МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАБОТА ПРОВЕРЕНА

Рецензент
Заместитель директора по
информационным технологиям
ООО «ИТ Энигма»

_____ П.Л. Заостровных

«___»_____ 2024 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор

_____ Л.Б. Соколинский

«___»_____ 2024 г.

Применение методов машинного обучения для анализа структуры сетевого трафика

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.04.02.2024.308-1394.ВКР

Научный руководитель,
доцент кафедры СП, к.ф.-м.н.
_____ А.Т. Латипова

Автор работы,
студент группы КЭ-220
_____В.А. Лисовец

Ученый секретарь
(нормоконтролер)
_____ И.Д. Володченко
«___»_____ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

УТВЕРЖДАЮ
Зав. кафедрой СП
_____ Л.Б. Соколинский
29.01.2024 г.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы магистранта
студенту группы КЭ-220
Лисовец Валентину Аркадьевичу,
обучающемуся по направлению
02.04.02 «Фундаментальная информатика и информационные технологии»
(магистерская программа «Машинное обучение и анализ больших данных»)

1. **Тема работы** (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)
Применение методов машинного обучения для анализа структуры сетевого трафика.
2. **Срок сдачи студентом законченной работы:** 20.05.2024 г.
3. **Исходные данные к работе**

3.1. Йен Г., Йошуа Б. Глубокое обучение: адаптивное вычисление и машинное обучение. – MIT Press, 2016. – 393 с.

3.2. Александр К., Евгений Х. Машинное обучение для анализа сетевого трафика. – СПАРК, 2019. – 596 с.

3.3. Шайлендра С., Гопал К. С. Анализ трафика с использованием алгоритмов машинного обучения. – ICCCA, 2021. – 608 с.

4. **Перечень подлежащих разработке вопросов**

4.1. Сбор интернет-трафика для тестирования разработанных моделей.

4.2. Разработка, обучение, и тестирования машинного обучения.

4.3. Тестирование разработанных алгоритмов на собственном наборе данных.

5. Дата выдачи задания: 29.01.2024 г.

Научный руководитель,
доцент кафедры СП, к.ф.-м.н. А.Т. Латипова

Задание принял к исполнению В.А. Лисовец

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Актуальность

В современном мире, интернет играет важную роль в жизни людей и организаций. Он позволяет обмениваться информацией, вести бизнес, обучаться и развлекаться. Однако, с ростом числа пользователей и объема передаваемых данных, возникает необходимость в анализе и оптимизации интернет-трафика. В этом контексте, машинное обучение становится важным инструментом для анализа и прогнозирования трафика.

Нейронные сети являются мощным инструментом, который применяется в различных областях, включая компьютерное зрение, распознавание речи и обработку естественного языка. Они способны обучаться на больших объемах данных и выявлять сложные закономерности, которые могут быть не очевидными для человека. Машинное обучение также позволяет создавать алгоритмы, способные к самообучению и адаптации к новым данным, что делает их особенно полезными для анализа сетевого трафика.

Анализ интернет-трафика представляет собой сложную задачу, так как данные защищены с помощью шифрования и не могут быть прочитаны напрямую. Однако, нейронные сети и машинное обучение позволяют обрабатывать данные на уровне статистических закономерностей и паттернов, которые могут дать представление о характере трафика и его особенностях. Это может быть использовано для обнаружения аномалий, идентификации вредоносного ПО, выявления нарушений политик безопасности и многого другого.

Постановка задачи

Целью выпускной квалификационной работы является применение методов машинного обучения для анализа структуры сетевого трафика. Для достижения поставленной цели необходимо решить следующие задачи:

1) сбор интернет-трафика для тестирования разработанных моделей;
2) разработка, обучение, и тестирования машинного обучения;

3) тестирование разработанных алгоритмов на собственном наборе данных.

Структура и содержание работы

Работа состоит из введения, четырех глав, заключения и списка литературы. Объем работы составляет 43 страниц, объем списка литературы – 20 источников.

В первой главе приведен обзор научной литературы.

Во второй главе описываются основные теоретические сведения, связанные с машинным обучением, DDoS-атаками.

В третьей главе описан метод DDoS-атак на ОС Kali Linux с помощью инструмента Slowloris.

Четвертая глава посвящена сбору интернет-трафика, описаны функциональные и нефункциональные требования.

Пятая глава посвящена разработке алгоритмов машинного обучения Random Forest, K-NN, Logistic Regression.

В заключении сделаны выводы о проделанной работе и полученных результатах.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Обзор литературы

В работах представлены исследования, направленные на определение эффективных методов использования нейронных сетей в работе с зашифрованными данными, а также на разработку систем обнаружения аномалий.

Так, в работе «Deep Learning for Encrypted Traffic Classification Using Inter-Packet Information» [1] авторы предлагают использовать глубокую нейросеть для определения закономерностей в шифрованных данных.

В работе «Machine Learning-Based Anomaly Detection for Encrypted Network Traffic» [2] авторы представляют систему обнаружения аномалий, основанную на машинном обучении. Кроме того, в литературе представлен обзор существующих подходов к анализу шифрованного трафика на базе машинного обучения («A Systematic Literature Review on Machine Learning Acroaches for Malware Traffic Detection» [6]) и анализ эффективности различных архитектур нейросетей («A Survey on Deep Learning Techniques for Encrypted Traffic Analysis») [4].

Также в литературе представлены исследования, посвященные разработке систем обнаружения аномалий при помощи глубокого обучения («Anomaly Detection in Encrypted Networks Using Deep Autoencoders») [3]. Эти системы способны обнаруживать различные виды вредоносной активности в интернет-трафике.

В статье «Исследование методов и средств обнаружения DDoS-атак» [5] автор анализирует различные подходы к обнаружению DDoS-атак. Он выделяет пассивные и активные методы обнаружения.

Пассивные методы основаны на мониторинге сетевого трафика без вмешательства в работу сети. Они включают анализ логов серверов и сетевых устройств, а также использование систем обнаружения аномалий. Эти методы не влияют на работу сети и серверов, но могут быть менее эффективными при обнаружении новых или измененных методов атаки.

Активные методы обнаружения DDoS-атак предполагают активное вмешательство в работу сети и трафика. Они включают методы пороговой фильтрации и изоляции трафика от атаки [10].

Метод пороговой фильтрации основан на установлении порогового значения для нормального трафика. Если объем трафика превышает установленный порог, система активирует меры защиты.

Метод обнаружения и изоляции трафика от атаки предполагает изоляцию потенциально вредоносного трафика от основной сети. Это позволяет защитить сеть от DDoS-атак [11].

Автор также отмечает, что для эффективной защиты от DDoS-атак необходимо использовать сочетание пассивных и активных методов обнаружения.

В статье «A taxonomy of DDoS attack and DDoS defense mechanisms» [8] представлена таксономия атак и механизмов защиты от распределенных атак типа «отказ в обслуживании» (DDoS). Авторы разделяют атаки на основе их характеристик и предлагают классификацию защитных механизмов.

Таксономия включает следующие категории:

1) атаки на основе протоколов;
2) атаки на основе приложений;
3) атаки на основе сервисов;
4) атаки на основе инфраструктуры;
5) атаки на основе пользователей.

Для каждой категории атак предлагаются соответствующие защитные механизмы, такие как фильтрация подсетей, атака «Smurf», атаки с использованием флуда ICMP, SYN-флуд, атаки с использованием флуда UDP, флуд TCP-сегментами, флуд пакетами с ошибками, флуд пакетами с неправильными заголовками, флуд пакетами с неправильными адресами, флуд пакетами с неправильными портами, флуд пакетами с неправильным прото-

колом, флуд пакетами с неправильным типом службы, флуд пакетами с неправильным кодом приложения, флуд пакетами с неправильным номером версии приложения, флуд пакетами с неправильным номером сессии, флуд пакетами с неправильным номером службы, флуд пакетами с неправильным номером протокола, флуд пакетами с неправильным адресом источника, флуд пакетами с неправильным адресом назначения, флуд пакетами с неправильным временем жизни, флуд пакетами с неправильным идентификатором соединения, флуд пакетами с неправильным номером последовательности, флуд пакетами с неправильным номером подтверждения.

Статья «Hadoop based defense solution to handle distributed denial of service DDoS attacks» [9] и соавторов исследует характеристики DDoS-атак, включая их модели и механизмы, а также предлагает решение для обнаружения DDoS-атак с использованием модели программирования MapReduce на базе Hadoop.

В статье рассматриваются различные модели DDoS-атак, такие как уязвимость, отказ в обслуживании, перегрузка ресурсов и другие. Авторы также анализируют механизмы защиты от DDoS-атак, включая брандмауэры, системы обнаружения вторжений, балансировщики нагрузки и другие.

Основное внимание уделяется использованию Hadoop для обнаружения DDoS-атак. Hadoop – это распределенная файловая система и вычислительная платформа, которая использует модель MapReduce для параллельной обработки данных. Авторы предлагают модифицировать существующую модель MapReduce для обнаружения DDoS-атак [7].

В статье представлены результаты экспериментов, которые показывают эффективность предложенного решения. Эксперименты проводились на реальных данных, собранных из различных источников, и показали снижение времени обнаружения DDoS-атак на 30–40% по сравнению с традиционными методами.

В статье [16] описывают принцип работы ботнетов.

Принято различать три основные цели использования сети зараженных устройств.

1. Фишинговые рассылки. Хакеры используют ботнеты для организации масштабных рассылок спам-писем с вредоносными вложениями, открывая которые пользователь рискует заразить устройство разного рода вирусами, шпионским и рекламным программным обеспечением.

2. Мошенничество с кликами. Некоторые предприниматели используют маркетинговую стратегию, которая заключается в оплате за переходы по рекламным объявлениям. Таким образом предполагается заинтересовать пользователя и улучшить поведенческие факторы для продвижения в социальных сетях. Ботнеты применяются мошенниками для имитации большого количества переходов. Злоумышленники получают от этого финансовую прибыль, но не приносят выгоды бизнесмену, организующему стратегию.

3. Генерация вредоносного трафика. Для совершения атаки типа «отказ в обслуживании» (она же DDoS) хакеру необходимо сгенерировать большое количество трафика и направить его на нужную сеть. С этой целью используются зараженные компьютеры. Пользователи устройств зачастую не подозревают, что являются частью большой хакерской атаки на организацию или частное лицо.

В статье [17] описывается работа кроссплатформенного ПО Apache.

Apache – это посредник между серверным компьютером и браузером пользователя. Получив от клиента запрос, он находит нужную страницу в каталоге сайта и отправляет ее в ответ. Браузер анализирует присланный файл и преобразует его в веб-страницу, которую и видит пользователь. Схема работы выглядит так.

1. Сервер работает на порту 80, 8080 или 8000, но иногда бывают и другие порты, которые открыты для сторонних программ клиента.

2. Когда на один из портов поступает запрос, программа сопоставляет его с внутренними правилами и решает, исполнять или нет.

3. Если в правилах на запрос есть запрет, пользователь отказ в доступе к данным.

4. После того как серверная программа исполнила запрос, она переходит в режим ожидания.

В статье [18] описывается сферы применение ОС Kali Linux, утилиты, которые сразу имеются при установке ОС. Основное отличие Kali Linux от остальных ОС заключается в наличии широкого спектра приложений для проверки взлома и проникновения. Их общее количество превышает 600, и часть из них представлена ниже:

1) nmap, goofile, p0f – сбор информации;
2) cisco-torch, BED, DotDotPwn – анализ уязвимостей;
3) Aircrack-ng, Kismet, Pyrit – атаки на беспроводное соединение;
4) Blind Elephant, hURL, Nikto, Paros – веб-приложения;
5) DHCPig, t50, Reaver – стресс-тесты;
6) ddrescue, pdf-parser, iPhone-Backup-Analyzer – аналитика;
7) Hashcat, THC-Hydra, John the Ricer – взлом паролей;
8) _hexinject, mitmproxy, Wireshark – просмотр интернет трафика;

В статье [19] описывается алгоритм машинного обучения KNN.

Чтобы применять метод ближайших соседей, нужно найти соседей. Можно просто перебрать все объекты из обучающей выборки, посчитать для каждого из них расстояние до тестового объекта и затем найти минимум. Однако несмотря на то, что сложность такого поиска линейная, она также зависит и от размерности пространства признаков. Проблема в том, что данный поиск необходимо выполнять на этапе применения модели, который должен быть быстрым. Это означает, что возникает необходимость в более быстрых методах поиска ближайших соседей, чем простой перебор.

Все такие методы можно поделить на две основные группы: точные и приближенные. Приближенные, как следует из их названия, находят соседей лишь приблизительно, то есть найденные объекты хоть и будут действительно близки, но не обязательно будут самыми близкими [14].

В статье [20] описываются интернет-протоколы и компьютерные сети.

Компьютерная сеть – это набор компьютеров, совместно использующих ресурсы, расположены на сетевых узлах. Они используют общие протоколы связи по цифровым соединениям для связи с друг другом. Узлы могут включать в себя серверы, сетевое оборудование или другие универсальные хост-компьютеры. Компьютерные сети можно классифицировать по многим критериям протоколы связи, размер сети, топологию.

Большинство действующих стандартов интернета и протоколов TCP/IP регламентируются документами Request For Comments (RFC).

Уровни модели TCP/IP:

1) канальный;
2) межсетевой;
3) траснпортный;
4) прикладной;

IP объединяет сегменты сети в единую сеть, связанных между собой хостов. Хосты связаны непосредственно или косвенно при помощи ретранслирующих устройств (маршрутизаторов и коммутаторов). IP не гарантирует надежной доставки пакета до адресата – в частности, пакеты могут прийти не в том порядке, в котором были отправлены, продублироваться (приходят две копии одного пакета), оказаться поврежденными (обычно поврежденные пакеты уничтожаются) или не прийти вовсе.

Выводы по первой главе

В первой главе квалификационной работы был проведен обзор научной литературы, в ходе которого были рассмотрены основные аспекты для обнаружения DDoS-атак, применение ОС Kali Linux и инструменты, которые имеются в ОС.

Проведен анализ структуры компьютерной сети, уровни модели TCP/IP, как работает IP.

Какие методы машинного обучения предлагаются для анализа интернет-трафика.

2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1. Интернет-трафик

Интернет-трафик состоит из всех данных, которые передаются и принимаются через интернет. Этот термин включает в себя все данные, которые проходят через ваше устройство, такое как ваш компьютер, смартфон или планшет. Интернет-трафик может включать в себя множество различных типов данных, таких как веб-сайты, электронная почта, видео, и даже сообщения, которые пользователь отправляет через социальные сети [11].

Объем интернет-трафика может значительно варьироваться в зависимости от того, что пользователь делает в интернете. Например, если пользователь смотрит видео на YouTube, то используется гораздо больше трафика, чем если бы пользователь просто просматривал веб-страницы. Если пользователь отправляет большие файлы через электронную почту или загружает большие файлы из интернета, это также будет использовать больше трафика.

Управление интернет-трафиком важно для многих причин. Во-первых, это помогает контролировать использование данных на вашем устройстве, чтобы знали, сколько данных используется и когда.

Кроме того, управление трафиком может помочь улучшить скорость интернета, так как меньшее количество данных означает, что устройству не нужно обрабатывать столько информации.

Существуют различные способы управления интернет-трафиком на устройствах. Некоторые устройства имеют встроенные функции для ограничения использования данных, такие как «режим экономии данных» на Android. Также можно использовать приложения, которые помогают управлять трафиком, или можно настроить параметры сети, чтобы ограничить использование данных.

В целом, интернет-трафик является важной частью работы интернета и управления вашим устройством. Понимание того, как работает трафик, и умение управлять им может помочь сделать более осознанный выбор при использовании интернета.

2.2. Машинное обучение

Машинное обучение (Machine Learning) – это область искусственного интеллекта, которая занимается разработкой алгоритмов и методов для автоматической обработки и анализа данных с целью извлечения полезной информации и принятия решений. Машинное обучение включает в себя множество различных подходов и методов, таких как обучение с учителем, обучение без учителя, reinforcement learning и другие.

Один из основных принципов машинного обучения заключается в том, что машина должна самостоятельно обучаться на основе данных, а не быть жестко запрограммированной на выполнение определенной задачи. Для этого используются различные алгоритмы обучения, такие как методы опорных векторов, нейронные сети, деревья решений и другие.

Машинное обучение находит широкое применение в различных областях, включая медицину, финансы, маркетинг, компьютерное зрение и многие другие. С помощью машинного обучения можно, например, предсказывать цены на акции, определять наиболее эффективные рекламные кампании, распознавать объекты на изображениях и многое другое [11].

Важно отметить, что машинное обучение не является заменой человеческого интеллекта, а скорее дополняет его. Машины могут обрабатывать большие объемы данных быстрее и точнее, чем люди, но они не могут полностью заменить человеческий опыт и интуицию. Поэтому машинное обучение должно использоваться в сочетании с человеческими знаниями и опытом для достижения наилучших результатов.

Метод Random Forest

Random Forest (случайный лес) – это метод машинного обучения, разработанный Лео Брейманом. Он представляет собой ансамбль из нескольких деревьев решений. Каждое дерево строится на основе случайно выбранного подмножества данных и признаков. Композиция этих деревьев позволяет повысить точность и надежность модели, а также снизить вероятность переобучения.

Одна из ключевых особенностей Random Forest – способность работать с большим количеством признаков. Он также устойчив к выбросам и не требует предобработки данных, что делает его гибким инструментом для решения широкого спектра задач машинного обучения. Однако, Random Forest может быть недостаточно эффективным для данных с малым числом переменных, так как это может привести к созданию избыточного количества деревьев.

Кроме того, Random Forest обладает высокой интерпретируемостью. Каждое дерево в ансамбле представляет собой отдельную модель, которую можно анализировать отдельно. Это позволяет понять, какие признаки наиболее важны для принятия решений.

Наконец, Random Forest является одним из наиболее популярных методов машинного обучения благодаря своей эффективности, простоте использования и универсальности. Он может быть использован в различных областях, от классификации до регрессии, и обеспечивает надежные результаты при правильном использовании.

Метод K-NN

Алгоритм K-Nearest Neighbors (K-NN) – это простой и интуитивно понятный метод машинного обучения, основанный на идее ближайшего соседства. Он используется для решения задач классификации и регрессии.

В основе KNN лежит предположение, что объекты, находящиеся рядом друг с другом, имеют схожие характеристики. Таким образом, для определения класса нового объекта достаточно найти K его ближайших соседей в обучающем наборе данных и отнести новый объект к тому классу, который наиболее часто встречается среди этих соседей.

KNN широко применяется в различных областях, где требуется классификация или регрессия. Например, в медицине он может использоваться для диагностики заболеваний, в маркетинге – для сегментации клиентов, а в распознавании образов – для идентификации объектов на изображениях.

Преимуществами KNN являются простота реализации, отсутствие необходимости в предварительной обработке данных и высокая интерпретируемость результатов. Однако у него есть и недостатки, такие как высокая вычислительная сложность при большом количестве объектов в обучающем наборе данных и чувствительность к шуму и выбросам в данных.

Выбор оптимального значения K является ключевым моментом в использовании KNN. Слишком маленькое значение может привести к переобучению, а слишком большое – к недообучению. Обычно значение K выбирается эмпирически, исходя из специфики задачи и характеристик данных.

Logistic Regression

Логистическая регрессия (Logistic Regression) – это статистический метод, используемый для предсказания вероятности наступления определенного события на основе набора входных данных. Она применяется в задачах бинарной классификации, где результат может быть только одного из двух возможных исходов.

Логистическая регрессия использует логистическую функцию для преобразования линейного предиктора в вероятность. Линейный предиктор – это сумма произведений входных данных на их коэффициенты. Логистическая функция, известная как сигмоида, преобразует любое действительное число в значение между 0 и 1, что соответствует вероятности наступления события.

Логистическая регрессия находит широкое применение в различных областях, включая медицину (прогнозирование заболеваний), маркетинг (предсказание покупательского поведения), социологию (анализ общественного мнения) и многие другие.

Преимуществами логистической регрессии являются простота интерпретации коэффициентов, возможность учета нескольких предикторов одновременно и устойчивость к выбросам в данных. Однако она может столк-

нуться с проблемой мультиколлинеарности (когда предикторы сильно коррелируют друг с другом) и может не справиться с нелинейными зависимостями между переменными.

Для реализации логистической регрессии используются различные методы оптимизации, такие как метод максимального правдоподобия и градиентный спуск. Также применяются методы регуляризации, например, L1 и L2, чтобы предотвратить переобучение модели.

2.3. DDoS атака

DDoS-атака (Отказ в обслуживании) – это попытка злоумышленников так загрузить сервер, чтобы он просто перестал работать. Для этого на него отправляется очень большое количество запросов [15].

Разумеется, делается это не вручную – запросы автоматические и в случае с крупными атаками могут посылаться одновременно с сотен устройств. Мощностей сервера не хватает на то, чтобы обработать их все, система выходит из строя, а конечный пользователь не может получить к ней доступ.

Устройства, используемые хакерами для DDoS-атак, почти никогда не являются их собственными. Под удаленным управлением находятся целые сети ПК, зараженных вредоносным ПО, а количество запросов может достигать многих тысяч. Во время серии атак на сферу ИТ в России в январе 2023 года самая мощная DDoS отправляла на сервер 400 тысяч запросов в секунду.

У DDoS-атак может быть очень много причин. Нападать могут на сайты государственных учреждений и банков по политическим мотивам, ради кражи информации частных компаний или из-за конкуренции в бизнесе, для отвода глаз и параллельного мошенничества, вымогательства и даже ради развлечения.

Малый и средний бизнес чаще всего сталкивается с DDoS-атаками на почве конкуренции.

Классификация DDoS-атак

Существует несколько разновидностей DDoS-атак. Далее рассмотрены четыре самых популярных атаки.

UDP

UDP работает поверх протокола IP – данные просто отсылаются безо всякого контроля целостности. Поэтому злоумышленник может, например, подменить IP-адрес источника – рассылать пакеты со своего устройства, но делать, вид, что они приходят из других мест. Проверить это нельзя, и именно в таком виде они придут на сервер.

При такой атаке злоумышленник генерирует множество пакетов максимального размера и отправляет на сервер-жертву. Опасность в том, что даже если сервер закрыт на firewall, невозможно повлиять на фильтрацию таких данных до их получения сетевым интерфейсом.

Также есть методы усиления, позволяющие многократно усилить атаку. Злоумышленник рассылает совершенно нормальным серверам по всему миру запрос, в котором подменяет свой адрес на адрес жертвы в заголовках. Соответственно все сервера, на которые пришел запрос, отправляют ответ не на адрес атакующего, а на адрес жертвы, который был указан в заголовках.

Если он не работаете через UDP, его вообще можно закрыть – так делают многие провайдеры, размещая свои DNS-сервера внутри сети.

Сейчас активно внедряют новый протокол QUIC, который будет являться транспортным для HTTP3. Этот протокол работает как раз поверх UDP и, скорее всего, будет подвержен таким атакам. Пока не известно, как с ними планируют бороться. Может, разработают какие-то подходящие инструменты.

ICMP-флуд

ICMP-флуд – один из самых опасных видов DDoS-атак. Потому что злоумышленник использует рассылку для проверки работающих узлов в системе. Затем адрес атакующего меняется на адрес жертвы. ICMP-пакет, отправленный злоумышленником через усиливающую сеть, содержащую 200 узлов, будет усилен в 200 раз. Сам же ICMP является протоколом 3 уровня и используется для диагностики сети на связь между цепочками сети.

TCP SYN Flood

TCP – это протокол сетевой передачи данных в цифровом виде. Способ передачи от источника информации к пользователю. В этом способе 4 уровня передачи описанных правилами протоколов.

1. Канальный уровень.
2. Сетевой уровень.
3. Прикладной уровень.
4. Транспортный уровень.

У TCP есть механизм установки соединения. Сначала источник посылает SYN-запрос о том, что хочет установить соединение. Сервер-получатель отвечает пакетом SYN+ACK о том, что готов к соединению.

Источник отвечает ACK-пакетом, подтверждая получение SYN+ACK.

Соединение устанавливается, потому что обе стороны подтвердили готовность, и начинают передаваться данные.

Здесь уже есть проверка соответствия IP-адреса, поэтому подменить его не получится. Но атакующий может генерировать SYN-пакет, инициируя новую сессию с сервером-жертвой, а соединение не установить, не отправляя АСК. Такая атака переполняет таблицу соединений, вызывая падение производительности. На настоящие запросы просто не остается места.

Надо блокировать через firewall по превышению и настраивать лимиты по количеству SYN-пакетов в секунду, которые он ожидает для сервиса.

HTTP Flood

Это самый примитивный вид DDoS-атаки. Для атаки на сервер обычно применяется HTTP-флуд. Атакующий шлет маленький по объему HTTP-пакет, но такой, чтобы сервер ответил на него пакетом, размер которого в сотни раз больше. Даже если канал сервера в десять раз шире канала атакующего, то все равно есть большой шанс насытить полосу пропускания жертвы. А для того, чтобы ответные HTTP-пакеты не вызвали отказ в обслуживании у злоумышленника, он каждый раз подменяет свой IP-адрес на IP-адреса узлов в сети.

Методы предотвращения и защиты от DDoS-атак

Для предотвращения атак нужно сделать следующее.

1. Составить план инфраструктуры. Однозначно понять, что и где расположено, какие сервисы и серверы используются.

2. Проанализировать, какие элементы инфраструктуры должны быть доступны извне. Все, которые не должны быть – закрыть. Например, СУБД не должна быть доступна извне. Стоит в firewall ограничить доступ и сменить порт со стандартного 80, 8080 и 8000.

3. Убедиться, что IP-адреса инфраструктуры не скомпрометированы. Даже если основной сервис отразил атаку, другой элемент инфраструктуры может стать целью атаки.

Минимизация зоны атаки

Для того что бы DDoS-атака с меньшей вероятностью смогла провести успешную атаку по важным точкам инфраструктуры, нужно выполнить следующее.

1. Настроить firewall сервера. В политиках ни в коем случае нельзя оставлять настройки по дефолту. Важно закрыть все, кроме доверенных адресов и сетей.

2. Скрыть все реальные IP-адреса инфраструктуры. Периодически их менять.

3. По возможности отказаться от нешифрованного трафика. Перестать использовать HTTP и перейти на HTTPS. Это важно для безопасности в целом, но и от DDoS защищает – чтобы злоумышленники не смогли подсмотреть пакеты пользователя и понять, как их формирует пользователь, чтобы потом подделать.

4. Проверить бизнес-логику, чтобы понять, как и куда легитимный клиент должен делать запросы.

5. Если на физическом сервере находится не один сервис, важно тщательно разграничить их по ресурсам. Чтобы упавший сервис не мог съесть все ресурсы и повредить другим сервисам.

Выводы по второй главе

Во второй главе были обобщены ключевые теоретические аспекты, касающихся методов машинного обучения KNN, Logistic Regression, Random Forest.

Разъяснения, что такое интернет-трафик, как на него можно повлиять с помощью DDoS-атак, какими протоколами чаще всего идут атаки на интерне-трафик пользователя. Также были рассмотрены как минимизировать зону атаки на важные точки инфраструктуры пользователя.

3. МЕТОД DDOS АТАК НА KALI LINUX

3.1. Инструмент Slowloris

Slowloris – это бесплатный инструмент с открытым исходным кодом, доступный на Github. С помощью этого инструмента мы можем выполнить атаку типа «отказ в обслуживании» посылая botnet. Это фреймворк, написанный на Python. Этот инструмент позволяет одной машине отключать веб-сервер другой машины. Он использует совершенно законный HTTP-трафик. Он устанавливает полное TCP-соединение, а затем требует всего несколько сотен запросов через длительные и регулярные промежутки времени. В результате инструменту не нужно тратить много трафика на исчерпание доступных соединений на сервере.

Botnet – это совокупность зараженных вредоносным программным обеспечением электронно-вычислительных устройств, подключенных к глобальной сети. В список таких устройств могут входить: мобильные гаджеты, включая смартфоны, планшеты и ноутбуки, корпоративные серверы, рабочие и персональные компьютеры, устройства интернета вещей. При помощи установленного на них вредоносного кода, злоумышленнику удается получить удаленный контроль над оборудованием и использовать его в своих незаконных целях, часто без ведома владельца и пользователей [16].

Сокеты – это технология передачи низкого уровня, на его основе реализованы многие сетевые протоколы.

Функционал инструмента Slowloris:

1) отправка целевому объекту несколько запросов botnet;
2) проведение DDoS-атак на любой веб-сервер;
3) генерация большого трафика ботнетов для атаки отказа в обслуживании.

3.2. Реализация инструмента Slowloris

Для начала атаки с помощью Slowloris нужно установить и внедрить инструмент в ОС Kali Linux, а именно.

1. Запуск Kali Linux, а затем открытие терминала (рисунок 1).

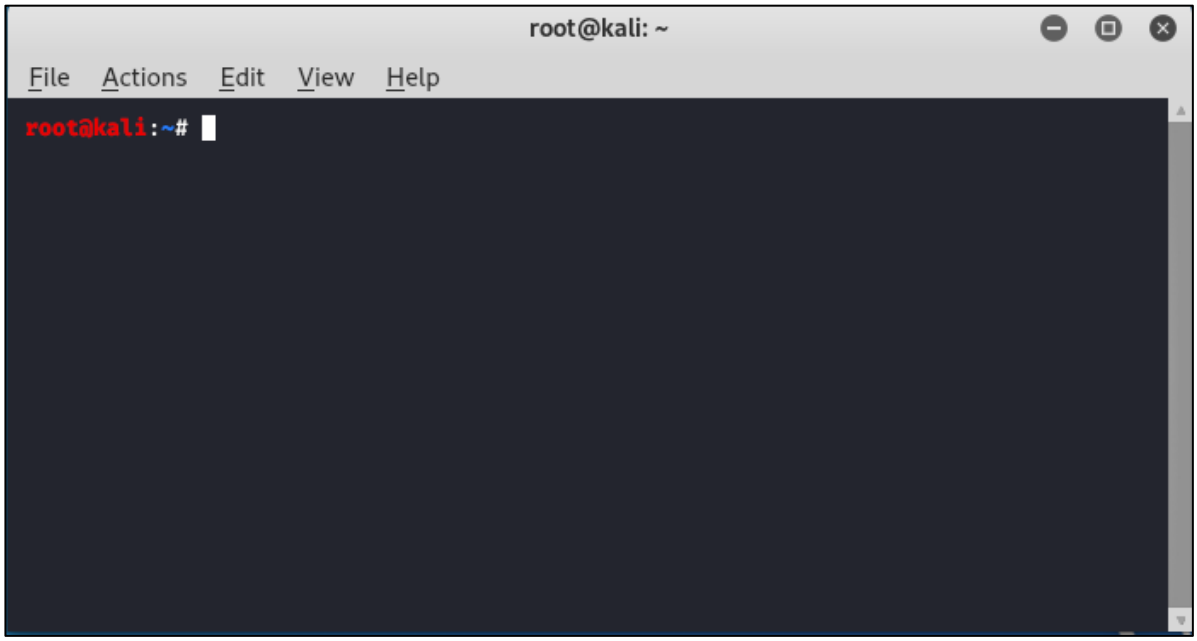

Рисунок 1 – Терминал

2. Создается новый каталог на рабочем столе с именем Slowloris, используя следующую команду `mkdir Slowloris` (рисунок 2).

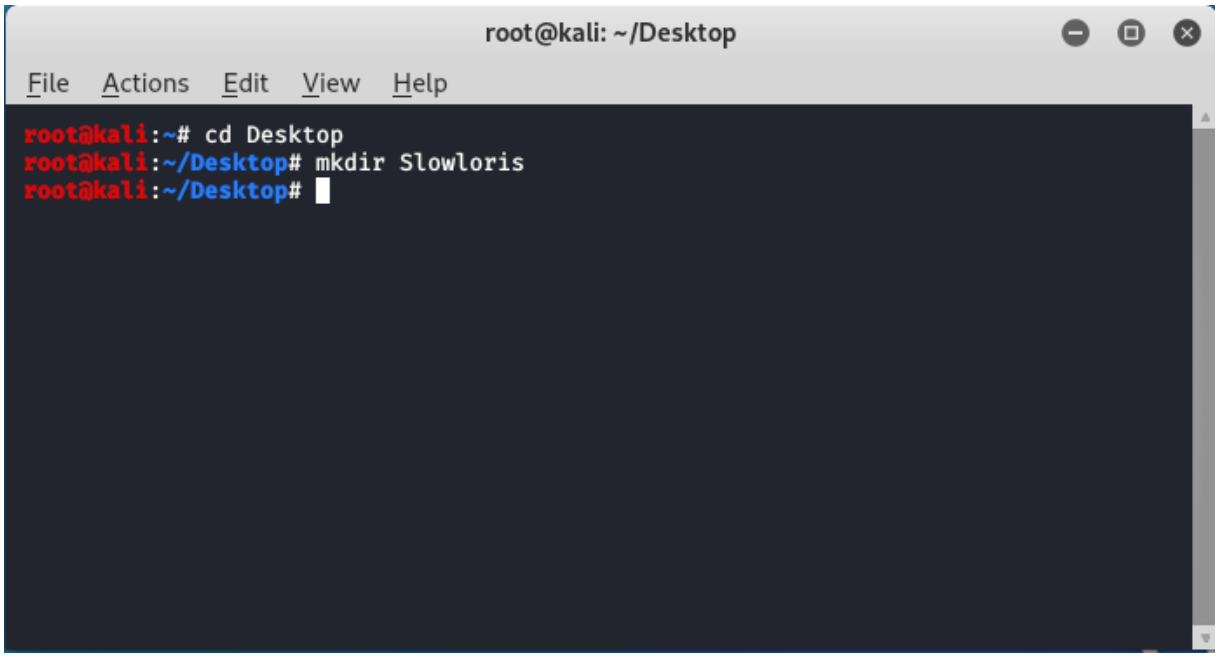

Рисунок 2 – Создание нового каталога Slowloris

3. Переход в каталог, котором нужно создать папку с именем Slowloris, `cd Desktop`, `mkdir Slowloris` (рисунок 3).

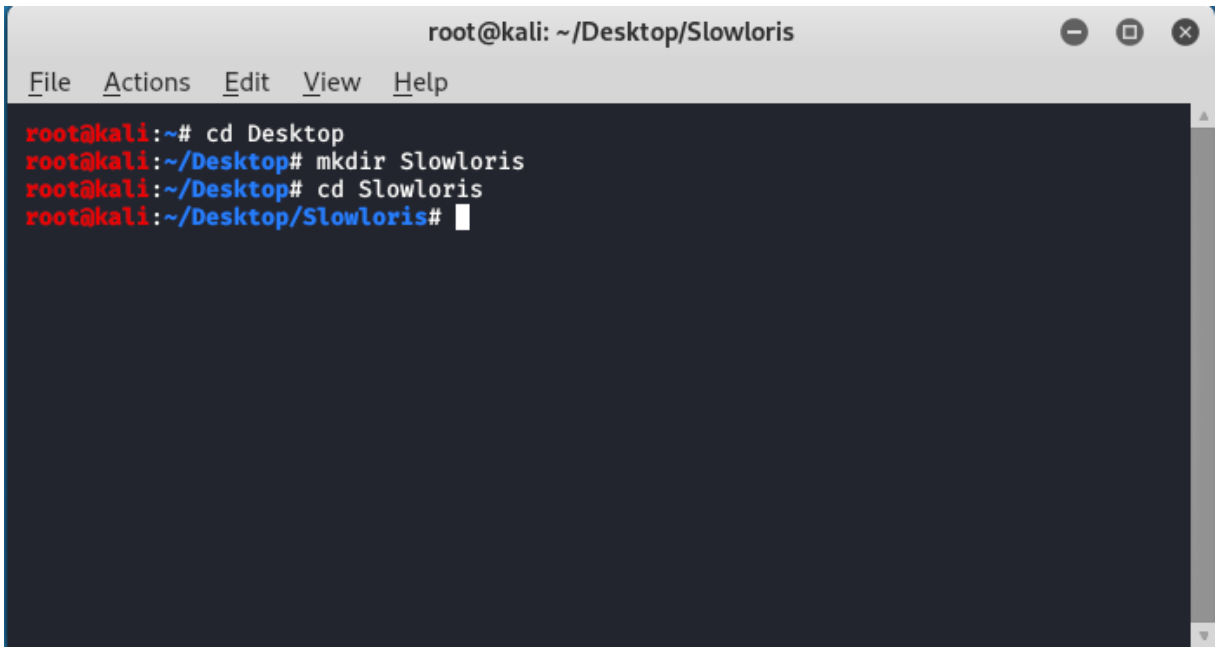

Рисунок 3 – Переход в директорию Slowloris

4. Теперь нужно клонировать инструмент Slowloris с Github, чтобы установить его на компьютер с Kali Linux. Для этого нужно ввести следующий URL-адрес (https://github.com/gkbrk/slowloris.git) в терминал в созданном каталоге Slowloris. Таким образом был успешно установлен инструмент для DDoS-атак (рисунок 4).

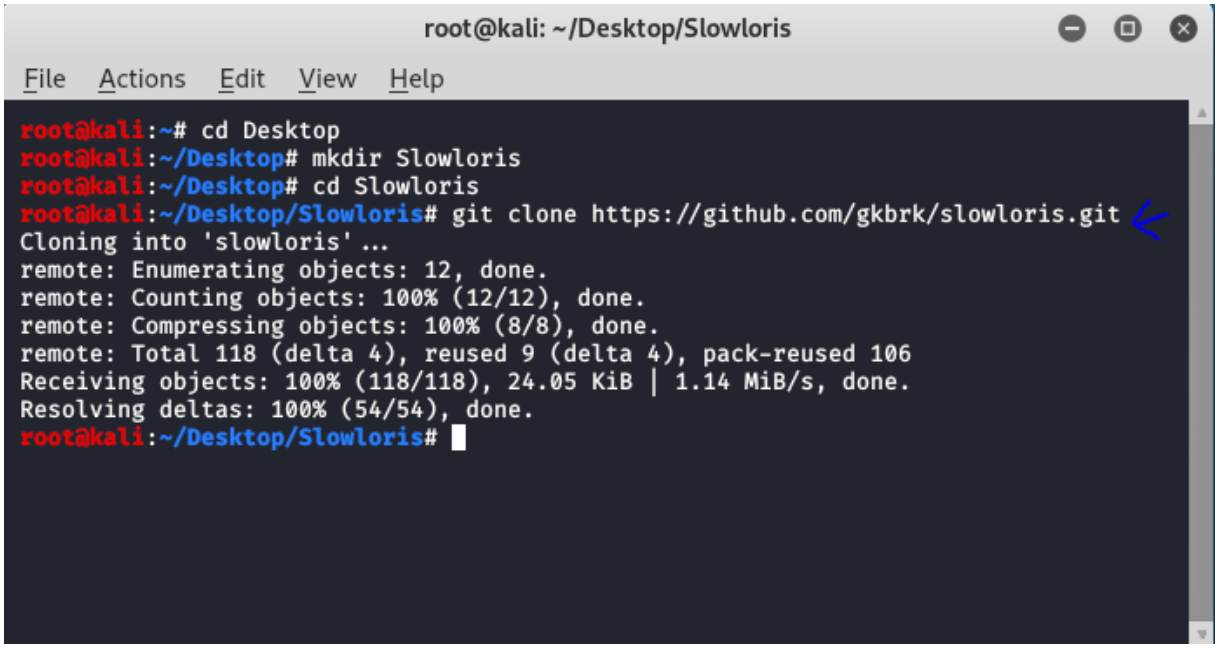

Рисунок 4 – Клонирование Slowloris с репозитория Github

5. Теперь нужно проверить IP-адрес компьютера, с которого будет DDoS атак и IP-адрес на который будет атака, чтобы выполнить следующую команду такого типа `ifconfig` в kali linux (рисунок 5) и `arp -a` в Windows (рисунок 6).

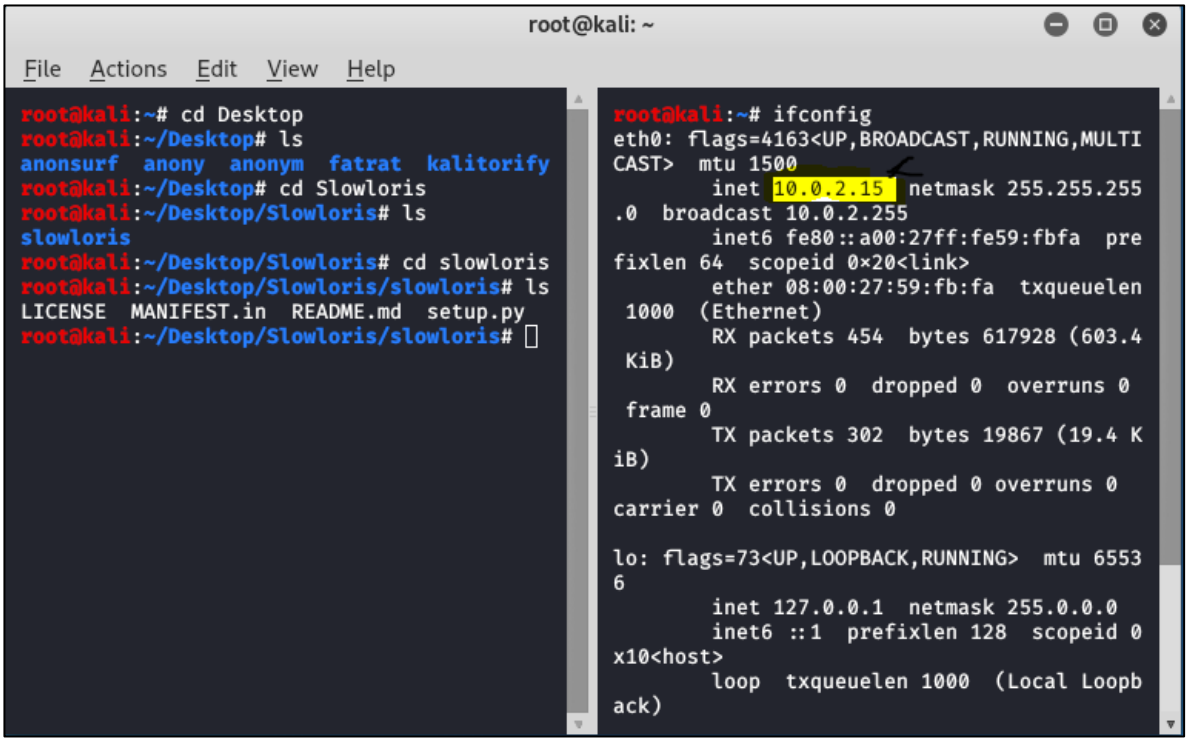

Рисунок 5 – IP-адрес Kali Linux

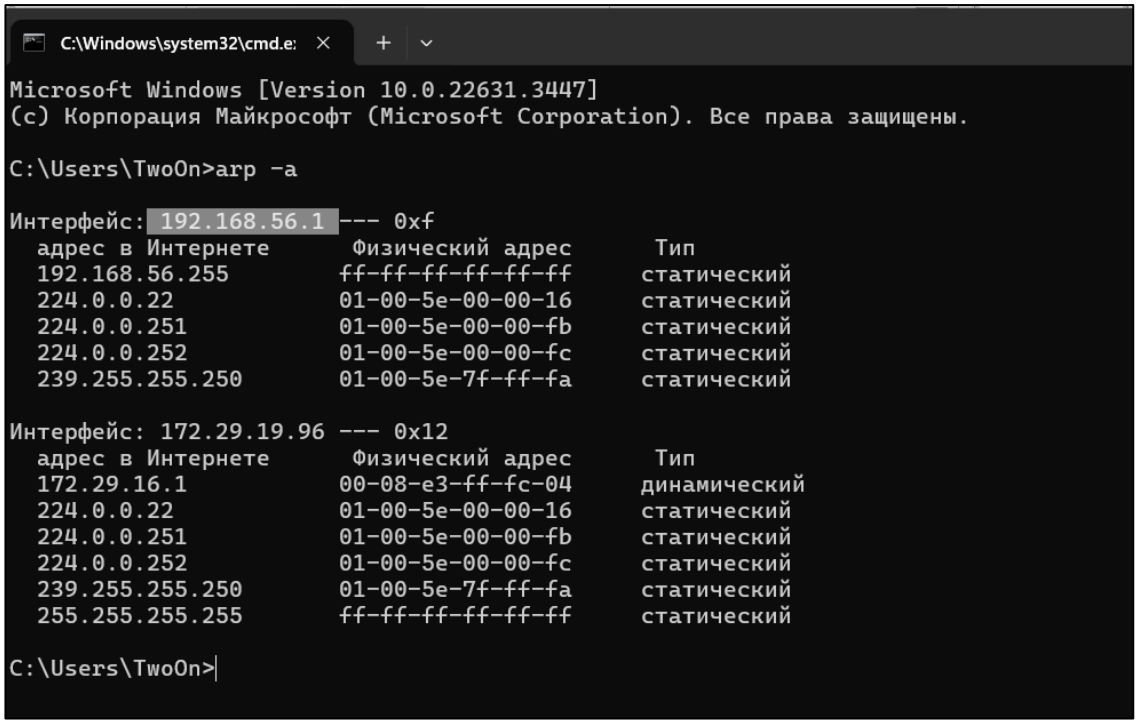

Рисунок 6 – IP-адрес Windows

6. Запуск сервера Apache, используя следующую команду `sudo service apache2 start` (рисунок 7).

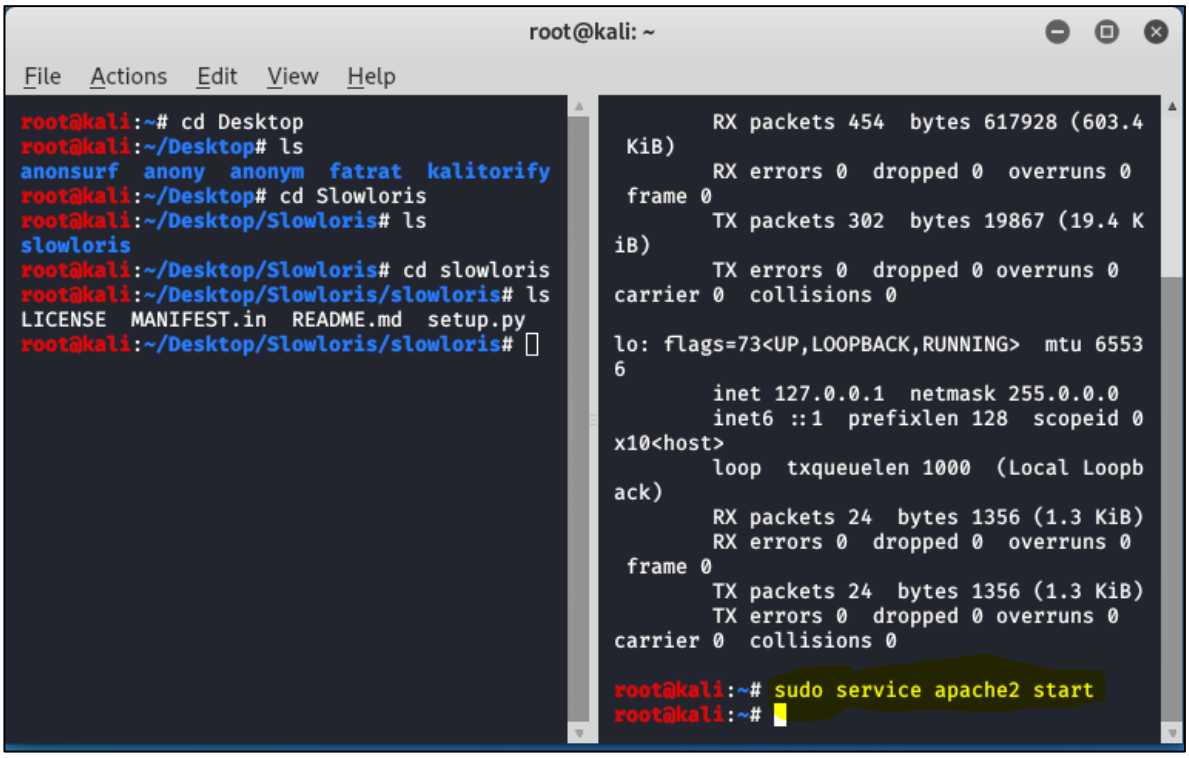

Рисунок 7 – Запуск Apache

7. Сервер находится в активном состоянии (рисунок 8), это означает, что он работает.

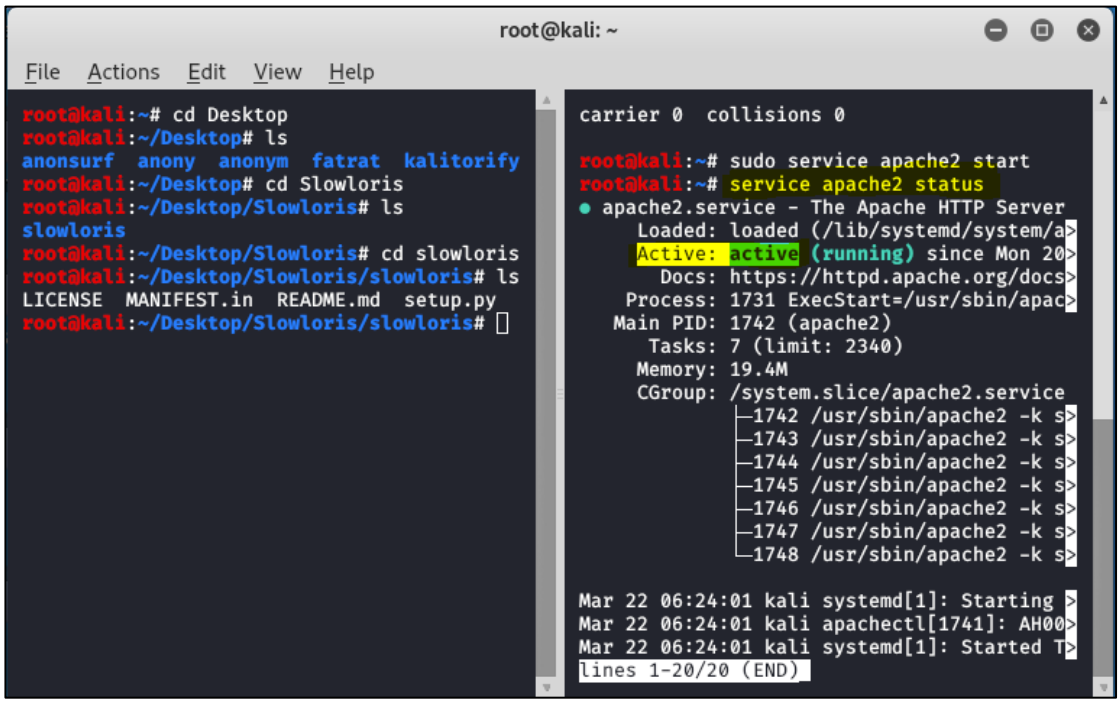

Рисунок 8 – Статус сервера

8. Теперь пришло время запустить инструмент с помощью следующей команды `python3 slowloris.py [IP-адрес] -s 500` (рисунок 9). В этой команде `[IP-адрес]` – это адрес пользователя на которого совершается DDoS-атака, `-s 500` – это количество сокетов используем для атаки, `slowloris.py` – это открытый код на языке Python с помощью которого и запускается DDoS-атака.

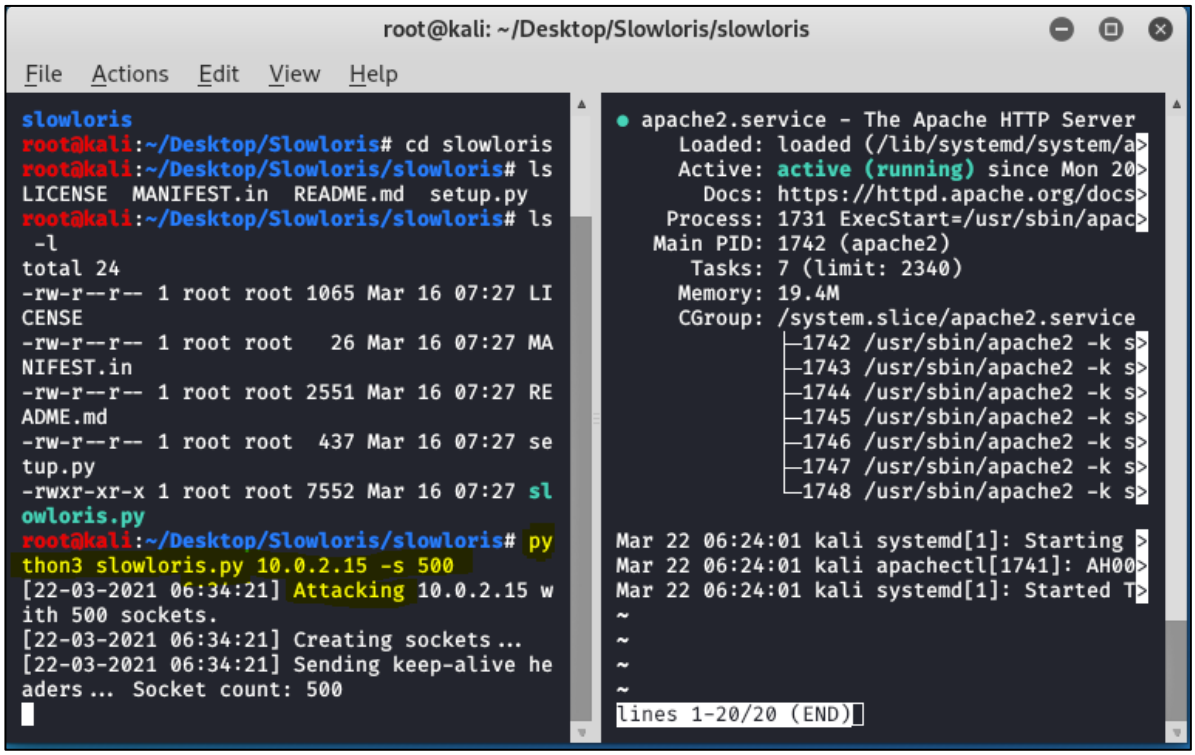

Рисунок 9 – Запуск Slowloris

9. Инструмент начал атаковать тот конкретный IP-адрес, который выступал в качестве жертвы злоумышленника, чтобы проверить, работает ли он или нет. На рисунке 10 представлен браузер, и в строке URL введен IP-адрес на который производилась DDoS-атака, и видно, что сайт только загружается и загрузка, но не открытие, вот как работает инструмент Slowloris.

Рисунок 10 – Браузер

Выводы по третьей главе

В третьей главе были описаны процесс установки инструмента Slowloris на ОС Kali Linux и эмуляция DDoS-атаки с ноутбука на персональный компьютер через IP-адрес по нескольким протоколам. Был разобран функционал инструмента Slowloris, какие команды нужны для его установки и запуска атак. Был произведен запуск сервера Apache через терминал.

4. СБОР ИНТЕРНЕТ-ТРАФИКА В НАБОР ДАННЫХ

4.1. Анализ требований

Функциональные требования

Приложение должно иметь следующие функциональные требования.

1. Доступ к сети, на которой происходит сбор данных.
2. Подходящий протокол передачи данных (например, TCP или UDP).
3. Определение формата собираемых данных (бинарные данные, текстовые данные, пакеты приложений и т.д.).
4. Обеспечивать достаточный объем собираемых данных для обучения модели.
5. Обрабатывать собранные данные, удаляя ненужные и искаженные данные.

Нефункциональные требования

Система должна демонстрировать следующие требования, иначе сбор интернет-трафика может не быть точным.

1. Скорость сбора данных: система должна быть способна собирать данные с достаточной скоростью для отслеживания изменений в трафике.
2. Точность сбора данных: система должна собирать данные с высокой точностью, чтобы избежать потери или искажения информации.
3. Гибкость системы: система должна быть гибкой и настраиваемой для различных сценариев сбора данных.
4. Надежность системы: система должна обеспечивать надежную и стабильную работу без сбоев и потерь данных.
5. Безопасность системы: система должна обеспечивать защиту данных от несанкционированного доступа и утечки информации.

4.2. Варианты сбора

Сбор интернет-трафика в набор данных – это процесс сбора информации о сетевом трафике для создания набора данных, который может быть

использован для обучения и тестирования моделей машинного обучения [13]. Собранные данные могут быть использованы для анализа сети, обнаружения сетевых атак, оптимизации работы сети и других задач. Этот процесс может включать в себя сбор различных видов данных, таких как метаданные о соединениях, содержимое пакетов, информация о протоколах и т.д.

Далее приведет список вариантов сбора интернет-трафика.

1. Использование систем глубокого обучения. Эти системы могут автоматически анализировать трафик и определять, какие данные являются наиболее важными для анализа.

2. Применение методов машинного обучения для анализа трафика. Машинное обучение может использоваться для выявления аномалий в трафике и определения причин их возникновения.

3. Использование протоколов передачи данных, таких как TCP и UDP, для сбора бинарных данных и пакетов приложений.

4. Обработка собранных данных с помощью специализированного программного обеспечения для удаления ненужных и искаженных данных.

5. Хранение собранных данных в удобном для анализа формате, таком как CSV или SQL.

6. С помощью программ Wireshark, tcpdump, SolarWinds Network Bandwidth Analyzer.

Большую часть из списка вариантов имеет программы Wireshark, tcpdump, SolarWinds. Удобный интерфейс у программы Wireshark.Таким образом используется в данной работе программа Wireshark.

Чтобы приступить к сбору интернет-трафика, надо выбрать интересующую сеть (рисунок 11).

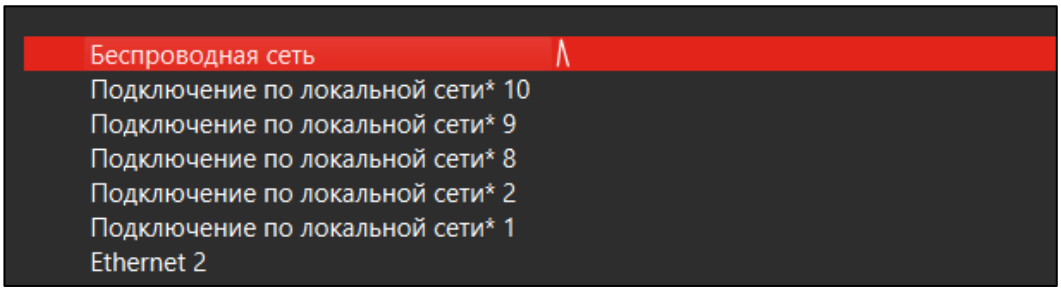

Рисунок 11 – Выбор сети

После выбора сети программа начнет показывать проходящий трафик через эту сеть (рисунок 12).

Time	Source	Destination	Protocol	Length	Info
11.498194	173.194.222.101	172.29.19.96	TLSv1.2	242	Application Data
11.498194	173.194.222.101	172.29.19.96	TLSv1.2	93	Application Data
11.498194	173.194.222.101	172.29.19.96	TCP	93	[TCP Retransmissio
11.498389	172.29.19.96	173.194.222.101	TCP	66	26497 → 443 [ACK]
11.512512	172.29.19.96	173.194.222.101	TLSv1.2	89	Application Data
11.512777	172.29.19.96	173.194.222.101	TLSv1.2	93	Application Data
11.522487	172.29.19.96	216.239.32.180	TLSv1.2	2124	Application Data
11.530262	172.29.19.96	37.75.250.46	DNS	81	Standard query 0xb
11.531908	62.128.100.45	172.29.19.96	TCP	60	443 → 26448 [ACK]
11.532681	172.29.19.96	216.239.32.180	TLSv1.2	352	Application Data,
11.532825	37.75.250.46	172.29.19.96	DNS	81	Standard query res
11.565508	173.194.222.101	172.29.19.96	TCP	60	443 → 26497 [ACK]
11.572138	216.239.32.180	172.29.19.96	TCP	60	443 → 26500 [ACK]
11.572138	216.239.32.180	172.29.19.96	TCP	60	443 → 26500 [ACK]
11.581542	216.239.32.180	172.29.19.96	TCP	60	443 → 26500 [ACK]
11.582158	216.239.32.180	172.29.19.96	TLSv1.2	93	Application Data
11.627589	172.29.19.96	216.239.32.180	TCP	54	26500 → 443 [ACK]
11.801872	216.239.32.180	172.29.19.96	TLSv1.2	563	Application Data
11.803417	216.239.32.180	172.29.19.96	TLSv1.2	85	Application Data
11.803417	216.239.32.180	172.29.19.96	TLSv1.2	93	Application Data
11.803512	172.29.19.96	216.239.32.180	TCP	54	26500 → 443 [ACK]
11.809894	172.29.19.96	216.239.32.180	TLSv1.2	93	Application Data
11.864831	216.239.32.180	172.29.19.96	TCP	60	443 → 26500 [ACK]
11.879096	5e:13:77:5d:4e:ff	Broadcast	ARP	56	Who has 172.29.16.
11.879096	5e:13:77:5d:4e:ff	Broadcast	ARP	60	Who has 172.29.16.
12.720815	5e:13:77:5d:4e:ff	Broadcast	ARP	60	Who has 172.29.16.
12.720815	5e:13:77:5d:4e:ff	Broadcast	ARP	60	Who has 172.29.16.

Рисунок 12 – Трафик через беспроводную сеть

Программа имеет следующий интерфейс:

1) нумерация пакета (No.);
2) время (Time);
3) источник (Source);
4) назначение (Destination);
5) протокол (Protocol);
6) длина (Length);

7) информация (Info).

Чтобы остановить сбор трафика, нужно нажать на красный квадрат (рисунок 13).

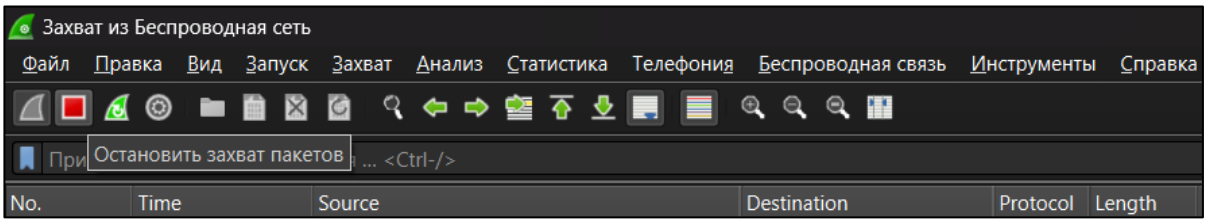

Рисунок 13 – Остановка захвата трафика

После остановки захвата интернет-трафика нужно сохранить его в нужный формат (рисунок 14).

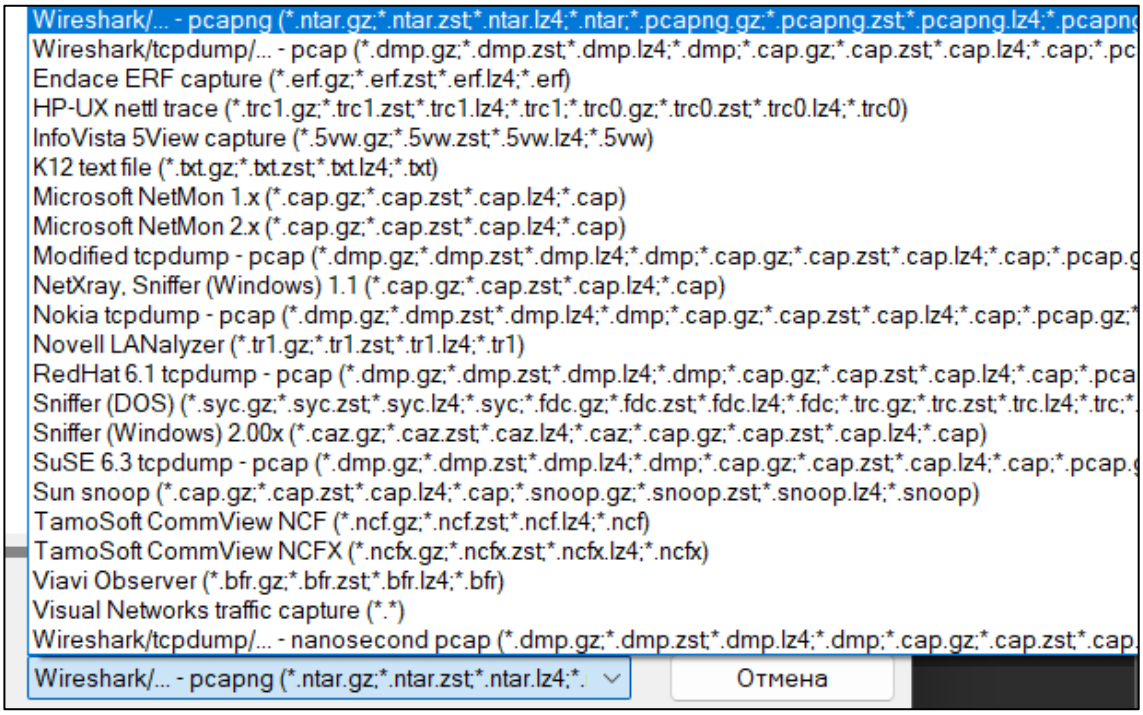

Рисунок 14 – Выбор сохранения в виде файла

Интернет-трафик можно экспортировать в нужный формат (рисунок 15). Экспорт используется для сохранения интернет-трафика в набор данных, а именно.

1. Обычный текст.
2. CSV.
3. Массива С.

4. PSML XML.
5. PDML XML.
6. JSON.

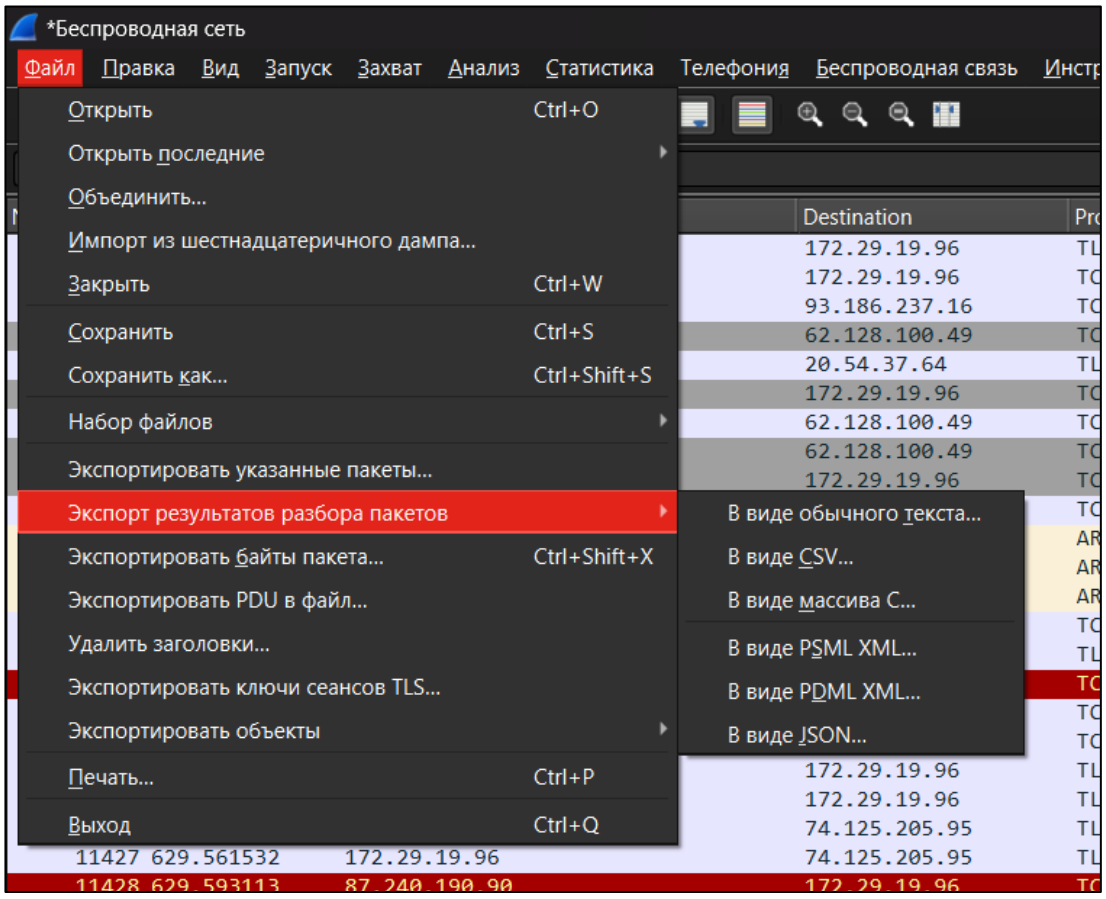

Рисунок 15 – Экспорт результата в набор данных

4.3. Собранный набор данных

CSV-файл (рисунок 16) для тестирования был собран через приложение Wireshark. Количество записей в наборе данных: 120 548.

Общее количество важных признаков из набора данных 13 по оси Y, по оси X количество запросов, а именно:

1) TCP – 56 804;
2) TLSv1.3 – 54 443;
3) TLSv1.2 – 7 452;
4) DNS – 1 249;
5) SSLv2 – 291;
6) SSL – 86;

7) SSDP – 64;
8) ICMP – 57;
9) HTTP – 51;
10) IGMPv3 – 32;
11) OCSP – 10;
12) TLSv1 – 4;
13) ARP – 4.

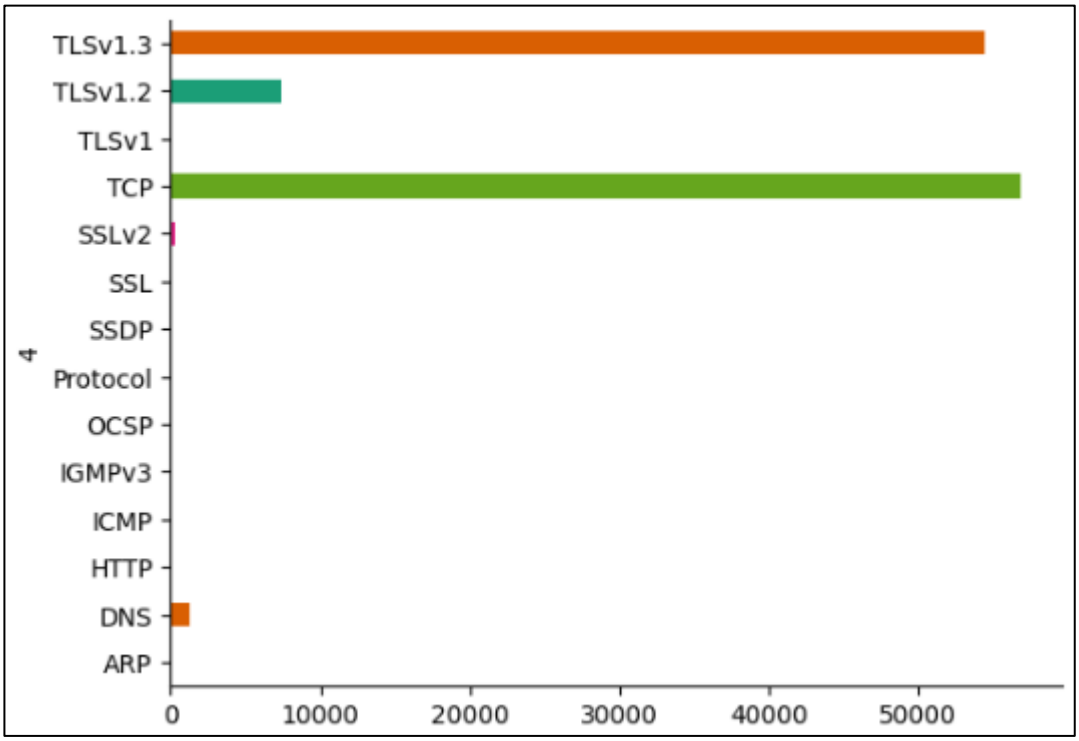

Рисунок 16 – Набор данных для тестирования

Выводы по четвертой главе

В четвертой главе был показан сбор интернет-трафика с помощью программы Wireshark. Сбор трафика проходил на персональном компьютере пользователя, на который проходила эмуляция DDoS-атаки. Собранный интернет-трафик был сохранен в CSV файл как валидационные данные для машинного обучения.

5. РАЗРАБОТКА АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

Эффективность алгоритмов машинного обучения, таких как Random Forest (RF), K-Nearest Neighbors (KNN) и Logistic Regression (LR), для анализа интернет-трафика зависит от специфики задачи, качества данных и требований к скорости обработки.

Random Forest является мощным методом, который объединяет множество деревьев решений для повышения точности и устойчивости к переобучению. Он хорошо подходит для задач классификации и регрессии, особенно когда данные содержат много признаков или имеют нелинейные зависимости. RF также способен обрабатывать пропущенные значения и устойчив к выбросам [12].

K-Nearest Neighbors (KNN) – это простой и интуитивно понятный метод, который классифицирует новый объект на основе его близости к обучающим примерам. KNN требует меньше предположений о данных и может быть легко адаптирован под разные задачи. Однако он может быть медленным для больших объемов данных и чувствителен к выбору параметра K.

Logistic Regression используется для задач бинарной классификации, где результат может быть только одного из двух возможных исходов. LR предполагает линейную связь между входными данными и результатом, что может ограничивать его применимость в сложных ситуациях. Однако он быстр в обучении и интерпретации, а также устойчив к выбросам.

Для обучения был использован набор данных, взятый с сайта Kaggle.

5.1. Алгоритм Random Forest

Random Forest может использоваться для анализа интернет-трафика для различных целей, таких как обнаружение DDoS-атак, анализ поведения пользователей и оптимизация маршрутизации. Вот основные этапы разработки модели Random Forest для анализа интернет-трафика.

1. Данные разделяются на обучающую и тестовую выборки (рисунок 17). Обучающая выборка будет использоваться для обучения модели Random Forest, а тестовая выборка – для оценки ее точности.

2. Выбираются гиперпараметры модели: количество деревьев в ансамбле (рисунок 18).

3. Построение модели: использовать библиотеку машинного обучения, такую как Scikit-Learn, для создания модели Random Forest (рисунок 19).

4. Обучение модели: обучение каждого дерева на обучающей выборке (рисунок 20).

5. Оценка точности: оценка точности модели на тестовой выборке.

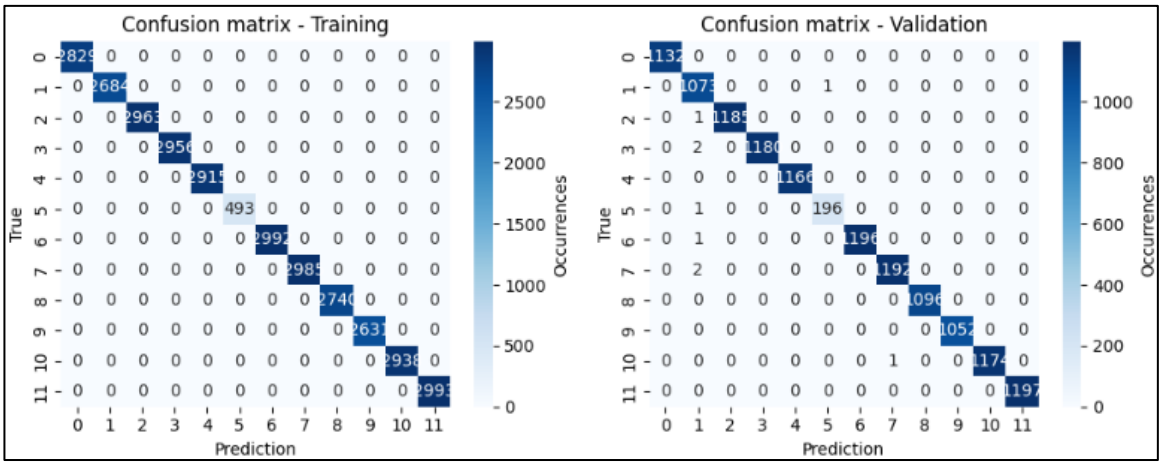

Рисунок 17 – Разделение на обучающую и тестовую выборку

```
param_range = [1, 5, 10, 30, 50, 100]
train_scores, val_scores = validation_curve(
  RandomForestClassifier(), X_train, y_train,
param_name="n_estimators", param_range=param_range, cv=5
                                )
```

Рисунок 18 – Код для выборки параметров алгоритма Random Forest

```
rf_clf = RandomForestClassifier()
rf_clf.fit(X_train, y_train)
y_train_pred_rf = rf_clf.predict(X_train)
y_val_pred_rf = rf_clf.predict(X_val)
```

Рисунок 19 – Код алгоритма Random Forest

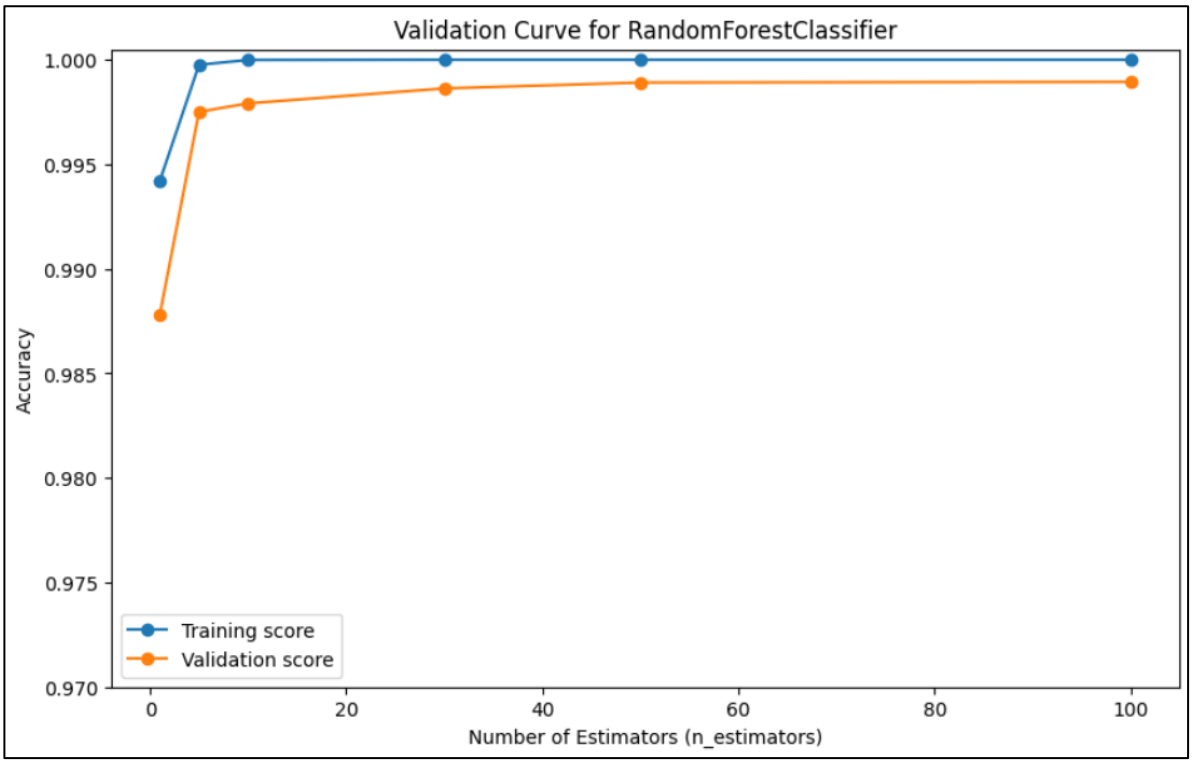

Рисунок 20 – Обучение модели

Алгоритм Random Forest обучился на 99,52%.

5.2. Алгоритм Logistic Regression

Для эффективного применения логистической регрессии в анализе интернет-трафика необходимо выполнить следующие шаги.

1. Вероятность наступления события на основе входных данных.
2. Сбор данных: собрать данные о сетевом трафике, включая информацию о времени, объеме, типе трафика и т.д.
3. Предварительная обработка данных: удалить выбросы, нормализовать данные и преобразовать категориальные переменные в числовые (рисунок 21).
4. Построение модели: использовать библиотеку машинного обучения, такую как Scikit-Learn, для создания модели логистической регрессии (рисунок 22).
5. Обучение модели: обучить модель на собранных данных (рисунок 23).

6. Тестирование модели: оценка производительности модели на тестовых данных.

```
# Разделение на признаки и целевую переменную
features = df.drop('label', axis=1)
target = df['label']

# Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(fea-
tures, target, test_size=0.3, random_state=30)

# Масштабирование признаков
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Рисунок 21 – Код для обработки данных

```
model_name = "Логистическая регрессия (Scikit-learn)"

model = LogisticRegression(solver='saga', max_iter=10000)
model.fit(X_train_scaled, y_train_encoded)
y_pred = model.predict(X_test_scaled)
model_result = {
    'name': model_name,
    'speed': round(end_time - start_time, 2),
    'accuracy': make_classification_report(labeler,
y_test_encoded, y_pred)
}
model_results_list.append(model_result)
```

Рисунок 22 – Код построения модели

	benign	ddos_dns	ddos_ldap	ddos_mssql	ddos_netbios	ddos_ntp	ddos_snmp	ddos_ssdp	ddos_syn	ddos_tftp	ddos_udp	ddos_udp_lag	All
benign	1666	2	3	0	1	0	0	0	0	0	0	0	1672
ddos_dns	0	1524	38	0	0	0	0	0	3	0	0	0	1565
ddos_ldap	0	6	1757	11	4	0	0	0	0	0	0	0	1778
ddos_mssql	0	3	64	1734	22	0	0	0	0	0	0	0	1823
ddos_netbios	0	0	0	59	1676	0	0	0	0	0	0	0	1735
ddos_ntp	0	1	0	0	0	313	0	0	0	0	0	0	314
ddos_snmp	0	0	0	0	2	0	1789	0	0	0	0	0	1791
ddos_ssdp	0	0	0	0	1	1	3	1834	0	0	0	0	1839
ddos_syn	0	0	0	0	0	0	0	0	1640	8	0	0	1648
ddos_tftp	0	0	0	0	0	0	1	0	37	1525	0	0	1563
ddos_udp	0	0	0	0	0	0	0	0	0	1	1743	0	1744
ddos_udp_lag	0	0	0	0	0	0	0	1	0	0	0	1799	1800
All	1666	1536	1862	1804	1706	314	1793	1835	1680	1534	1743	1799	19272

Рисунок 23 – Обучение модели

Алгоритм Logistic Regression обучился на 98,65%.

5.3. Алгоритм K-NN

Для эффективного применения KNN в анализе интернет-трафика необходимо выполнить следующие шаги.

1. Предварительная обработка данных: удалить выбросы, нормализовать данные и преобразовать категориальные переменные в числовые.

2. Выбор значения K: определить оптимальное значение K, которое будет использоваться для расчета ближайших соседей (рисунок 24).

3. Построение модели: использовать библиотеку машинного обучения, такую как Scikit-Learn, для создания модели KNN.

4. Обучение модели: обучить модель на собранных данных (рисунок 25).

```
param_range = [1, 3, 5, 7, 9]
train_scores, test_scores = validation_curve(
    KNeighborsClassifier(), X_train, y_train,
    param_name="n_neighbors", param_range=param_range, cv=5
)
```

Рисунок 24 – Код для выборки параметров K

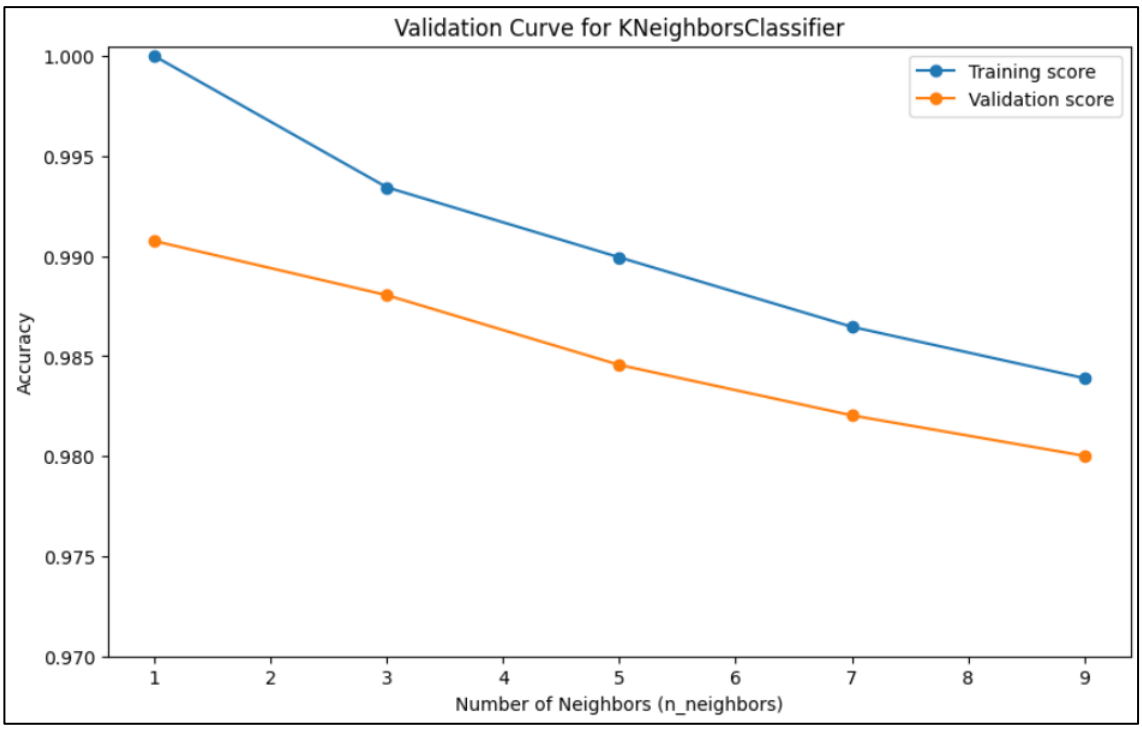

Рисунок 25 – Обучение

Алгоритм KNN обучился на 98,13%.

Выводы по пятой главе

В пятой главе была описана реализация методов машинного обучения K-NN, RandomForest, Logistic Regression. Была проведена оптимизация гиперпараметров для RandomForest и k-NN. Исходя из результатов обучения методов. самым точным метод оказался RandomForest, который рекомендуется использовать при анализе сетевого трафика для детекции DDoS-атак.

Таблица 1– Результаты обучения методов

Метод	**Время обучения**	**Метрика**	**Значение**
K-NN	3 минуты 11 секунд	accuracy (точность)	98,13%
Random Forest	34 секунды	accuracy (точность)	99,52%
Logistic Regression	2 часа 03 минуты	accuracy (точность)	98,65%

ЗАКЛЮЧЕНИЕ

В рамках данной работы были разработаны методы машинного обучения для анализа структуры интернет-трафика. Для этого были реализованы следующие задачи.

1. Собран интернет-трафика для тестирования разработанных моделей.

2. Разработаны и обучены алгоритмы машинного обучения.

3. Проведено тестирование алгоритмов машинного обучения.

Для реализации DDoS-атак были установлены VM Kali Linux на ПК и инструмент Slowloris, который находится в открытом доступе с открытым исходным кодом.

Сбор интернет-трафика в набор данных был реализован с помощью программы Wireshark. В наборе данных были выделены 13 наборов признаков, по которым можно вычислить DDoS-атаки.

Примененные алгоритмов машинного обучения (Random Forest. Logistic Regression. k-NN) успешно выявляли DDoS-атаки с точностью больше 90% за счет оптимизации гиперпараметров и признаков в наборе данных. Для детекции DDoS-атаки рекомендуется применять алгоритм RandomForest с точностью 99,52%.

Для дальнейшего развития обученную модель RandomForest можно реализовать в виде библиотеки, которую можно подключить при разработке программного обеспечения для защиты компьютера от DDoS-атак.

ЛИТЕРАТУРА

1. Alduailij M., Khan Q. W., Tahir M., Sardaraz M., Alduailij M., Malik F. Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method. Symmetry, 2022. – 1095 с.

2. Apache [Электронный ресурс]. URL: https://blog.skillfactory.ru/glossary/apache/ (дата обращения: 15.04.2024 г.).

3. Azizan A.H., Mostafa S.A., Mustapha A., Foozy C.F.M., Wahab M.H. A., Mohammed M.A., Khalaf B.A. A Machine Learning Acroach for Improving the Performance of Network Intrusion Detection Systems. – Annals of Emerging Technologies in Computing, 2021. – 208 с.

4. Cherdantseva Y., Burnap P., Blyth A., Eden P., Jones, K. Cyber-security in the age of Industry 4.0: A review of threats and emerging defence technologies. Computers in Industry, 2018. – 206 с.

5. DDoS–атаки. Причины возникновения, классификация и защита от DDoS-атак. [Электронный ресурс]. URL: https://efsol.ru/articles/ddos-attacks/ (дата обращения: 06.04.2024 г.).

6. DDoS-атаки: что это такое и как защитить свои сервисы. [Электронный ресурс]. URL: https://habr.com/ru/companies/x-com/articles/761036/ (дата обращения: 03.04.2024 г.).

7. Mirkovic J., Reiher P. A taxonomy of DDoS attack and DDoS defense mechanisms. – ACM SIGCOMM Computer Communication Review, 2004. – 53 с.

8. Singh S., Chhabra S. DDOS attacks: detection and prevention mechanisms–a review. – Journal of Ambient Intelligence and Humanized Computing, 2019. – 3539 с.

9. Tripathi S., Gupta B., Almomani A. Hadoop based defense solution to handle distributed denial of service DDoS attacks., 2013. – 164 с.

10. Vanitha K., Mala C. A survey on detection and prevention of distributed denial of service (DDoS) attacks. – Journal of Ambient Intelligence and Humanized Computing, 2019. – 1931 с.

11. Александр К., Евгений Х. Машинное обучение для анализа сетевого трафика. – СПАРК, 2019. – 596 с.

12. Аманжолов, Олжас Маратулы. Исследование методов и средств обнаружения DDoS-атак – Молодой ученый, 2023. – 58 с.

13. Бирюков А. А. Информационная безопасность: защита и нападение // А. А. Бирюков. – М.: ДМК-Пресс, 2013. – 474 с.

14. Ботнеты: что это и как они влияют на кибербезопасность [Электронный ресурс]. URL: https://habr.com/ru/companies/x-com/articles/761036/ (дата обращения: 03.04.2024 г.).

15. Йен Г., Йошуа Б. Глубокое обучение: адаптивное вычисление и машинное обучение. – MIT Press, 2016. – 393 с.

16. Лео Брейман Random Forests. Machine Learning. – UC Berkeley, 2001. – 34 с.

17. Метрические методы [Электронный ресурс]. URL: https://education.yandex.ru/handbook/ml/article/metricheskiye-metody (дата обращения: 01.03.2024 г.).

18. Обзор Kali Linux 2021.2 [Электронный ресурс]. URL: https://habr.com/ru/companies/ruvds/articles/566164/ (дата обращения: 25.03.2024 г.).

19. Протоколы семейства TCP/IP. Теория и практика [Электронный ресурс]. URL: https://habr.com/ru/companies/ruvds/articles/759988/ (дата обращения: 25.02.2024 г.).

20. Шайлендра С., Гопал К. С. Анализ трафика с использованием алгоритмов машинного обучения. – ICCCA, 2021. – 608 с.