует, но имеет не правильные настройки.

Рисунок 3 – Анализатор трафика Kismet

1.3.4 Анализатор трафика EtherApe

Своими функциональными способностями EtherApe достаточно сильно похож на WireShark, он точно так же является программной средой с открытым исходным кодом и распространяется бесплатно. Но, он действительно очень сильно отличается на фоне других программных обеспечений — главное отличие то что он ориентирован на визуальное представление данных с помощью графических возможностей интерфейса.

Если результаты WireShark просматриваем в привычном цифровом и табличном формате, то весь трафик EtherApe демонстрируется с использованием продвинутого графического интерфейса, каждая вершина графа олицетворяет собой отдельное устройство, размеры этих самых вершин и ребер показывают нам размер сетевого трафика на данном устройстве, а цветовыми маркерами отмечаются различные протоколы, которые удалось получить. Пользователи, которые предпочитают визуальное восприятие статистической информации, используют анализатор EtherApe. Доступен для сред UNIX и macOS.

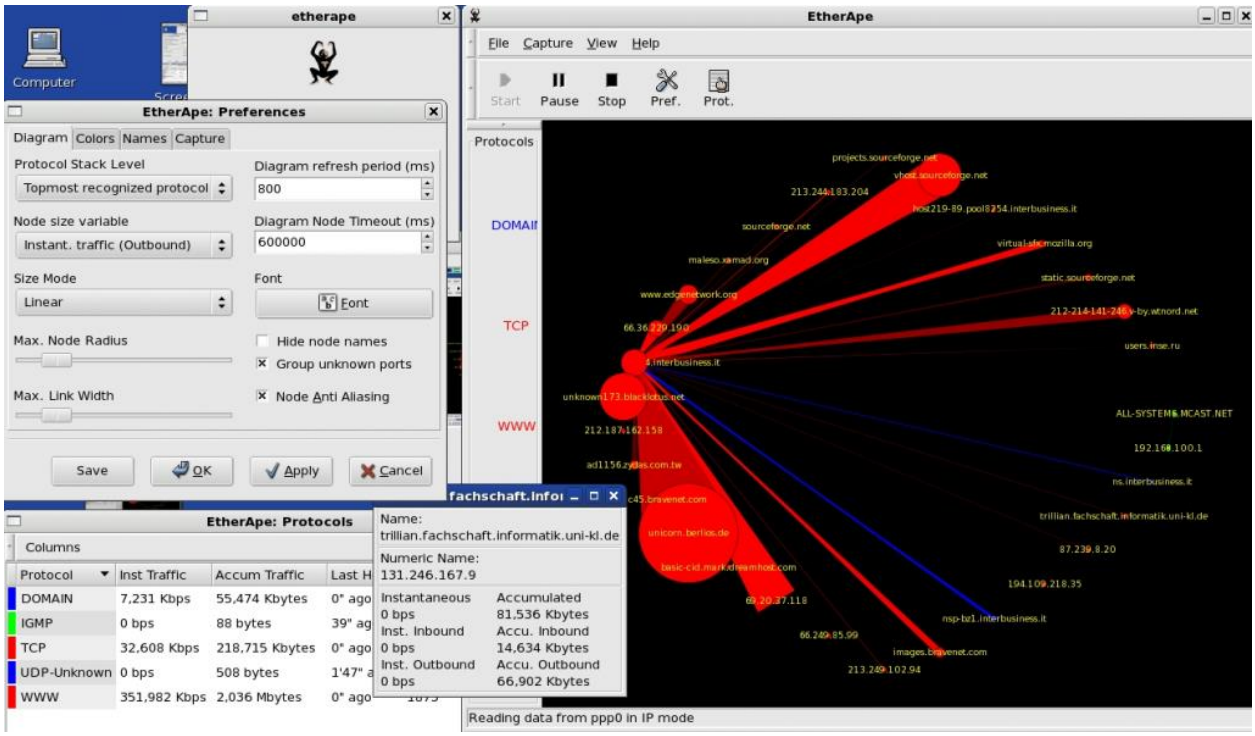


Рисунок 4 – Анализатор трафика EtherApe

1.3.5 Анализатор трафика Cain and Abel

У программной среды Cain and Abel, способность анализа трафика являет больше дополнительной или вспомогательной функцией, чем основной. Когда задачи пользователей является не просто анализ трафика, то в большинстве случаев вспоминают именно это программное обеспечение. С помощью данной программы мы получаем возможность восстанавливать утерянные пароли для операционной системы Windows, осуществлять «атаки» для нахождения утерянных учетных данных, получать данные VoIP в сети, производить анализ и маршрутизацию пакетов, и некоторые другие возможности.

Cain and Abel является мощным инструментом для опытного системного администратора с большим количеством полномочий. Недостатком является то что работать он может только в среде Windows.

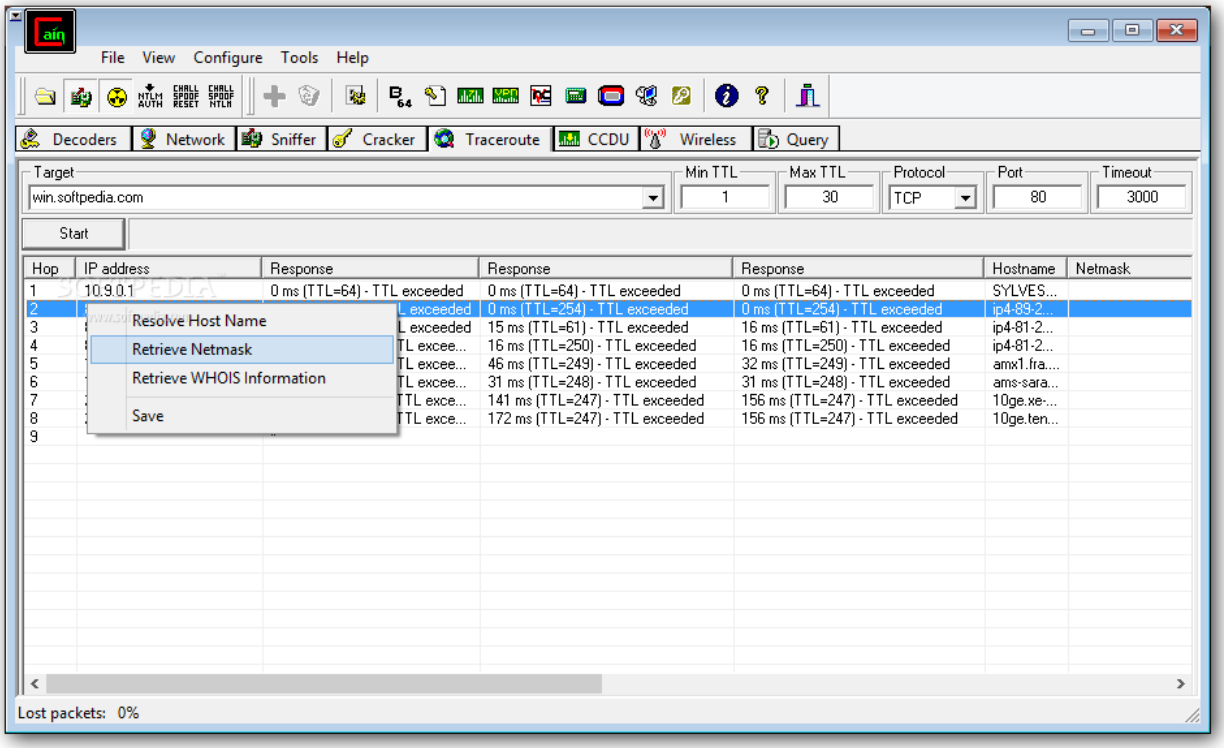


Рисунок 5 – Анализатор трафика Cain and Abel

Данная программа обладает функциями отслеживания данных пользователей и уязвимостей сети, но мы не можем полностью контролировать все участки сети.

1.3.6 Анализатор трафика NetworkMiner

Программное обеспечение NetworkMiner — еще одно решение, чьи функциональные возможности превосходят рамки анализатора трафика. Большинство других анализаторов трафика работают на такие параметры как отправка и получение пакетов, NetworkMiner обращает внимание на то, кто и как осуществляет эти отправки и получение. Данное программное обеспечение подходит для выявления проблемных компьютеров или пользователей в нашей сети, а не для глубокого анализа данных, мониторинга или диагностики сети как таковой. Как и прошлое программное решение NetworkMiner разработан только для ОС Windows.

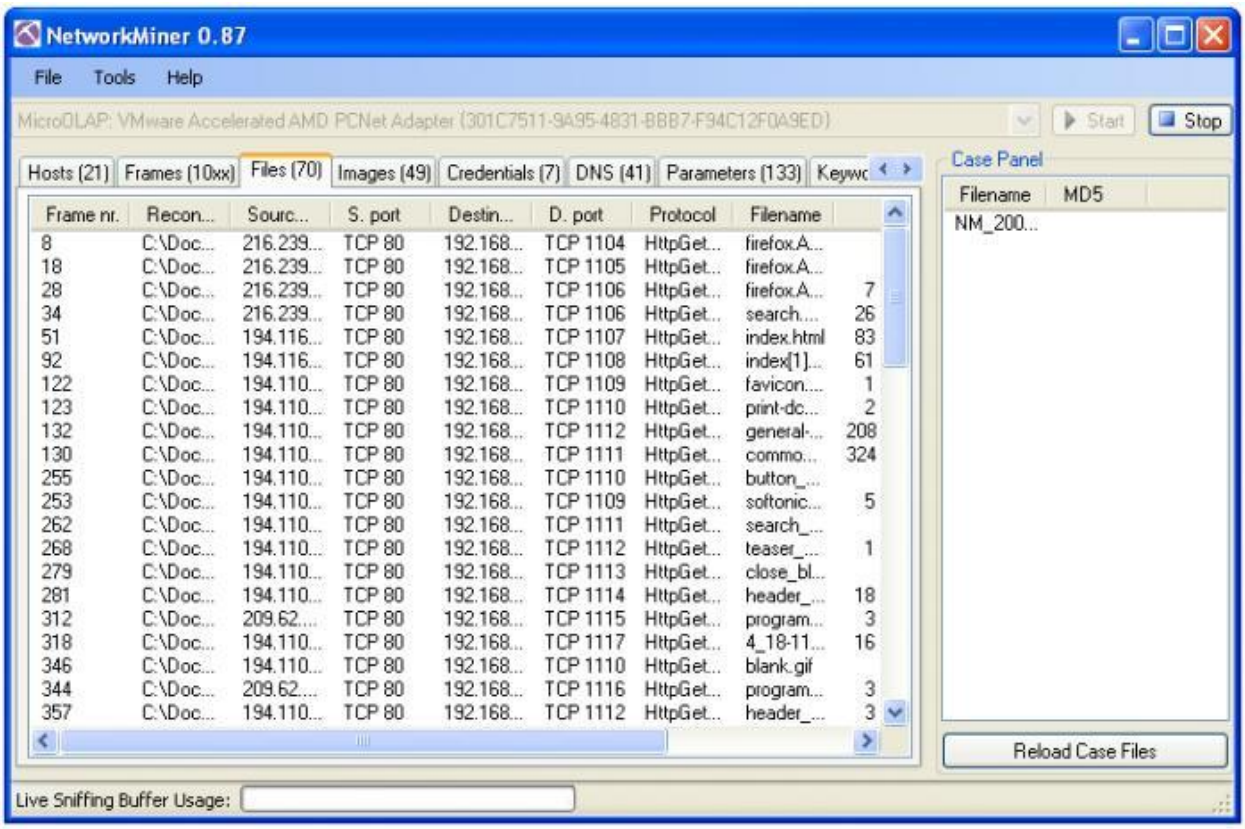


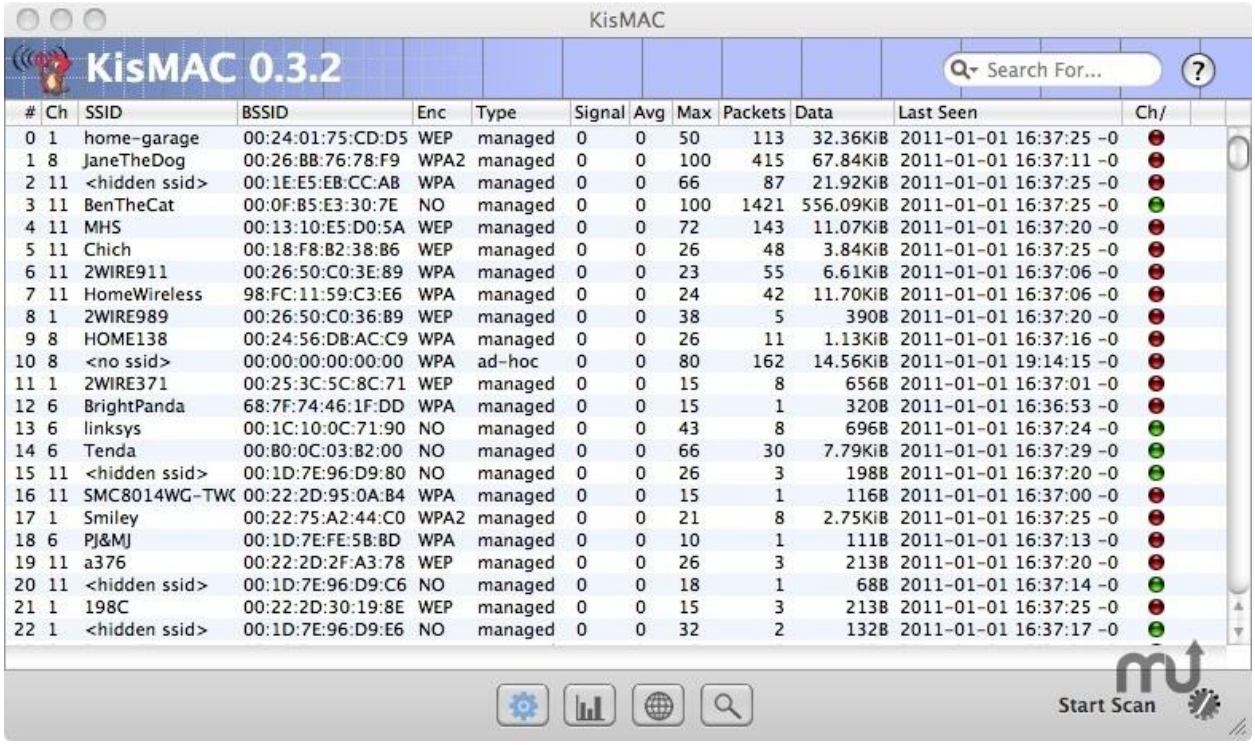
Рисунок 6 – Анализатор трафика NetworkMiner

Данная программа помогает проанализировать пропускную способность сети и устройств сети, это не всегда поможет нам обнаружить уязвимости.

1.3.7 Анализатор трафика KisMAC

KisMAC — это описанный ранее Kismet, только более удобно оформленный для macOS. На данный момент Kismet имеет порт только для операционной системы macOS, из-за этого мы можем посчитать что существование KisMAC будет ненужным, но нам необходимо обратить внимание на такой факт, как программное обеспечение KisMAC имеет свою кодовую базу и не относится к явным производным анализатора Kismet. Так же необходимо обратить внимание на то, что KisMAC дает нам некоторые редкие возможности, такие как нанесение на карту местоположения устройств в сети или атака де аутентификации на операционной системе macOS.

Kismet не предоставляет данный функционал. Данные уникальные особенности в очень редких ситуациях смогут перевесить выбор программного обеспечения пользу именно этого решения.



#	Ch	SSID	BSSID	Enc	Type	Signal	Avg	Max	Packets	Data	Last Seen	Ch/
0	1	home-garage	00:24:01:75:CD:D5	WEP	managed	0	0	50	113	32.36KiB	2011-01-01 16:37:25 -0	
1	8	JaneTheDog	00:26:BB:76:78:F9	WPA2	managed	0	0	100	415	67.84KiB	2011-01-01 16:37:11 -0	
2	11	<hidden ssid>	00:1E:E5:EB:CC:AB	WPA	managed	0	0	66	87	21.92KiB	2011-01-01 16:37:25 -0	
3	11	BenTheCat	00:0F:B5:E3:30:7E	NO	managed	0	0	100	1421	556.09KiB	2011-01-01 16:37:25 -0	
4	11	MHS	00:13:10:E5:D0:5A	WEP	managed	0	0	72	143	11.07KiB	2011-01-01 16:37:20 -0	
5	11	Chich	00:18:F8:B2:38:B6	WEP	managed	0	0	26	48	3.84KiB	2011-01-01 16:37:25 -0	
6	11	2WIRE911	00:26:50:C0:3E:89	WPA	managed	0	0	23	55	6.61KiB	2011-01-01 16:37:06 -0	
7	11	HomeWireless	98:FC:11:59:C3:E6	WPA	managed	0	0	24	42	11.70KiB	2011-01-01 16:37:06 -0	
8	1	2WIRE989	00:26:50:C0:36:B9	WEP	managed	0	0	38	5	390B	2011-01-01 16:37:20 -0	
9	8	HOME138	00:24:56:DB:AC:C9	WPA	managed	0	0	26	11	1.13KiB	2011-01-01 16:37:16 -0	
10	8	<no ssid>	00:00:00:00:00:00	WPA	ad-hoc	0	0	80	162	14.56KiB	2011-01-01 19:14:15 -0	
11	1	2WIRE371	00:25:3C:5C:8C:71	WEP	managed	0	0	15	8	656B	2011-01-01 16:37:01 -0	
12	6	BrightPanda	68:7F:74:46:1F:DD	WPA	managed	0	0	15	1	320B	2011-01-01 16:36:53 -0	
13	6	linksys	00:1C:10:0C:71:90	NO	managed	0	0	43	8	696B	2011-01-01 16:37:24 -0	
14	6	Tenda	00:B0:0C:03:B2:00	NO	managed	0	0	66	30	7.79KiB	2011-01-01 16:37:29 -0	
15	11	<hidden ssid>	00:1D:7E:96:D9:80	NO	managed	0	0	26	3	198B	2011-01-01 16:37:20 -0	
16	11	SMC8014WG-TWC	00:22:2D:95:0A:B4	WPA	managed	0	0	15	1	116B	2011-01-01 16:37:00 -0	
17	1	Smiley	00:22:75:A2:44:C0	WPA2	managed	0	0	21	8	2.75KiB	2011-01-01 16:37:25 -0	
18	6	PJ&MJ	00:1D:7E:FE:5B:BD	WPA	managed	0	0	10	1	111B	2011-01-01 16:37:13 -0	
19	11	a376	00:22:2D:2F:A3:78	WEP	managed	0	0	26	3	213B	2011-01-01 16:37:20 -0	
20	11	<hidden ssid>	00:1D:7E:96:D9:C6	NO	managed	0	0	18	1	68B	2011-01-01 16:37:14 -0	
21	1	198C	00:22:2D:30:19:8E	WEP	managed	0	0	15	3	213B	2011-01-01 16:37:25 -0	
22	1	<hidden ssid>	00:1D:7E:96:D9:E6	NO	managed	0	0	32	2	132B	2011-01-01 16:37:17 -0	

Рисунок 7 – Анализатор трафика KisMAC

Данная программа разработана исключительно под MAC OC, в связи с этим имеет очень ограниченную область применения и в нашем случае мы не сможем ее использовать на серверах с ОС Linux и Windows.

1.4 Вывод по разделу

В результате анализа задания на дипломное проектирование были сформулированы четкие требования к разрабатываемой, системе. Было решено, что разрабатываемая система, будет представлять из себя программу состоящею из двух частей: первая часть производит захват данных устройств одного сетевого маршрута и сохраняет их в формате pcap. Вторая часть производит анализ полученных данных с целью выявления сетевых уязвимостей сети через которые может произойти несанкционированный доступ в рассматриваемую сеть.

Было рассмотрено несколько аналогов программ по работе с данным сети и выявлено что не одна программа не обладает функционалом анализа трафика на всех участках сети.

Так же была спроектирована структурная схема разрабатываемой программы, в которой продемонстрирован принцип работы.

2 Структура программной среды подготовки данных трафика

2.1 Способы получение файлов данных сети

На рисунке 8 представлена структурная схема локальной сети, в которой предустановлена система по сбору данных трафика. С помощью программ прослушивания, рассмотренных ранее по типу tcpdump и wireshark мы можем получать трафик, приходящий на пользовательский компьютер.

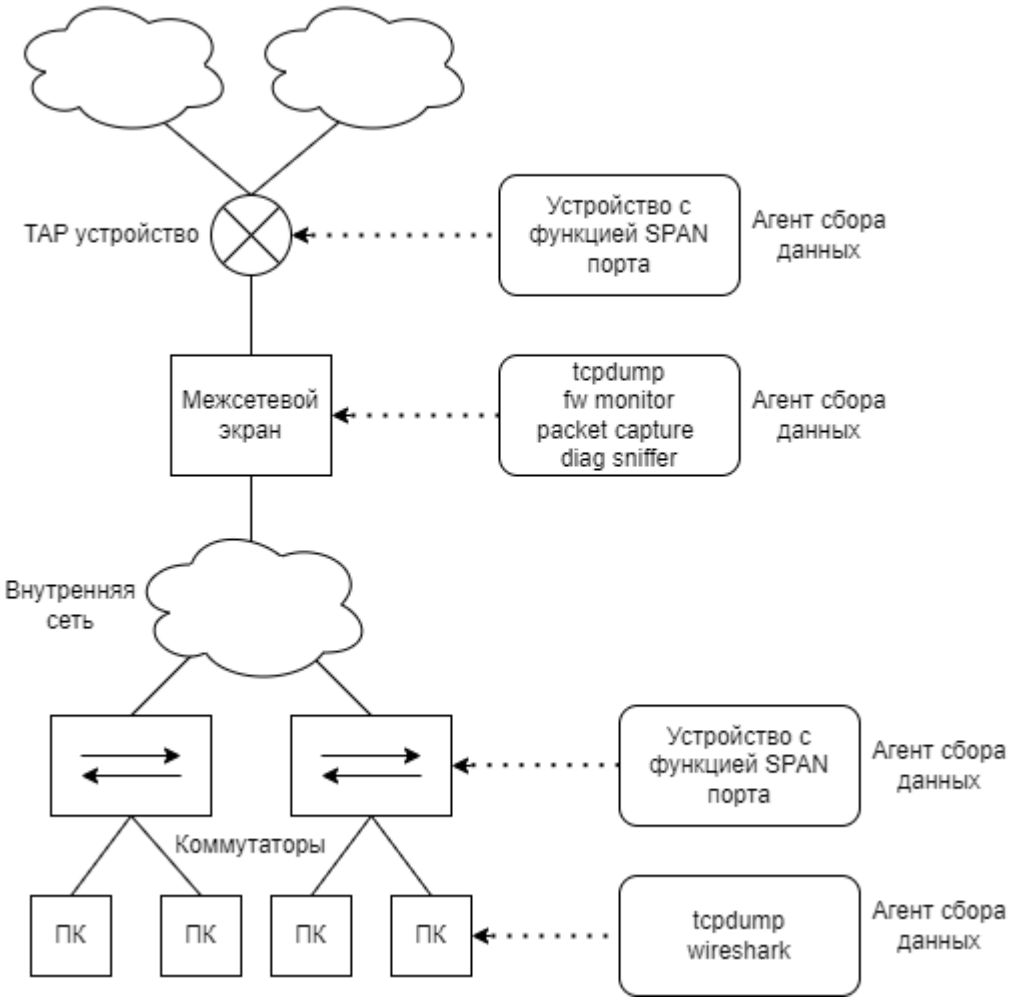


Рисунок 8 – Схема локальной сети

Межсетевой экран — это программный элемент компьютерной сети или программно-аппаратный, он служит для контроля и фильтрации проходящей через

него сетевой информации в соответствии с заданными правилами и протоклами. Основные задачи, которые решает межсетевой экран, является предостережение элементов сети или отдельных точек сети от несанкционированного доступа с использованием уязвимостей в протоколах сетевой модели OSI или в программном обеспечении нашей локальной сети и оборудование в ней. Межсетевые экраны пропускают или запрещают трафик, сравнивая его характеристики с заданными шаблонами. Межсетевые экраны обладают предустановленными утилитами по фильтрации данных, такие утилиты зависят от сетевого оборудования.

Например:

- Check point — fw monitor;
- Cisco — packet capture;
- Fortigate — diag sniffe;
- Nix — tcpdump.

За получение файлов данных отвечают агенты сбора данных. Агенты представляют из себя устройства с предустановленным ПО имеющим способность захватывать трафик в определенных участках сети и отсылать данные на основное устройство где будет происходить дальнейший анализ трафика.

Есть несколько таких способов получения данных, один из таких способов возможно осуществить если в сети которую мы хотим анализировать предустановлен HUB или же сетевой концентратор. Благодаря тому, что HUB работает на физическом уровне сетевой модели OSI, ретранслируя входящий сигнал с одного из портов на все остальные подключенные порты мы очень просто можем перехватывать все данные и сигналы, проходящие через HUB.

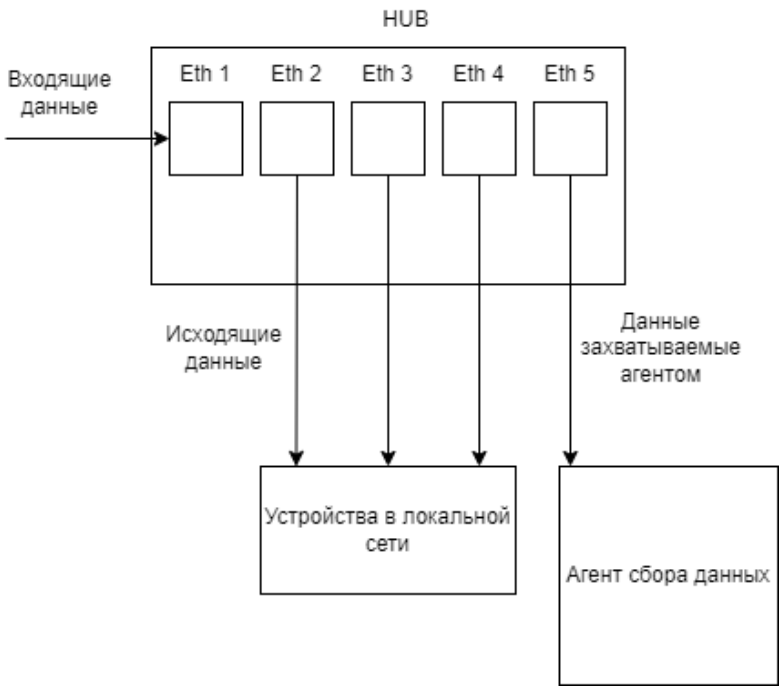


Рисунок 9 – Пример работы HUB в локальной сети

Из-за того, что технология HUB не актуальна и устарела в данный момент мы не всегда можем использовать данную возможность и захватывать трафик просто прослушивая наше сетевое устройство, поэтому нужно иметь еще один вариант захвата данных сети.

Зеркалирование трафика будет нашим еще одним вариантом. Данная технология представляет из себя способность копирования пакетов одного порта сетевого коммутатора и отдельных VLAN на другой порт. На данный момент большое количество настраиваемых сетевых коммутаторов позволяют отзеркалить трафик от одного или даже нескольких портов, или VLAN на отдельный порт выделенный для данной необходимости. Метод зеркалирования используется на сетевых устройствах по типу коммутатор или маршрутизатор для отправки копий сетевых пакетов с данными, который может видеть пользователь на исходных портах, отправка происходит на другие указанные порты назначения. При включенном зеркалирование портов пакеты мы имеем возможность отслеживать и анализировать.

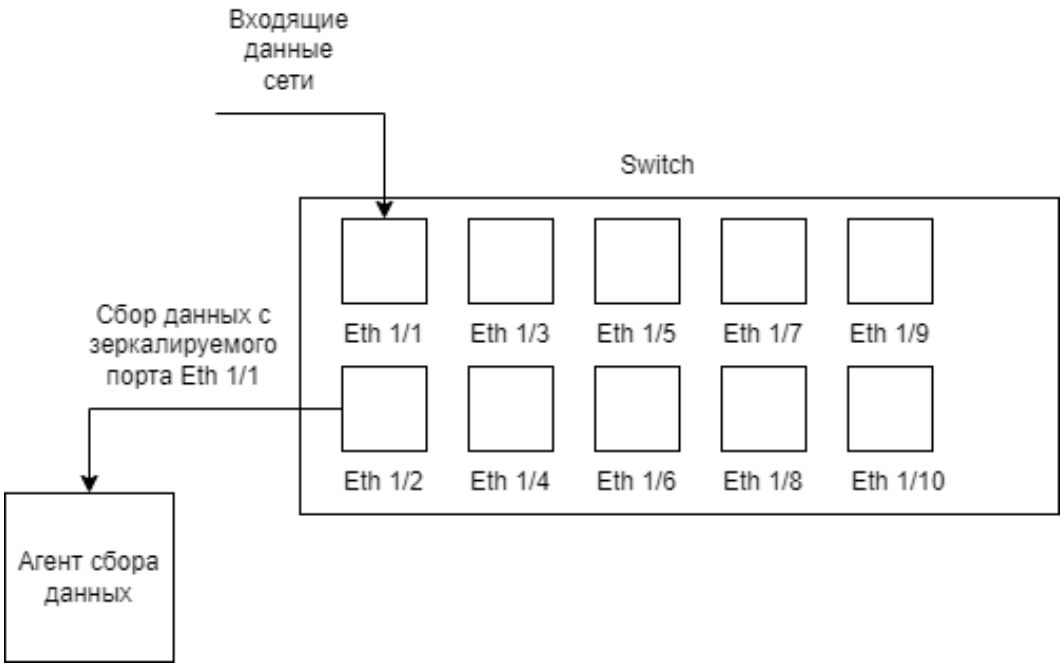


Рисунок 10 – Принцип работы локального зеркалирования портов

Есть два типа зеркалирования: локальное и удаленное, отличаются они тем, что принцип работы, этих типов основан на различных диапазонах зеркалирования. Они работают по разным принципам.

Локальное зеркалирование портов это простая форма зеркалирования. Все первоначальные порты находятся на том сетевом устройстве на котором и порт назначения. Копирование трафика локального порта дает возможность сетевому устройству переслать копию пакета данных с порта откуда были отправлена информация на порт назначения. Далее устройство отслеживания, установленное на порт назначения, может получать и анализировать пакет.

О зеркалирования удаленных портов, соновное отличие в том, что порт с исходящими данными и порт назначения находятся на разных устройствах. Порт с исходящими данными находится на первом коммутаторе, а порт назначения - на втором коммутаторе. Исходный порт отправляет копию отзеркаленного пакета на порт назначения с помощью соединения восходящей линии связи, достигнутое

портами на двух данных коммутаторах. В последствие, копировании данных с локальных портов позволяет осуществлять мониторинг и анализ данных на разных устройствах.

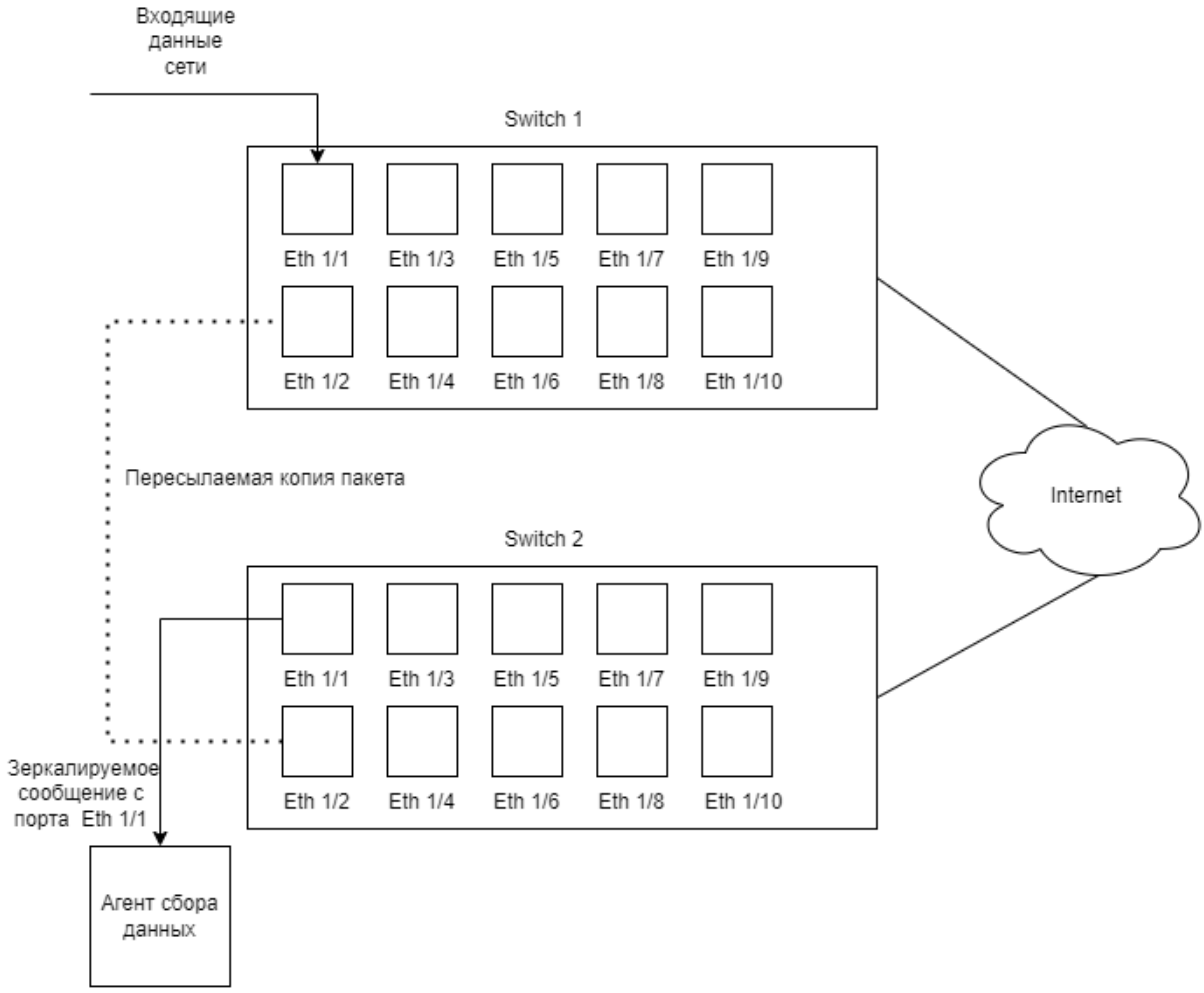


Рисунок 11 – Принцип работы удаленного зеркалирования портов

Наиболее универсальным и подходящим нам вариантом является функция зеркалирование трафика. Данная способность коммутатора, создана для перенаправления потока данных сети с одного порта сетевого устройства на другой порт этого же сетевого устройства или на удаленный порт коммутатора либо любого другого устройства с подобным функционалом. Исходный порт копирует поток данных, соответствующий по заданными правилам от клиента к порту назначения,

который в последствие отправляет скопированный поток данных на устройство мониторинга. Данный поток данных трафика может быть настроен с помощью ACL (Access Control List) или командами конфигурации. При зеркалировании трафика на устройство мониторинга данных сети отправляются только выбранные или согласованный трафик, а при зеркалировании портов копируется каждый пакет, который проходит через интерфейс, на устройство мониторинга.

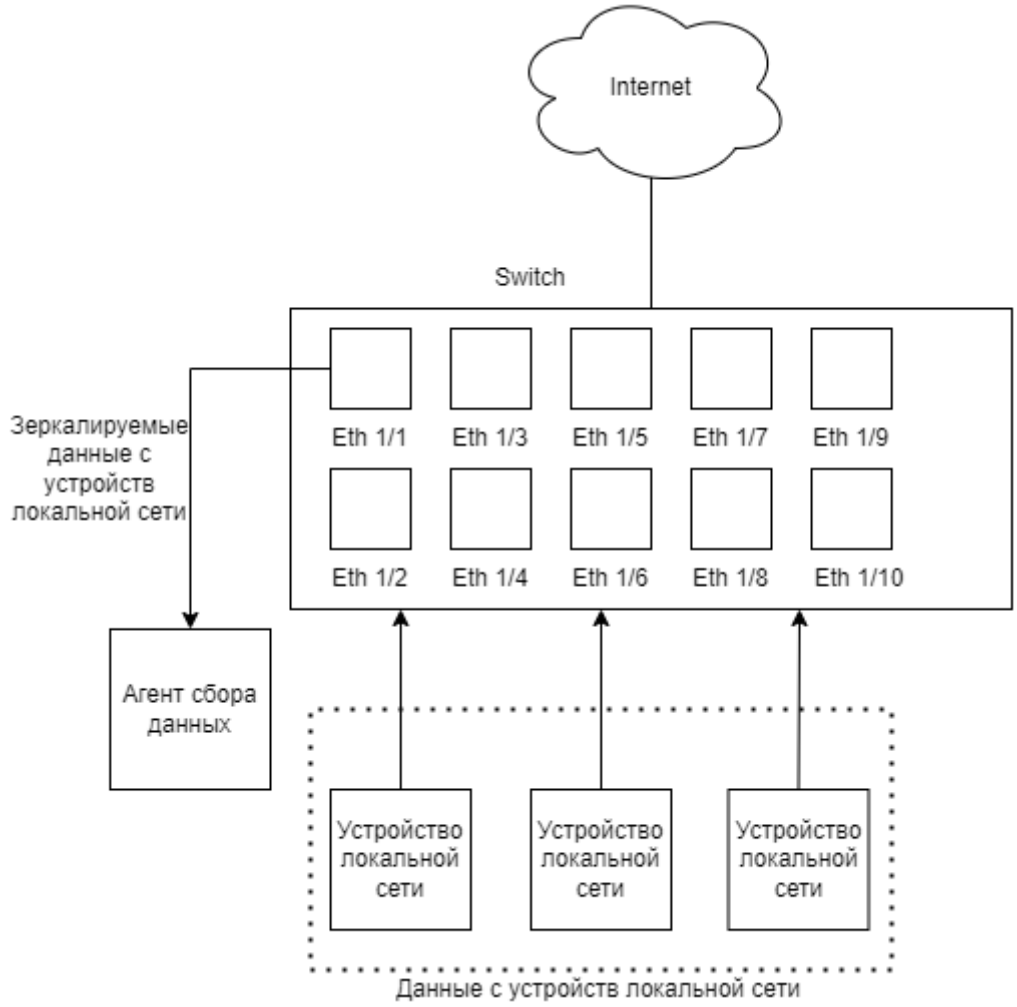


Рисунок 12 – Принцип работы зеркалирования трафика

2.2 Функционал агента по сбору данных

Агент по сбору данных это дополнительное устройство в сети функционал которого рассчитан на получение данных сети для дальнейшего анализа сети. Полученные данных происходит с помощью снифферров(sniffers).

Снифферы — это программное обеспечение, которое способно прослушивать, перехватывать и анализировать сетевой трафик. Снифферы используются в случае, когда необходимо получить из потока данных трафика сети какие-либо сведения или провести диагностику сети этой самой сети. Программу нужно установить на устройствах, к которому имеет доступ пользователь, и в течение некоторого промежутка времени получить все возможные к получению передаваемые данные.

Так как получение данных происходит на разных участках сети, все снифферы должны быть синхронизированы между собой для точного отслеживания данных сети дальнейшего анализа трафика. Данные которые получает сниффер преобразуется в пакеты данных для дальнейшей транспортировки.

Агенты по сбору должны одновременно включатся и начинать получение трафика определенное количество времени. Включение происходит по сигналу пользователя. После того как снифферы выполняют свою задачу по сбору трафика, данные передаются на основное устройство где уже и будет происходит анализ полученной информации из сети. Процесс включение я отключения снифферof контролирует пользователь удаленно с основного устройства.

2.3 Задачи основного ПО

На рисунке 13 представлена схема локальной сети, в которую внедрена система по сбору информации и передача полученных данных в основную программу.

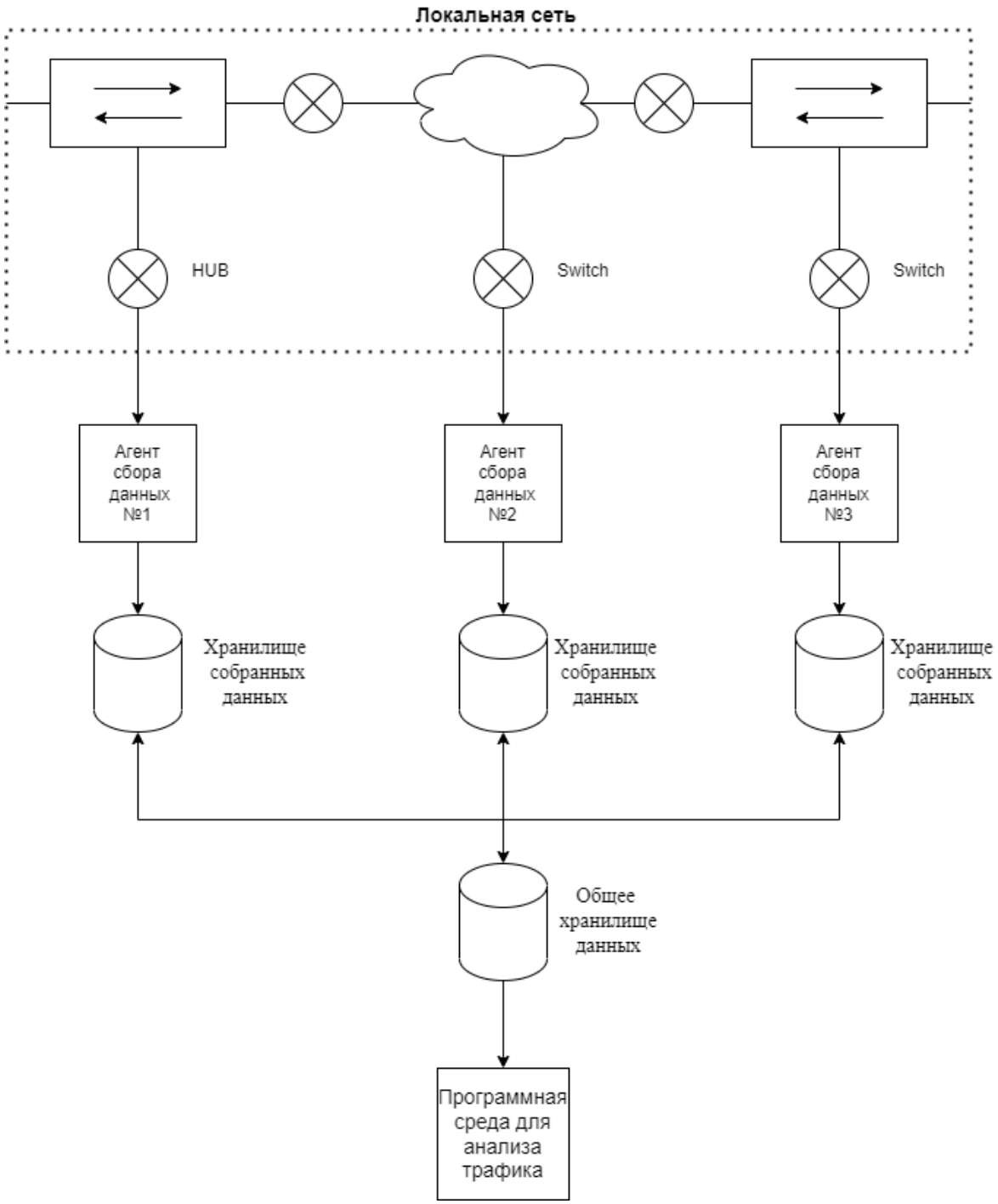


Рисунок 13 – Структурная схема разрабатываемой программы

Первой задачей основной программы будет работа с данными трафика полученными с помощью агентов по сбору данных сети. Подготовка полученных данных к дальнейшему анализу путем определения содержимого пакетов данных.

Далее нужно сопоставление данных между собой полученных на разных участках сети. Это необходимо для точного определения и отслеживания маршрута данных и дальнейшего анализа, происходящего в нашей локальной сети.

После загрузки данных в нашу программу подготовки пакетов нужно перейти к анализу полученной информации. Данные могут анализироваться пользователем путем сравнения параметров сети которые будут демонстрироваться на экране.

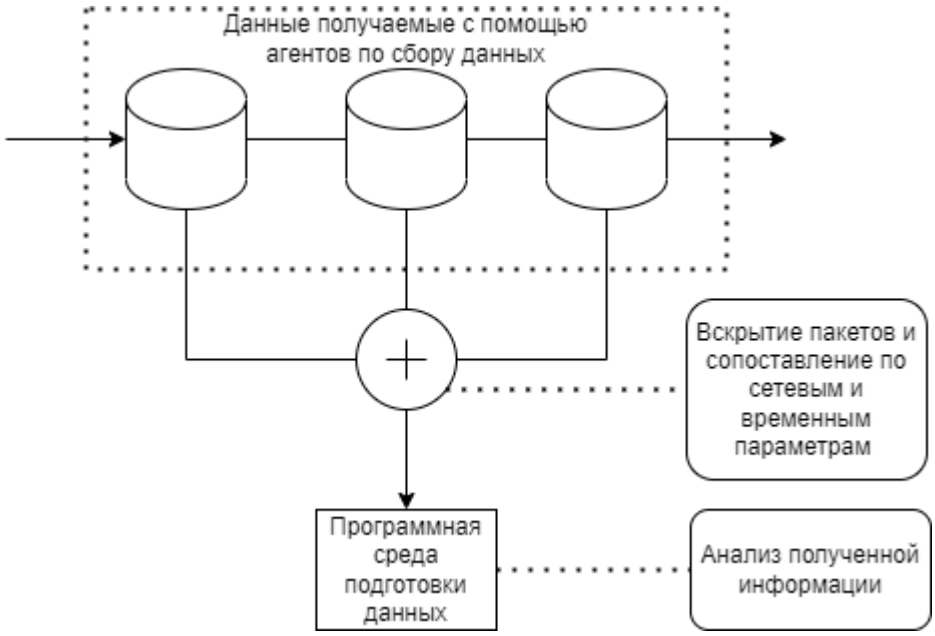


Рисунок 14 – Функция загрузки данных в разрабатываемую программу

Так же необходимо рассмотреть варианты если в сети может присутствовать NAT. NAT — это механизм преобразования IP-адреса транзитных пакетов в сетях TCP/IP.

2.4 Вывод по второму разделу

Во втором разделе пояснительной записки был описан принцип работы основного приложения по анализу трафика, а также показан способ получения данных сети с помощью нашего программного обеспечения и утилит, встроенных в сетевое оборудование.

Также были выделены необходимые функции, которое должно реализовать приложение:

- открытия пакетов данных сети и разбитие на параметры для анализа;
- синхронизация пакетов данных по времени отправления и прибытия;
- синхронизация пакетов данных по параметрам IP и MAC адресов;
- возможность просмотра данных с помощью пользовательского интерфейса;
- необходимо рассмотреть возможность сопоставления данных в сети с NAT.

3 Программная реализация

3.1 Средства разработки

3.1.1 Язык программирования C#

Программа будет разрабатываться на языке программирования C#.

C#— объектно-ориентированный язык программирования или же ООП. Данный язык был разработан в 1998—2001 годах группой инженеров компании из Microsoft. Основной задачей данного языка является разработка приложений для платформы Microsoft .NET Framework и .NET Core.

C# является языком с C-подобным синтаксисом, его синтаксис очень близок к C++ или Java. Язык обладает статической типизацей, поддерживает такие возможности как: полиморфизм, перегрузку операторов, делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. Данный язык получил многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в большой особенности, Java — C#, опирается на практику их использования, исключая несколько моделей, зарекомендовавших себя как имеющие проблематичные ситуации при разработке программных систем, например, C# в отличие от C++ не имеет возможность поддерживать множественное наследование классов.

3.1.2 Библиотека обработки сетевых пакетов

SharpPcap это библиотека для захвата, открытия и обработки сетевых пакетов. На данный момент SharpPcap активно развивается так как является библиотекой с открытым исходным кодом. Код библиотеки размещенным на площадке SourceForge. SharpPcap является полностью управляемой кроссплатформенной библиотекой. Библиотека работает на той-же сборке на которой работает под Microsoft .NET также как Моно на 32 или 64-битных платформах. Данный список демонстрирует возможности, которые в настоящее время поддерживаются в SharpPcap:

1. Одна сборка для Microsoft .NET и Моно платформ на Windows (32 или 64-разрядные), Linux (32 или 64 бит) и Mac.
2. Высокая производительность — SharpPcap позволяет захватывать данные до >3MB/s скорости передачи
3. WinPcap частично поддерживает:
   - Удаленный захват пакетов
   - Настройка размер буфера ядра
   - Инъекции пакетов, используя отправку очередей.
4. Сбор сетевой статистики по определенному сетевому интерфейсу
5. Поддержка AirPcap
6. Перечисление и отображение подробных сведений о физических сетевых интерфейсов на Windows-машине.
7. Захват низкоуровневых сетевых пакетов, проходящих через определенный интерфейс.
8. Использование Packet.Net для разбора пакетов

9. Чтение и запись в pcap файлы

Packet.Net поддерживает для анализа и разбора следующие протоколы:

- SLL (Linux Cooked-Mode Capture)
- Ethernet
- ARP (Address Resolution Protocol)
- IP (Internet Protocol):
  - IPv4
  - IPv6
- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)
- ICMP (Internet Control Message Protocol):
  - ICMPv4
  - ICMPv6
- IGMPv2
- PPPoE
- PTP
- LLDP
- Wake-on-LAN(WOL)
- SharpPcap имеет многоуровневую архитектуру, на верхнем уровне классы, которые работают с всеми устройствами:
- CaptureDeviceList — Возвращает список всех устройств в системе

- ICaptureDevice — Все устройства захвата имеют интерфейс ICaptureDevice

Иерархия пространства имен:

- LibPcap

- LibPcapLiveDevice — ICaptureDevice

- LibPcapLiveDeviceList – запрашивает список устройств (он включает pcap/winpcap и airpcap устройства)

- CaptureFileReaderDevice – Устройство, которое считывает из pcap файла

- CaptureFileWriterDevice – Устройство, которое создает и записывает в pcap файл

- WinPcap

- WinPcapDeviceList – Запрашивает список WinPcapDevices (он включает winpcap и airpcap устройства)

- WinPcapDevice — LibPcapLiveDevice с дополнительными WinPcap функциями и интерфейсами

- AirPcap

- AirPcapDeviceList – Запрашивает список AirPcapDevices

- AirPcapDevice — WinPcapDevice с дополнительными AirPcap функциями и интерфейсами

CaptureDeviceList возвращает список всех устройств. Каждый из ICaptureDevice будет либо LibPcapLiveDevice, WinPcapDevice или AirPcapDevice. Это позволяет получить всех устройств и дифференцировать их по типам. Если вы

хотите получить конкретный тип устройства, можно использовать один из особых *DeviceList классов.

3.1.3 Платформа создания интерфейса на C#

Windows Presentation Foundation — это платформа пользовательского интерфейса для создания клиентских приложений для настольных систем. Платформа разработки WPF поддерживает широкий набор компонентов для разработки приложений, включая модель приложения, ресурсы, элементы управления, графику, макет, привязки данных, документы и безопасность.

WPF является частью платформы .NET. WPF использует расширяемый язык разметки для приложений (XAML), чтобы предоставить декларативную модель для программирования приложений.

3.2 Функции приложения

Основной функцией приложения, является вскрытие файлов формата pcap. Библиотека SharpPcap имеет функции, позволяющие производить разбор по параметрам, которые хранятся в пакетах данных. Пакеты с данными будут передаваться в основную программу с агентов на удаленных устройствах.

Рисунок 15 – Uml диаграмма

3.2.1 Функционал ПО для захвата трафика

По для захвата трафика заранее установлено и запущено на устройствах в сети. Агенты синхронизируются между собой и основным ПО и ожидают сигнала на начало запуска прослушивания трафика. Пользователю необходимо убедиться перед началом работы, что связь со всеми агентами установлена, путем отправки сигнала и получение ответа. Далее пользователь устанавливает временные рамки для начала и окончания работы агента. Агент начинает захват пакетов с данными в

течение установленного времени. После окончания захвата, агенты отсылают пакеты с данными в основное хранилище. Файлы из основного хранилища попадают основную программу где и происходит дальнейшее взаимодействие пользователя с информацией.

3.2.2 Функция OpenFileDialog

Пакеты загружаются в программу с помощью с помощью функции WPF OpenFileDialog. Данная функция позволяет открывать проводник Windows в заранее прописанном каталоге и отбирать данные по указанным критериям. Выбирая нужный файл в проводнике, мы получаем загружаем в программу такие параметры как место нахождение файла в системе и имя файла.

Фрагмент кода программы с данной функцией:

```
1. var filePath = string.Empty;
2. string capFile;
3. using (OpenFileDialog openFileDialog = new OpenFileDialog())
4. {
5. openFileDialog.InitialDirectory = "c:\\";
6. openFileDialog.Filter = "pcapng files (*.pcapng)|*.pcapng|All files (*.*)|*.*";
7. openFileDialog.FilterIndex = 2;
8. openFileDialog.RestoreDirectory = true;
9. if (openFileDialog.ShowDialog() == DialogResult.OK)
10. {
11. //Get the path of specified file
12. filePath = openFileDialog.FileName;
13. //Read the contents of the file into a stream
14. }
```

Рисунок 16 – Функция загрузки файла в программу

3.2.2 Функция OpenRead

После получения информации по местонахождению файла в нашей системе, мы передаем эту информацию в функцию OpenRead, где происходит открытие этого файла с помощью функционала библиотеки SharpPcap.

Фрагмент кода программы с данной функцией:

```
1. public static void OpenRead(string capFile)
2. {
3. packetIndex = 0;
4. ICaptureDevice device;
5. device = new CaptureFileReaderDevice(capFile);
6. device.Open();
7. device.OnPacketArrival += device_OnPacketArrival;
```

8. device.StartCapture();
9. System.Threading.Thread.Sleep(50);
10. }

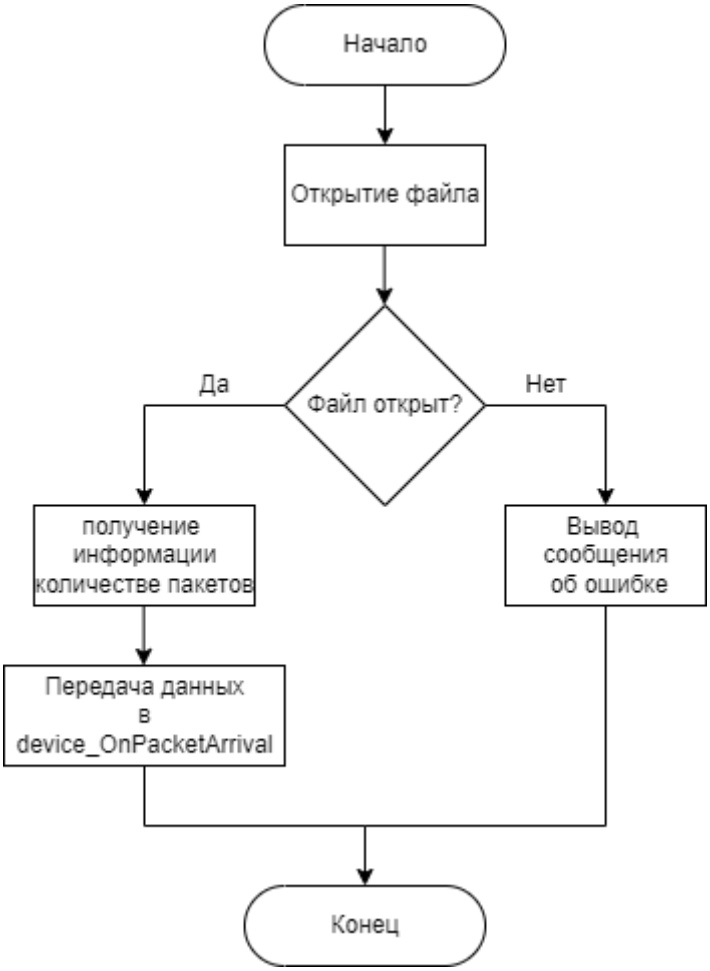


Рисунок 17 – Функция открытия файла

3.2.3 Функция device_OnPacketArrival

Далее мы переходим в событие device_OnPacketArrival в котором происходит разбор открытого нами файла на параметры, которые нам нужно будет продемонстрировать пользователю. Получение параметров происходит с помощью функционала библиотеки SharpPcap.

Фрагмент кода программы с данным событием:

```
1.  private static void device_OnPacketArrival(object sender, PacketCapture e)
2.  {
3.  var rawPacket = e.GetPacket();
4.  var packet = PacketDotNet.Packet.ParsePacket(rawPacket.LinkLayerType,
rawPacket.Data);
5.  var ethernetPacket = packet.Extract<EthernetPacket>();
6.  index = packetIndex;
7.  if (ethernetPacket != null)
8.  {
9.  var srsadd = new Regex(@"SourceAd-
dress=\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b");
10.  var protocol = new Regex(@"Protocol=\b\d{1,5}\b");
11.  TimevalDate.Add(e.Header.Timeval.Date.ToString());
12.  Millisecond.Add(e.Header.Timeval.Date.Millisecond.ToString());
13.  DestinationMACAddress.Add(ethernetPacket.DestinationHardwareAd-
dress.ToString());
14.  TypePacket.Add(ethernetPacket.Type.ToString());
15.  PacketLength.Add(ethernetPacket.TotalPacketLength.ToString());
16.  var match2 = srsadd.Match(Convert.ToString(ethernetPacket));
17.  SourceMACAddress.Add(match2.ToString());
18.  var match3 = protocol.Match(Convert.ToString(ethernetPacket));
19.  EthernetPacket.Add(match3.ToString());
20.  }
21.  packetIndex++;
22.  }
```

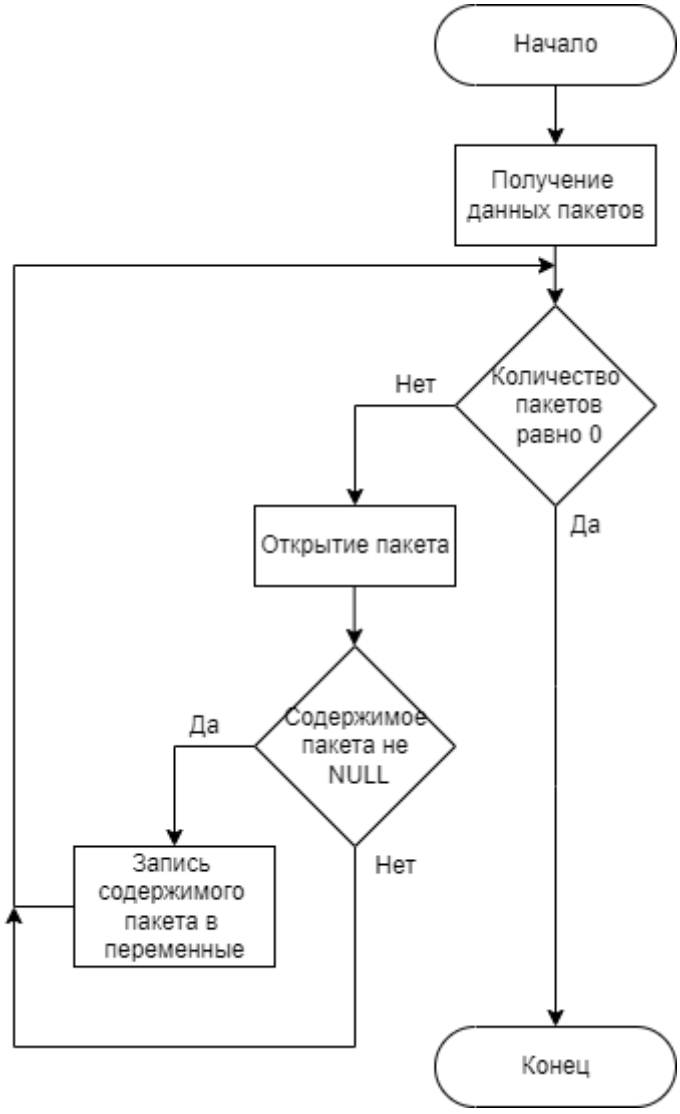


Рисунок 18 – Функция распаковки данных пакета

3.2.4 Функция treeView

После считывания необходимых параметров необходимо произвести демонстрацию полученных данных из пакета. Информацию будет выводиться в виде древовидной структуры в окнах treeView которые взяты из функционала WPF.

Фрагмент кода программы с данной функцией:

```
1.  treeView.Nodes.Clear();
2.  for (int i = 0; i < PcapFunck.index; i++)
3.  {
4.  treeView.BeginUpdate();
5.  treeView.Nodes.Add("Packet№" + i);
6.  treeView.Nodes[i].Nodes.Add("Timeval.Date:" + PcapFunck.Timeval-
Date[i]);
7.  treeView.Nodes[i].Nodes.Add("Millisecond:" + PcapFunck.Mil-
lisecond[i]);
8.  treeView.Nodes[i].Nodes.Add("SourceHardwareAddress MAC:" + Pcap-
Funck.SourceMACAddress[i]);
9.  treeView.Nodes[i].Nodes.Add("DestinationHardwareAddress  MAC:" +
PcapFunck.DestinationMACAddress[i]);
10.  treeView.Nodes[i].Nodes.Add("Type:" + PcapFunck.TypePacket[i]);
11.  treeView.Nodes[i].Nodes.Add("PacketLength:" + PcapFunck.Pack-
etLength[i]);
12.  treeView.Nodes[i].Nodes.Add(PcapFunck.EthernetPacket[i]);
13.  treeView.EndUpdate();
14.  }
```

Рисунок 19 – Функция вывода информации на экран

3.3 Вывод по третьему разделу

На основе сформулированных требований во втором разделе были определены технологии разработки. В результате программной реализации было создано приложение на языке программирования C# с использованием платформы для создания пользовательского интерфейса Windows Presentation Foundation и библиотеки для сбора данных сети SharpPcap.

Были реализованы функции по выводу информации содержащейся в пакетах данных.

4. Пример работы

4.1 Интерфейс приложения

Интерфейс приложения представляет из себя несколько полей. Первым полем является статическая карта устройств локальной сети, с которой может взаимодействовать пользователь. В этом поле находятся кнопки, с помощью которых пользователь может загружать файлы формата pcap полученные с устройств на которых был заранее произведен захват трафика сети. На данный момент в программу можно загрузить до четырех таких файлов.

Во втором поле происходит вывод файлов, загруженных в программу. Пользователь получает информацию из файлов в виде древовидной структуры. Пользователь имеет возможность получать более подробную информацию содержимого пакетов данных открывая их и считывая такие данные как: source ip, destination ip, размер пакета, временные характеристики.

Пользователь может устанавливать фильтры с помощью которых изменяется содержимое окон вывода информации по необходимым критериям пакетов.

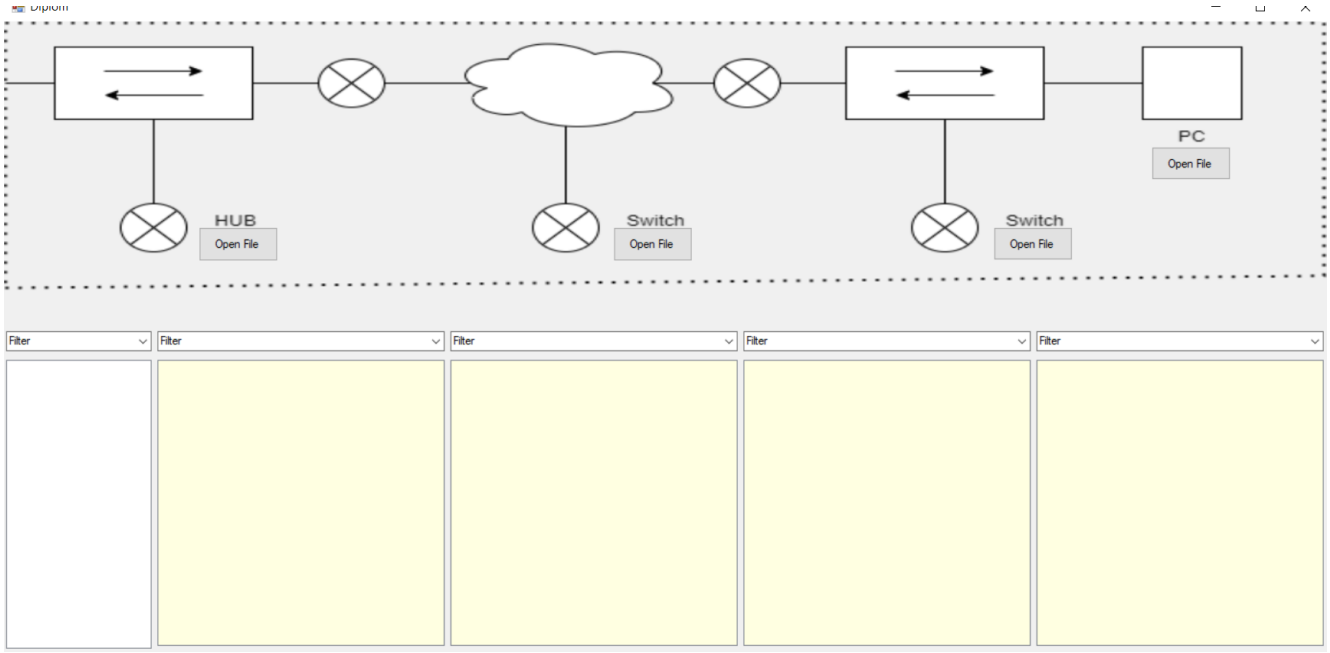


Рисунок 20 – Интерфейс приложения

Кнопки для загрузки файлов находятся на статической карте локальной сети рядом с устройствами, с которых должен браться заранее перехваченные данные трафика сети. Файлы загружаются в ту область на карте локальной сети из которой они были получены.

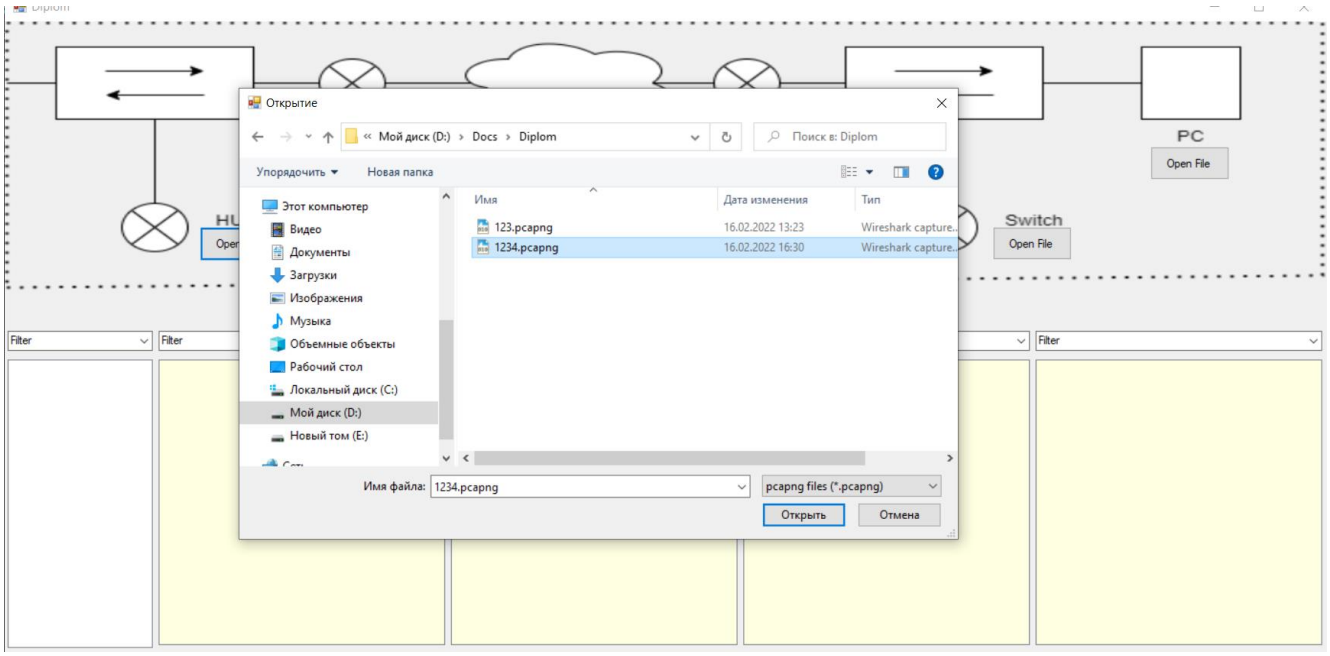


Рисунок 21 – Загрузка pcap файла в программу

Информация содержащиеся в пакетах выводится в 5 окнах различных окнах. Первое окно представляет из себя список временных характеристик пакетов, он является единым для следующих четырех окон, начало захвата данных и окончание производилось в одно и тоже время.

Окна с выводом информации о данных трафика сети заполняются по мери загрузки файлов. Каждое окно связано с участком на карте и демонстрация информации произойдет тогда, когда пользователь загрузит файл с помощью верхнего поля программы.

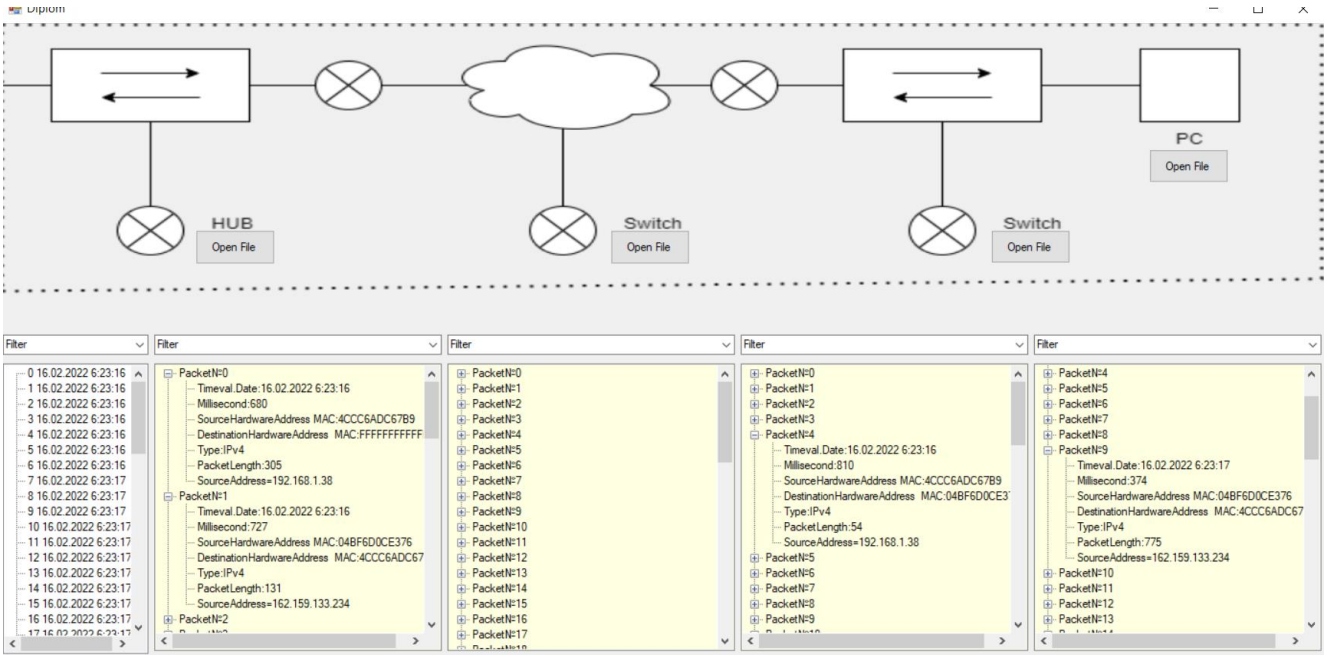


Рисунок 22 – Вывод информации из пакетов

Над окнами с выводом информации из пакетов данных расположены текстовые поля, предназначенные для изменения содержимого полей для вывода по определённым критериям, которые необходимы пользователю для дальнейшего анализа трафика. Если информация записанная в данное поле будет совпадать с информацией в пакетах данных, то содержимое окна от сортируется по наличию совпадений.

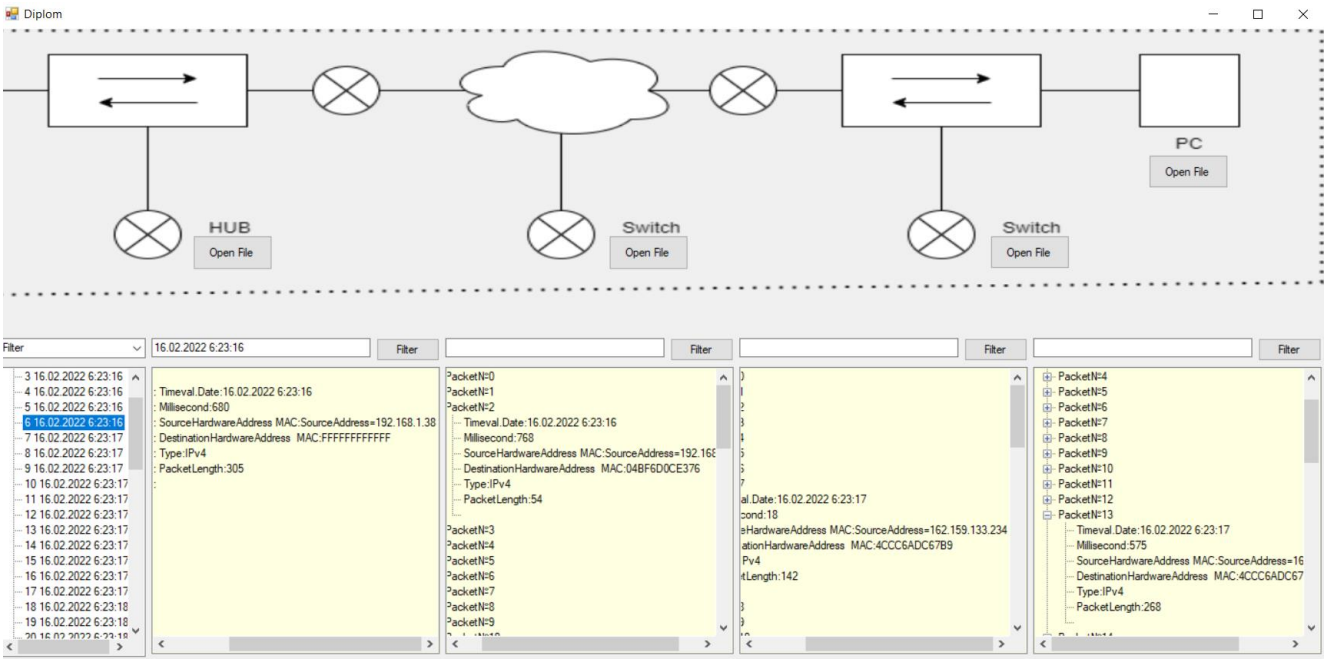


Рисунок 23 – Применение функции фильтрации данных

ЗАКЛЮЧЕНИЕ

В результате выполнения работы была изучена предметная область и существующие на данный момент аналоги.

После изучения аналогов был сформулирован ряд требований, предъявленных к приложению. Приложение было написано на языке C# с использованием платформы для создания пользовательского интерфейса Windows Presentation Foundation и библиотеки для сбора данных сети SharpPcap.

Была создана структура программной среди в которой описаны принципы работы основного приложения по подготовки трафика к анализу, а также показан способ получения данных сети с помощью нашего программного обеспечения и утилит, встроенных в сетевое оборудование.

В результате была разработана система получения данных с устройств локальной сети позволяющая проводить предварительную подготовку информации к обработке.

Было разработано приложение обработки большого количества полученной информации, в котором определяются и выделяются необходимые параметры данных сети подготавливающее данные с различных устройств сети к дальнейшему анализу и демонстрации, что позволяет пользователю выделять временные и неявных зависимости.

СПИСОК СОКРАЩЕНИЙ

ОС — операционная система

VoIP — Voice over IP (IP-телефония)

NAT — Network Address Translation (преобразование сетевых адресов)

WPF — Windows Presentation Foundation

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Wireshark [Электронный ресурс]: – Режим доступа: https://www.wireshark.org/
2. Tcpdump [Электронный ресурс]: – Режим доступа: https://www.tcpdump.org/manpages/tcpdump.1.html
3. Kismet [Электронный ресурс]: – Режим доступа: https://www.kismetwireless.net
4. Etherape [Электронный ресурс]: – Режим доступа: https://etherape.sourceforge.io/
5. Cain and Abel [Электронный ресурс]: – Режим доступа: https://en.wikipedia.org/wiki/Cain_and_Abel_(software)
6. NetworkMiner [Электронный ресурс]: – Режим доступа: https://www.netresec.com/?page=NetworkMiner
7. KisMAC [Электронный ресурс]: – Режим доступа: https://kismac-ng.org/
8. WPF [Электронный ресурс]: – Режим доступа: https://docs.microsoft.com/ru-ru/visualstudio/designers/getting-started-with-wpf?view=vs-2022
9. Sharppcap [Электронный ресурс]: – Режим доступа: https://github.com/dotpcap/sharppcap

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра «Вычислительная техника»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В. Непомнящий
«20» 06 2022 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Программная среда подготовки данных для анализа трафика
тема

Руководитель _____ доцент, канд.техн.наук Ф.А. Казаков
подпись, дата

Выпускник _____ С.С. Овсянников
подпись, дата

Нормоконтролер _____ доцент, канд.техн.наук Ф.А. Казаков
подпись, дата

Красноярск 2022