МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Факультет Энергетический

Кафедра Информатика и вычислительная техника

**КУРСОВОЙ ПРОЕКТ**  
по Протоколом вычислительных систем на тему

Разработка приложений, реализующих атаку, идентификацию, предотвращение UDP Flood

Выполнил ст. гр. ВМК-20

Шарин Р.А.

Проверил доцент кафедры ИВТ и ПИ

Машкин В.А.

г. Чита

2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Факультет Энергетический

Кафедра Информатика и вычислительная техника

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовому проекту

по Протоколам вычислительных систем на тему

Разработка приложений, реализующих атаку, идентификацию, предотвращение UDP Flood

Выполнил студент группы ВМК-20 Шарин Р.А.

Руководитель работы доцент кафедры ИВТ и ПМ Машкин В.А.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Факультет Энергетический

Кафедра Информатика и вычислительная техника

**ЗАДАНИЕ**

на курсовую работу

По дисциплине «Протоколы вычислительных систем»

Студенту Шарину Роману Антоновичу специальности 09.03.01 Информатика и вычислительная техника (Вычислительно машинный комплекс)

1. Тема курсовой работы (проекта) «Разработка приложений, реализующих атаку, идентификацию, предотвращение UDP Flood»;
2. Срок подачи студентом законченной работы 16.06.2023;
3. Исходные данные к работе (проекту): Необходимо разработать приложения, которые позволят атаковать, идентифицироваться и защититься от атаки типа UDP Flood;

Дата выдачи задания: 24.03.2023

Руководитель работы: доцент кафедры ИВТ и ПМ Машкин В.А.

Подпись руководителя работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание принял к исполнению «\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_ г.

Подпись студента: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РЕФЕРАТ**

Курсовой проект на по дисциплине «Протоколы вычислительных систем» на тему «Разработка приложений, реализующих атаку, идентификацию, предотвращение UDP Flood». Работа изложена на 41 листе машинописного текста и состоит из: введения, 2 глав, заключения и списка источников;

Ключевые слова: UDP Flood, сокет, порт, протокол, Ethernet-кадр.

Краткое содержание работы: Работа посвящена изучению метода атаки UDP Flood, уязвимостей протокола UDP и защите от представленного типа атаки.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| КАЛЕНДАРНЫЙ ГРАФИК выполнения курсового проекта | | | | | | | | | | | Утверждаю  Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. | | | | | | |
| *№* | *Наименование раздела курсовой работы* | *Март* | | | | *Апрель* | | | | *Май* | | | | *Июнь* | | | |
| 1 | Получение задания |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Поиск источников информации |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Оформление теоретической части |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Реализация необходимых приложений |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Тестирование приложений |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Оформление практической части |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Оформление пояснительной записки |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Защита работы |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 7](#_Toc137787644)

[Глава 1: Теоретическая часть 8](#_Toc137787645)

[Глава 1.1: Протокол UDP — достоинства и недостатки 8](#_Toc137787646)

[Глава 1.2: Уязвимости протокола UDP 11](#_Toc137787647)

[Глава 1.3: Вид атак UDP Flood 12](#_Toc137787648)

[Глава 1.4: Идентификация UDP Flood атаки 14](#_Toc137787649)

[Глава 1.5: Защита от UDP Flood атаки 15](#_Toc137787650)

[Глава 2: Практическая часть 16](#_Toc137787651)

[Глава 2.1: Структура практической части 16](#_Toc137787652)

[Глава 2.2: Реализация ПО для UDP Flood атаки 17](#_Toc137787653)

[Глава 2.2.1: Руководство пользователя 18](#_Toc137787654)

[Глава 2.2.2: Тестирование приложения и UDP Flood атаки 21](#_Toc137787655)

[Глава 2.2.3: Техническая реализация 24](#_Toc137787656)

[Глава 2.3: Реализация ПО для идентификации атаки 27](#_Toc137787657)

[Глава 2.3.1: Руководство пользователя 28](#_Toc137787658)

[Глава 2.3.2: Тестирование приложение идентификации 31](#_Toc137787659)

[Глава 2.3.3: Техническая реализация 33](#_Toc137787660)

[Глава 2.4.1: Программа защиты от UDP Flood 35](#_Toc137787661)

[Глава 2.4.2: Руководство пользователя 36](#_Toc137787662)

[Глава 2.4.3: Тестирование приложения 37](#_Toc137787663)

[Глава 2.4.4: Техническая реализация 39](#_Toc137787664)

[ЗАКЛЮЧЕНИЕ 40](#_Toc137787665)

[СПИСОК ИСТОЧНИКОВ 41](#_Toc137787666)

# ВВЕДЕНИЕ

UDP Flood – это один из видов атак на компьютерные сети, который направлен на перегрузку сетевых ресурсов целевой системы. В отличие от других видов атак, таких как DDoS или SYN Flood, UDP Flood использует протокол UDP, который не обеспечивает никакой формы контроля над передачей данных и не требует установки соединения между отправителем и получателем. Это делает UDP Flood особенно эффективным для атак на сетевые устройства с ограниченными ресурсами, такие как маршрутизаторы или мосты.

В курсовой работе будут рассмотрены особенности и механизмы работы UDP Flood атак, методы их обнаружения и защиты, а также практические примеры использования данного типа атаки. Также будет рассмотрен вопрос о том, какие шаги могут быть предприняты для предотвращения возникновения UDP Flood атак и минимизации их воздействия на сетевую инфраструктуру.

Цель работы:

* Разработка на языке C# программного обеспечения для идентификации и защиты от UDP Flood, а также ПО для непосредственного проведения указанного типа атак в исследовательских целях.

Задачи:

1. Исследование протокола UDP, его достоинств и недостатков;
2. Исследование уязвимости протокола UDP к атакам UDP Flood;
3. Исследование методов идентификации и защиты от атак UDP Flood;
4. Разработка программного обеспечения для проведения атак UDP Flood и проведение атаки на приложение-жертву в исследовательских целях;
5. Разработка программного обеспечения для анализа интернет-трафика в целях идентификации UDP Flood атак в сети;
6. Разработка программного обеспечения для защиты от атак UDP Flood.

# Глава 1: Теоретическая часть

## Глава 1.1: Протокол UDP — достоинства и недостатки

UDP (User Datagram Protocol) — это протокол транспортного уровня, который обеспечивает передачу данных через компьютерные сети без установления соединения и гарантий доставки. UDP является простым и быстрым протоколом, который используется для передачи данных в реальном времени, таких как потоковое видео или голосовая связь.

Протокол UDP не обеспечивает никакой формы контроля над передачей данных и не гарантирует доставку пакетов в порядке их отправки. Каждый пакет UDP содержит только минимальную информацию, необходимую для его доставки, такую как адрес отправителя и получателя, порт отправителя и получателя, а также длину пакета.

Преимуществом протокола UDP является его скорость и простота, что делает его особенно полезным для приложений, где скорость передачи данных является более важной, чем гарантии доставки. Однако, UDP не подходит для приложений, где необходима надежность и гарантии доставки, таких как отправка писем или передача больших файлов.

Высокая скорость протокола UDP достигается за счет его простоты и отсутствия механизмов контроля передачи данных и обеспечения доставки. В отличие от протокола TCP, который обеспечивает надежную передачу данных, UDP не требует установления соединения и не выполняет проверку на ошибки передачи. Это позволяет уменьшить накладные расходы на установление соединения и контроль за передачей данных, что в свою очередь позволяет достигнуть более высокой скорости передачи данных.

Кроме того, протокол UDP использует простой заголовок, который содержит только минимальную информацию, необходимую для доставки пакета. Это также позволяет уменьшить накладные расходы на передачу данных и ускорить процесс обработки пакетов.

Заголовок протокола UDP состоит из 8 байт и содержит всю необходимую информацию для доставки пакета на конечный узел. Заголовок UDP содержит следующие поля:

1. Номер порта источника *(2 байта)* — номер порта отправителя пакета;
2. Номер порта назначения *(2 байта)* — номер порта получателя пакета.
3. Длина пакета *(2 байта)* — длина пакета, включая заголовок и данные.
4. Контрольная сумма *(2 байта)* — используется для проверки целостности пакета в процессе передачи.

Поле номера порта источника и номера порта назначения используется для идентификации отправителя и получателя пакета. Номер порта является 16-битным целым числом, которое определяет конкретный приложение, которое должно обрабатывать пакет.

Поле длины пакета определяет общую длину пакета в байтах, включая заголовок и данные. Это поле имеет размер 16 бит.

Поле контрольной суммы используется для проверки целостности пакета в процессе передачи. Контрольная сумма вычисляется на основе заголовка и данных пакета и пересылается вместе с пакетом. Получатель использует это значение для проверки целостности пакета и отбрасывает пакеты, которые были повреждены в процессе передачи.

Основными недостатками UDP являются отсутствие механизмов контроля передачи данных и обеспечения доставки. Это означает, что UDP не гарантирует доставку пакетов и не обеспечивает надежности передачи данных. При использовании UDP пакеты могут быть потеряны, дублированы или доставлены в неправильном порядке. Это может привести к ошибкам в работе приложений, использования неверных данных или потере данных.

Кроме того, UDP не поддерживает управление потоком и регулирование скорости передачи данных. Это может привести к перегрузке сети и ухудшению производительности приложений при передаче большого объема данных.

Также протокол UDP не поддерживает механизмов управления сетевой нагрузкой, что делает его уязвимым для атак на сеть, таких как UDP Flood. В результате большого количества UDP пакетов, отправляемых на целевую систему, сетевые ресурсы могут быть исчерпаны, что приводит к отказу в обслуживании легитимных пользователей.

По итогу мы имеем следующее отношение достоинств к недостаткам:

|  |  |
| --- | --- |
| ***Достоинства***   * Простота формирования UDP-пакета и его заголовка; * Скорость передачи за счёт отсутствия каких-либо механизмов подтверждения доставки пакета. | ***Недостатки***   * Отсутствие механизмов контроля доставки пакета; * Отсутствие механизмов регулирования очерёдности доставки пакетов; * Отсутствие механизмов регулирования скорости передачи данных и защиты от перегрузки сети. |

Простота и скорость передачи данных UDP являются значимыми плюсами протокола, но его недостатки делают его менее подходящим для приложений, где надежность передачи данных является критически важной, таких как отправка писем или передача больших файлов. Однако, UDP широко используется в реальном времени приложениях, таких как потоковое видео или голосовая связь, где скорость передачи данных является более важной, чем гарантии доставки.

## Глава 1.2: Уязвимости протокола UDP

Протокол UDP имеет несколько уязвимостей, которые могут быть использованы злоумышленниками для атак на сети и устройства. Далее будут приведены некоторые из них:

* Атаки на сетевую нагрузку: Протокол UDP не имеет механизмов контроля нагрузки, что делает его уязвимым для атак на сетевую нагрузку, таких как UDP Flood. В результате большого количества UDP пакетов, отправляемых на целевую систему, сетевые ресурсы могут быть исчерпаны, что приводит к отказу в обслуживании легитимных пользователей.
* Отправка неверных данных: Протокол UDP не обеспечивает надежность передачи данных, что означает, что пакеты могут быть повреждены или изменены в процессе передачи. Злоумышленники могут использовать это для отправки неверных данных на целевую систему.
* Подмена адреса источника: Протокол UDP не имеет механизмов аутентификации и проверки подлинности отправителя пакета, что делает его уязвимым для атак на подмену адреса источника. Злоумышленники могут подделывать адрес источника пакета, что может привести к ошибкам в работе приложений или использованию неверных данных.
* Сканирование портов: Протокол UDP используется для передачи данных между приложениями на разных узлах сети по определенным номерам портов. Злоумышленники могут использовать сканирование портов для обнаружения уязвимостей в приложениях на узлах сети.

В целом, протокол UDP является уязвимым для различных атак на компьютерные сети, и его использование должно быть ограничено только в тех случаях, когда скорость передачи данных является критически важной, а надежность передачи данных не является основным требованием.

## Глава 1.3: Вид атак UDP Flood

Так как тема этого курсового проекта ставит акцент на UDP Flood, имеет смысл перед тем, как производить написание программного обеспечения, разобраться в том, что же этот вид атак из себя представляет.

UDP Flood - это один из наиболее распространенных видов атак на компьютерные сети, который используется для нарушения работы сетевых устройств, серверов и приложений, а также для воровства конфиденциальной информации. Атаки UDP Flood основаны на отправке большого количества пакетов UDP на целевую систему, что приводит к ее перегрузке и недоступности для легитимных пользователей.

Протокол UDP является простым и быстрым, но он не имеет механизмов контроля передачи данных и не гарантирует доставку пакетов, что делает его уязвимым для атак. Отправитель может легко сгенерировать и отправить большое количество UDP пакетов на целевую систему, причем каждый пакет будет обработан независимо от других. Это приводит к перегрузке сетевых ресурсов и недоступности сервисов, работающих на целевой системе.

Для защиты от атак UDP Flood используются различные методы, такие как фильтрация трафика, настройка брандмауэров, установка ограничений на количество пакетов, обработка только легитимных запросов и т.д. Также существует специальное программное обеспечение для обнаружения и предотвращения UDP Flood атак, которое может анализировать трафик и автоматически блокировать подозрительные запросы.

UDP Flood атака может производится двумя разными способами:

Перегрузка пропускного канала сети множественной отправкой пакетов в него. Такой способ атаки требует от злоумышленника иметь устройство для атаки значительной мощности, однако, в случае успеха, прохождение полезного трафика по сети становится полностью невозможным.

Атака конкретного устройства ограниченным количеством пакетов. Суть данного способа заключается в том, что злоумышленник, зная, какое программное обеспечение установлено на целевом устройстве жертвы, может подделать пакет UDP пакет, который ожидает программа жертвы от легитимного пользователя. Жертва примет пакет и попытается обработать, однако пакет сформирован так, что в программе жертвы, которая приняла пакет, произойдут ошибки, из-за чего она не сможет штатно функционировать до вмешательства системного администратора. Так же злоумышленник может отправить серию пакетов с разных IP-адресов, которые жертва примет, но из-за того, что пакетов пришло слишком много, буфер сокета переполнится, и на устройстве жертвы произойдут ошибки.

## Глава 1.4: Идентификация UDP Flood атаки

Чтобы идентифицировать UDP Flood атаку, можно использовать следующие признаки:

* Увеличение трафика на сетевом интерфейсе: Во время атаки увеличивается количество пакетов, отправляемых на целевую систему. Это приводит к увеличению трафика на сетевом интерфейсе устройства. Для идентификации атаки можно использовать инструменты мониторинга сетевой активности, которые отслеживают трафик на интерфейсе;
* Увеличение нагрузки на процессор и память: При получении большого количества UDP пакетов целевая система должна обрабатывать каждый пакет, что приводит к увеличению нагрузки на процессор и память. Для идентификации атаки можно использовать инструменты мониторинга производительности, которые позволяют отслеживать нагрузку на систему;
* Отказ в обслуживании легитимных пользователей: При атаке целевая система может отказывать в обслуживании легитимных пользователей из-за перегрузки ресурсов. Если обнаружены отказы в обслуживании, которые не могут быть объяснены другими причинами, то это может быть признаком UDP Flood атаки;
* Ошибки в работе приложений: UDP Flood атака может привести к ошибкам в работе приложений, которые используют UDP для передачи данных. Если обнаружены ошибки в работе приложений, которые не могут быть объяснены другими причинами, то это может быть признаком UDP Flood атаки;
* Подозрительные UDP пакеты в сети: ПО для анализа UDP трафика в сети может быть настроено так, что при обнаружении пакетов с определённым или одинаковым содержимым, отправленным с разных IP-адресов, сообщит системного администратору о том, что определённое устройство могло стать жертвой UDP Flood атаки.

## Глава 1.5: Защита от UDP Flood атаки

Существует несколько методов защиты от UDP Flood атаки, которые можно использовать для предотвращения перегрузки ресурсов целевой системы. Далее приведены некоторые из них:

* Фильтрация трафика: Один из наиболее эффективных методов защиты от UDP Flood атаки - это фильтрация трафика на сетевом оборудовании. Это может быть реализовано с помощью настройки брандмауэра или использования специализированных устройств для обнаружения и блокировки атак. Фильтрация трафика позволяет отсеивать подозрительные пакеты, не позволяя им достигать целевой системы;
* Ограничение скорости: Ограничение скорости передачи UDP пакетов на целевой системе может ограничить количество пакетов, которые могут достигнуть системы за определенное время. Это может быть реализовано на сетевом оборудовании или на уровне приложения. Ограничение скорости может помочь предотвратить перегрузку ресурсов целевой системы.
* Использование кэша: Кэширование UDP пакетов на сетевом оборудовании может помочь предотвратить повторную передачу пакетов, что может привести к перегрузке ресурсов. Кэширование позволяет сохранять копии пакетов на сетевом оборудовании и использовать их вместо повторной передачи.
* Использование CDN: Использование Content Delivery Network (CDN) может помочь предотвратить UDP Flood атаку, поскольку CDN распределяет трафик между несколькими серверами, что позволяет обработать большой объем трафика без перегрузки ресурсов одного сервера.

В целом, защита от UDP Flood атаки может быть достигнута путем комбинации различных методов, таких как фильтрация трафика, ограничение скорости, кэширование и использование CDN.

# Глава 2: Практическая часть

## Глава 2.1: Структура практической части

Так как целями работы было определено написание трёх различных программ, выполняющих атаку, идентификацию и защиту от UDP Flood, в практической части проекта приведено три независимых друг от друга главы, в каждой из которой будет подробно описана своя конкретная программа. Такое разделение оправдано тем, что каждая программа независима по своей сути, выполняет определённую функцию, требует определённого подхода и описания собственного руководство пользователя.

## Глава 2.2: Реализация ПО для UDP Flood атаки

Программа для выполнения атаки UDP Flood была мною написана на языке программирования C# на основе библиотеки PcapDotNet, которая позволяет работать с сетевой картой на достаточно низком уровне, что необходимо для формирования и отправления пакетов в сеть.

В программе реализовано несколько режимов для выполнения атаки:

1. Режим для перегрузки сети пакетами. Этот режим позволяет запускать максимально возможное для устройства, с которого выполняется атака, количество пакетов на определённый адрес.  
   Этот режим необходим для перегрузки сети и полного прекращения движения какого-либо трафика в ней, но он довольно требователен к характеристикам атакующего устройства, а также во время атаки такого типа невозможно использовать интернет и на атакующем устройстве;
2. Режим задержки позволяет настроить все поля необходимые поля UDP-пакета и выставить задержку между отправкой пакета. Этот режим необходим, например, для того, чтобы раз в сутки отправлять один пакет на адрес жертвы, которая имеет несовершенное ПО. То есть подобный способ атаки позволит раз в сутки вызывать ошибки у жертвы до тех пор, пока уязвимость ПО не будет устранена.

Так же мною сразу была заложена возможность многопоточного выполнения атаки. Эта возможность позволяет одновременно атаковать несколько разных устройств независимо друг от друга. Каждый поток может иметь свои настройки для формирования пакетов, которые будут отправлять в заданное ему время.

Дополнительной функцией является постройка ARP-таблицы сети, в которой находится устройство. Эта функция необходима для выполнения атаки без использования дополнительного ПО для анализа трафика и выявления устройств в сети.

### Глава 2.2.1: Руководство пользователя

При запуске программы атаки, пользователя встречает окно (рис. 1), в котором он может увидеть существующие на данный момент потоки, выбрать интерфейс, с которого будет происходить атака, перейти в меню построения ARP-таблицы, либо создать новый поток.

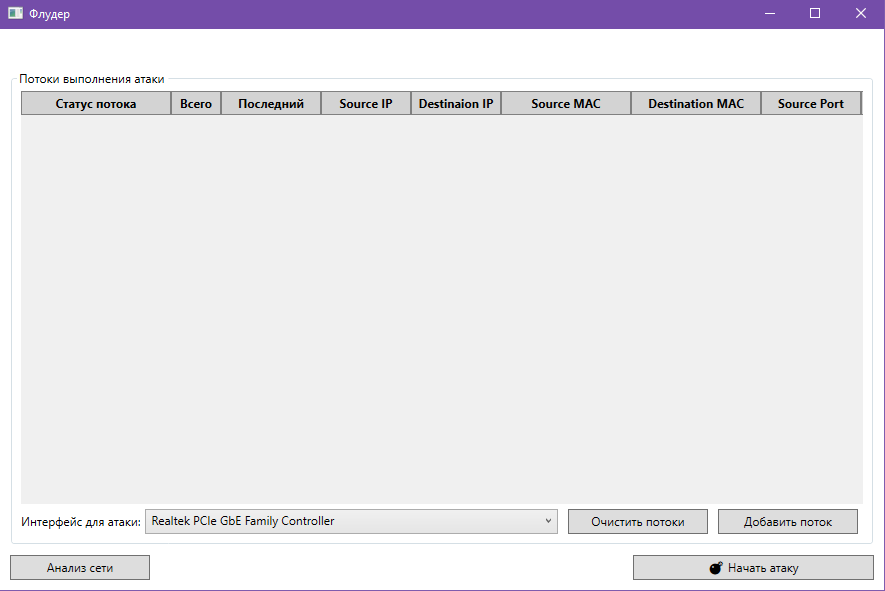


Рисунок 1 – Главное окно программы

В меню построения ARP-таблицы (рис. 2) пользователь может выбрать нужную сетевую карту и запустить анализ сети и увидеть другие устройства, находящиеся в его локальной сети.

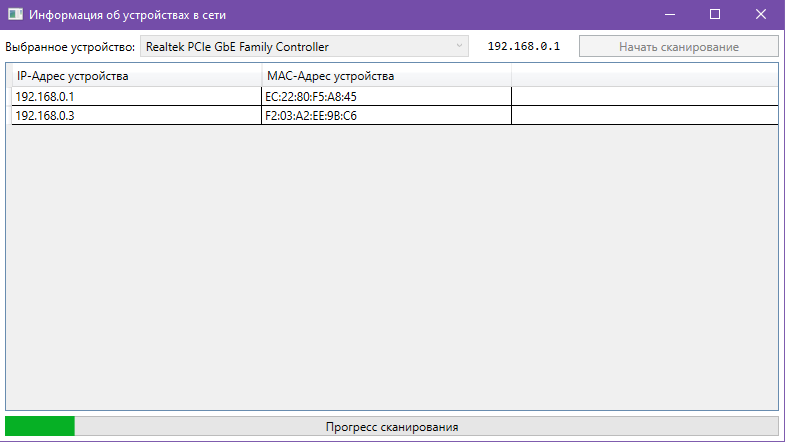


Рисунок 2 - Окно анализа сети в процессе сканирования

Если пользователь нажмём на создание нового потока, он увидит следующее окно (рис. 3), на котором он сможет настроить задержку отправки пакетов, установить параметры пакетов и выбрать режим, в котором могут использоваться параметры.

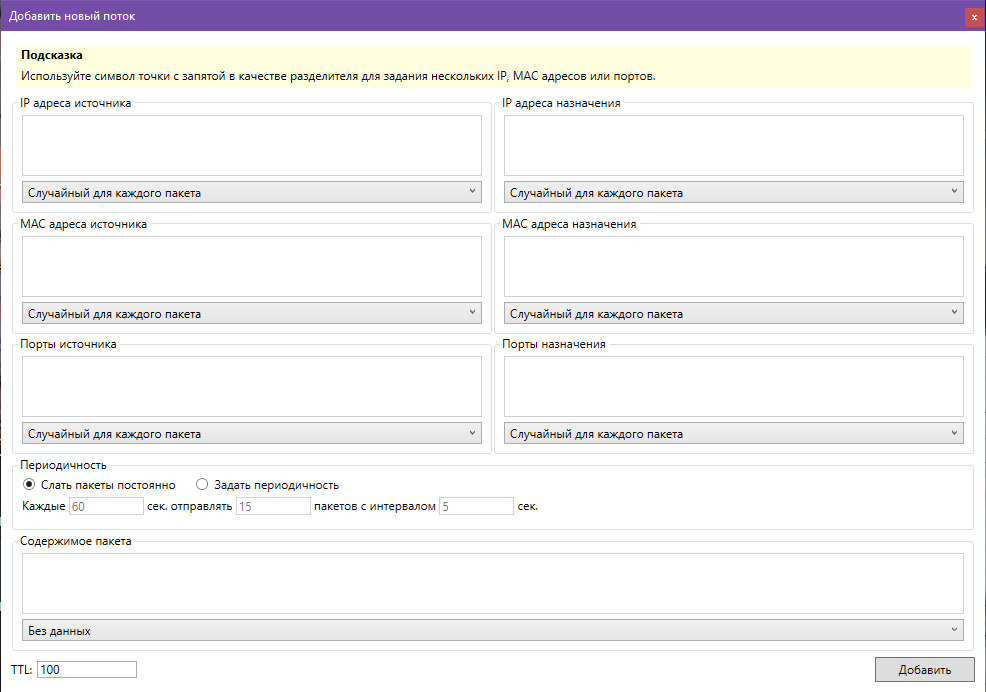


Рисунок 3 - Окно добавления нового потока

После добавления нового потока пользователь сможет начать атаку. После старта атаки, главное окно программы изменится (рис. 4) и там будет указано, сколько всего отправлено пакетов и когда был отправлен последний пакет.

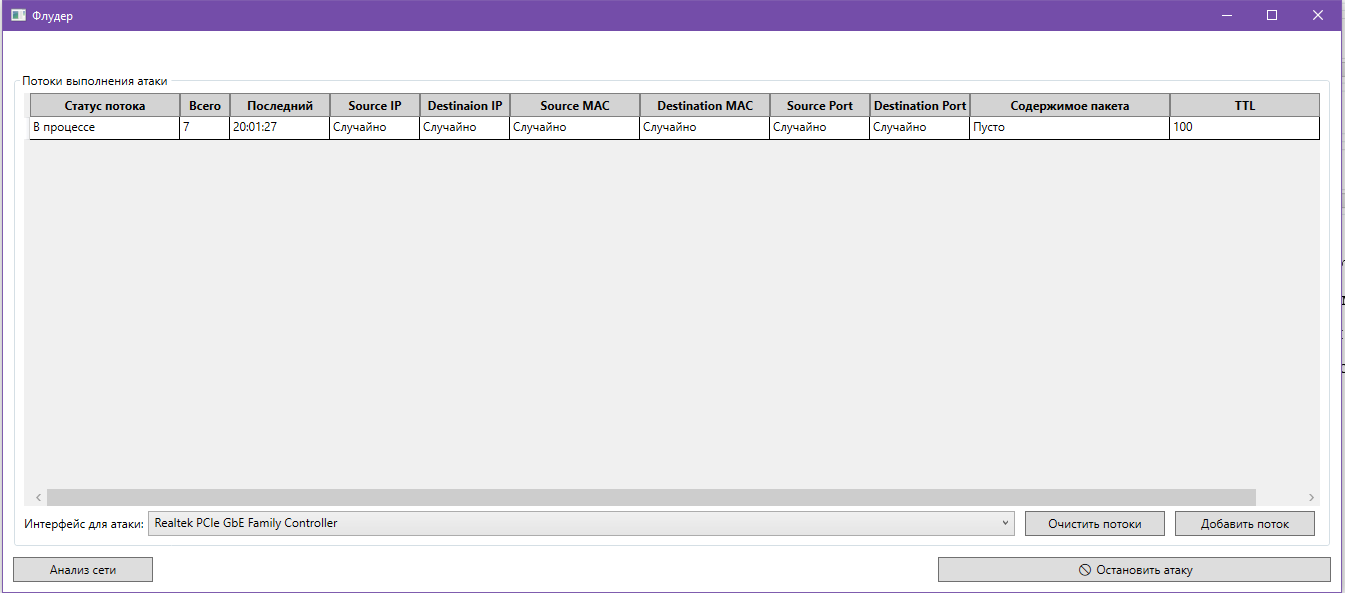


Рисунок 4 – Главное окно программы в момент выполнения атаки

### Глава 2.2.2: Тестирование приложения и UDP Flood атаки

Для тестирования UDP Flood атаки мной был написан простой TCP/UDP чат (рис. 5), поддерживающий соединение с пользователями. Я исполню роль злоумышленника и попробую нарушить общение в чате с помощью атаки на него через написанную мной программу UDP Flood атаки.

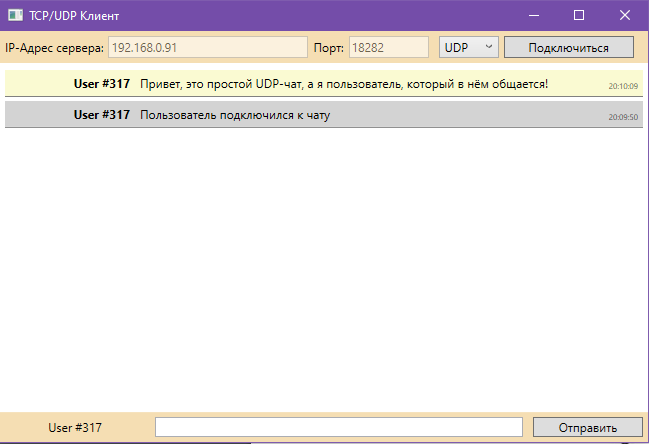


Рисунок 5 - Клиентское приложение UDP чата

Мне, благодаря анализу трафика в сети (рис. 6, рис. 7), стали известны открытые технические данные работы чата.

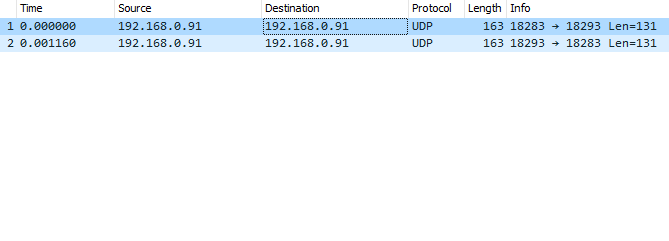


Рисунок 6 – Перехват трафика чата

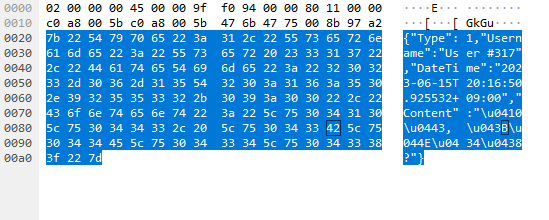


Рисунок 7 – Содержимое пакета чата

Благодаря тому, что я перехватил трафик чата, теперь я знаю следующее:

1. Сервер чата работает на IP-адресе 192.168.0.91;
2. Сервер чата прослушивает порт 18293;
3. Данные UDP пакета представляют собой закодированную UTF-8 строку, которая представляет собой JSON-объект и содержит в своём теле поля «Type», «Username», «Content» и «DateTime».

Благодаря всей этой информации я смогу подделать пакет и нарушить общение в чате поддельными сообщениями. Для этого я создаю новый поток в своём приложении для выполнения UDP Flood атаки. Настройки, которые я внесу в поток атаки приведены на рис. 8.

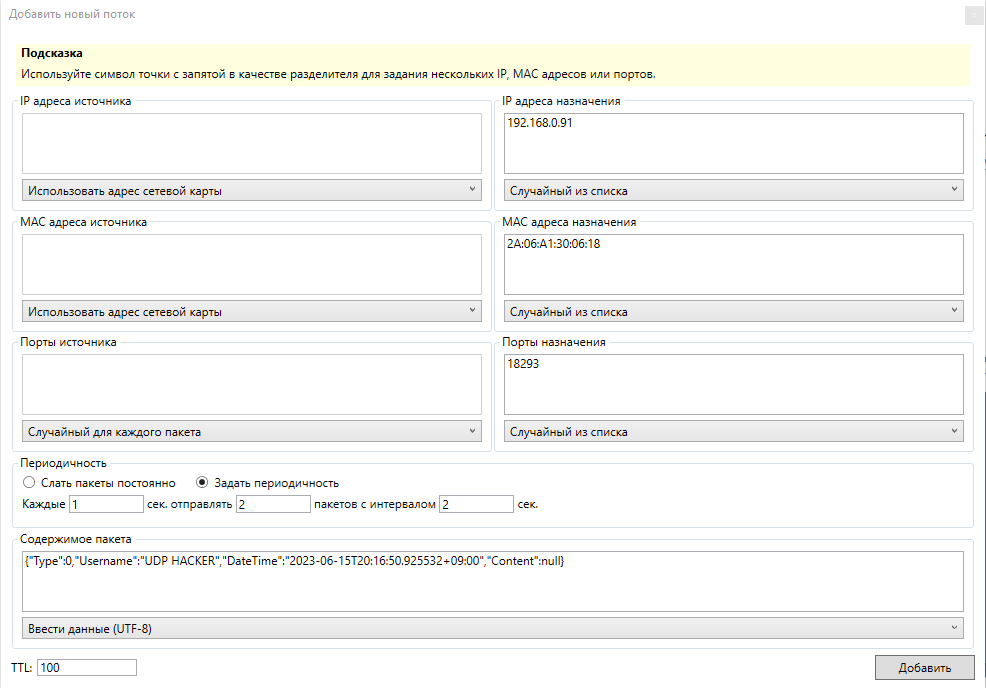


Рисунок 8 – Настройки для атаки на чат

Теперь, когда все настройки внесены, программа готова к тому, чтобы атаковать UDP-чат. После запуска атаки становится видно, что чат становится невозможно использовать из-за множественных сообщений о подключении пользователя к чату (рис. 9), а при запуске ПО для сканирования сети, видно огромное количество UDP-пакетов, адресованных серверу чата (рис. 10).

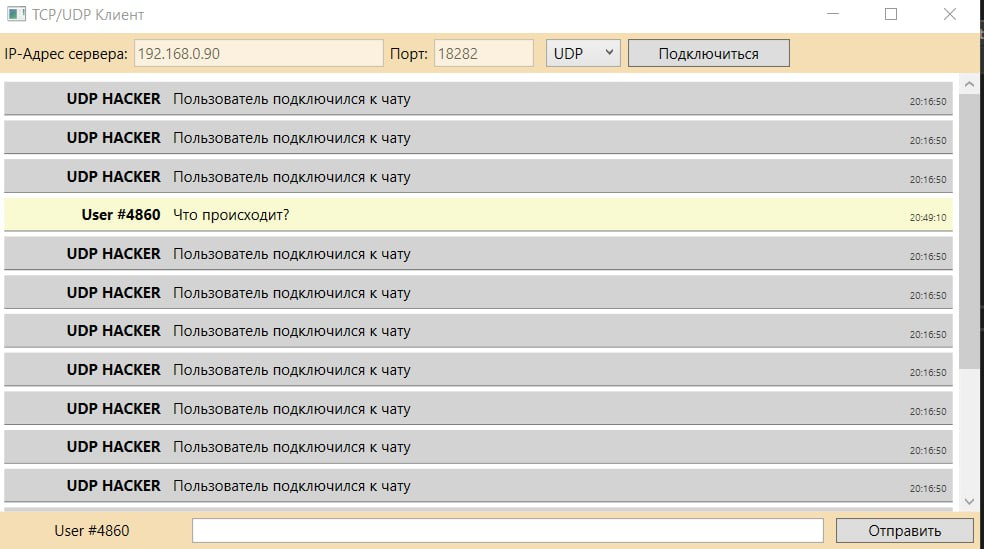


Рисунок 9 – Происходящее в чате в момент атаки

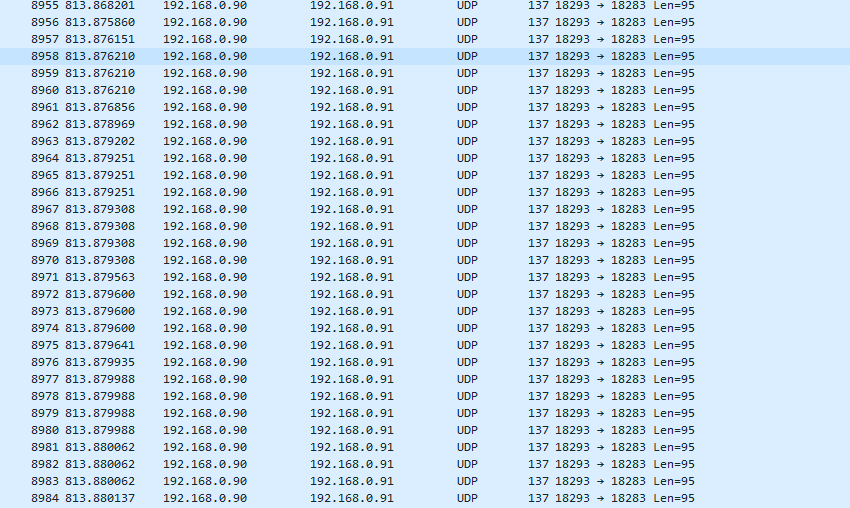


Рисунок 10 – Состояние сети во время атаки

Таким образом, мы атаковали простой чат с помощью метода UDP Flood и написанного для этого программного обеспечения. Можно сделать вывод, что программное обеспечение для атаки работает исправно и на его основе можно производить UDP Flood атаки.

### Глава 2.2.3: Техническая реализация

Так как приложение программы атаки выполнено таким образом, что её часть, непосредственно отвечающая за проведение атаки может быть подключена к любому пользовательскому интерфейсу (включая, консольный). В этой главе описана программная реализация той части программы, которая отвечает за непосредственное взаимодействие с сетевой картой.

Как видно на схеме (рис. 11) основной частью программы является класс «UdpBomber». Этот класс отвечает за управление потоками, которые выполняют атаку и сетевыми интерфейсами, между которыми может переключаться пользователь.

Класс «UdpBomber» должен создаваться единственном экземпляре на всю программу, так как по сути он и отвечает за выполнение атаки.

Композиционно связан с классом «UdpBomber» класс «Scanner», который отвечает за сканирование сети для построения ARP-таблицы. Класс «Scanner» так же всегда существует в единственном экземпляре, который инициирует «UdpBomber». «Scanner» не должен создаваться пользователем никогда.

Так как приложение выполнения атаки поддерживает многопоточность, непосредственно за атаку отвечает класс «UdpBomberThread», который создаётся классом «UdpBomber» при помощи вспомогательного класса «BomberThreadSettings». Вспомогательный класс позволяет передать от пользовательского интерфейса к программной части настройки, установленные пользователем для создаваемого потока. Вспомогательный класс настроек содержит данные о MAC и IP адресах источника и назначения.

Кроме того, во вспомогательном классе содержится информация о необходимой задержке в отправке пакетов, содержимом пакетов, портах назначения и отправки и некоторая другая настроечная информация.

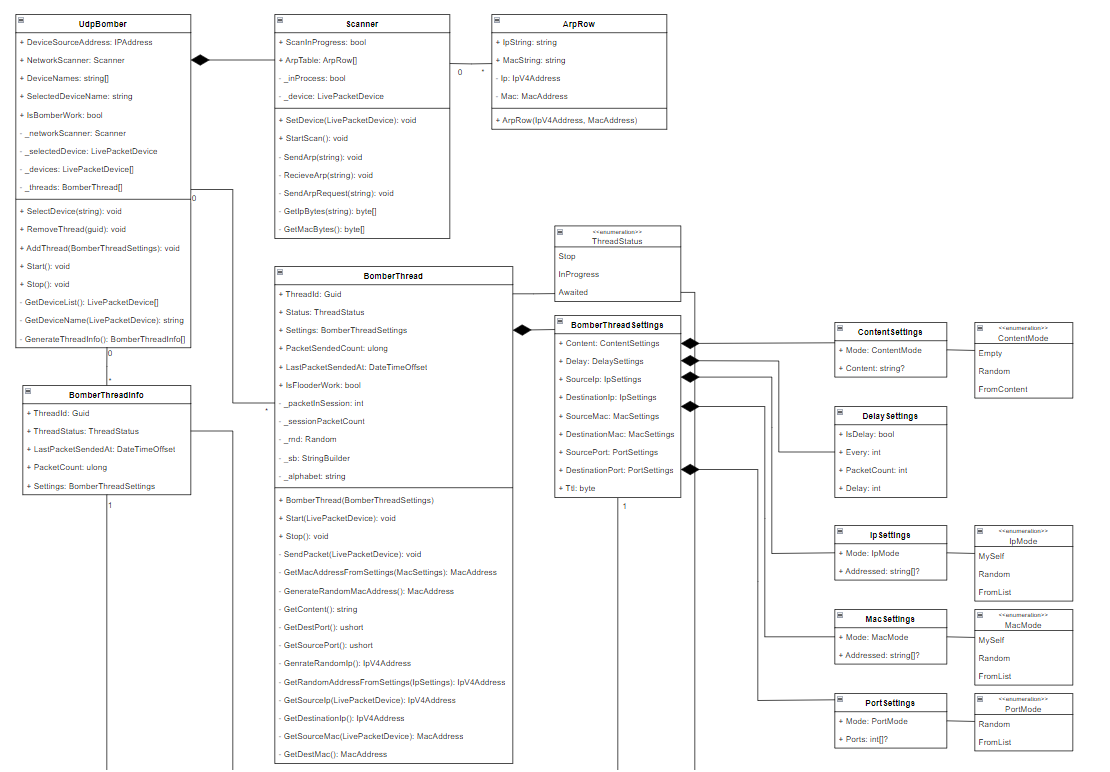


Рисунок 11 – Структура атакующей части программы UDP Flood

Когда пользователем будет добавлено необходимое количество потоков, он может запустить атаку, вызвав метод «Start» у класса «UdpBomber», который в свою очередь запустит каждый из добавленных пользователем потоков.

Для обратной связи программы атаки и пользовательского интерфейса применяется дополнительный вспомогательный класс «BomberThreadInfo». В нём содержится информации о текущем статусе потока (шлёт пакеты, ожидает установленную задержку, не активен), количестве отправленных пакетов, дате и времени последнего отправленного пакета, и настройках потока. Когда в статусе потока происходят какие-либо изменения, он вызывает установленную пользователем функцию методом обратного вызова вверх по цепочке.

## Глава 2.3: Реализация ПО для идентификации атаки

Среди нескольких разных вариантов идентификации UDP Flood атаки, описанных в главе 1.4 мною для написания программного обеспечения был выбран метод определения атаки по увеличению потока трафика на сетевом интерфейсе.

Программа-идентификатор непрерывно прослушает весь проходящий в сети трафик, подсчитывая общее количество UDP пакетов в сети. Программа рассчитывает количество пакетов в минуту приходящее на каждый IP-адрес, который был замечен и количество пакетов в процентах от общего их количества.

Далее программа выбирает пять адресов, на которые пришло наибольшее количество запросов и демонстрирует их пользователю, дополнительно предоставляя ему срез трафика в сети за последние 30 секунд.

Пользователь, отталкиваясь от данных о загруженности сети и информации о последних полученных пакетах может сделать выводы о том, происходит ли в настоящее время атака или сеть находится в штатном состоянии.

### Глава 2.3.1: Руководство пользователя

Сразу после запуска программы идентификации пользователь увидит окно приложения (рис. 12), на котором можно выбрать интерфейс, с которого будет осуществляться сбор трафика, и запустить сбор информации с помощью соответствующей кнопки.

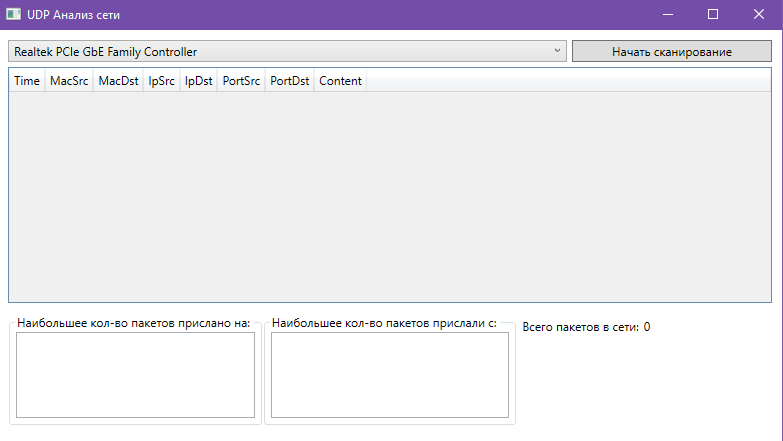


Рисунок 12 – Окно программы идентификации

Так же в окне программы можно увидеть разделы, где будут выведены адреса, на которые пришло больше всего пакетов, адреса, с которых пришло наибольшее количество адресов и общее количество пакетов в сети за последнюю минуту. Так же программа имеет окно, где будет представлен срез трафика за последние 30 секунд.

Сразу после запуска сканирования трафика окно программы никак не изменится, но через какое-то время программа соберёт первые данные и представит пользователю срез трафика (рис. 13).

Уже сейчас можно увидеть некоторые данные, например, кто прислал пакет и куда его прислал, а также его содержимое. Однако, срез трафика будет потом обновляться. Чтобы этого не происходило, необходимо сперва остановить сканирование, и только потом изучать трафик.

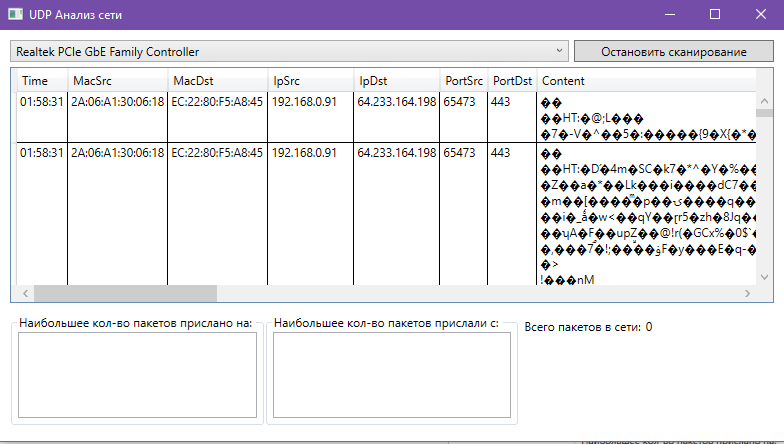


Рисунок 13 – Программа идентификации после сбора первых данных

Спустя минуту после запуска программы, она соберёт больше данных и представит их пользователю (рис. 14). Теперь пользователь может видеть, на какие адреса было прислано больше всего пакетов (в скобках – количество, а через дефис представлено процентное отношение к общему количеству пакетов), общее количество UDP-пакетов в сети.

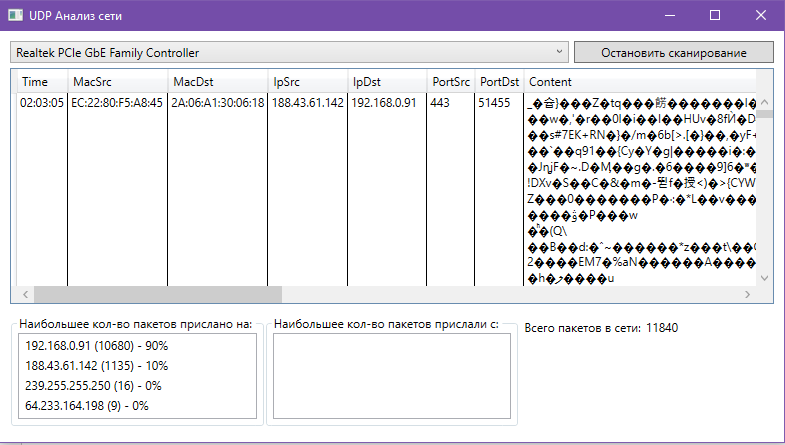


Рисунок 14 – Программа идентификации после сбора данных

Так же пользователь может нажать на какой-то из интересующих его адресов увидеть, кто прислал наибольшее количество пакетов (рис. 15).

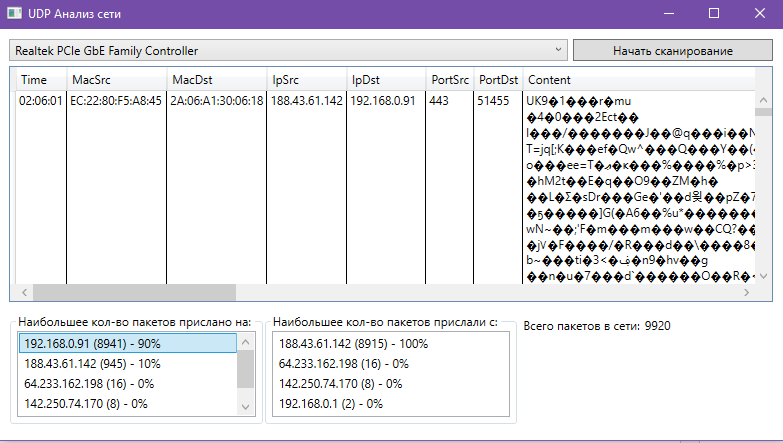


Рисунок 15 – Изучение количества отправленных пакетов на определённый адрес

Благодаря этим данным, пользователь может определить IP-адрес злоумышленника и заблокировать его в сети. Либо, если пакеты отправляются с разных IP-адресов, пользователь может обнаружить это по зашкаливающему счётчику пакетов в сети и процентному соотношению отправителей UDP-пакетов, где IP злоумышленника в процентном соотношении будет значительно превышать легитимных пользователей.

Данные о количестве пакетов в сети обновляются каждую минуту.

### Глава 2.3.2: Тестирование приложение идентификации

Для тестирования приложения, вернёмся к ситуации, описанной в главе 2.2.2, где злоумышленник атаковал чат, тем самым мешая пользователям общаться в нём.

Теперь мы в лице администратора чата запустим написанное нами приложение идентификации UDP Flood и попробуем отследить IP-адрес злоумышленника.

Сразу после запуска идентификатора, даже то того, как он соберёт данные, мы уже видим чрезвычайно подозрительную активность в сети (рис.16). Огромной количество одинаковых пакетов, атакующих машину-сервер чата.

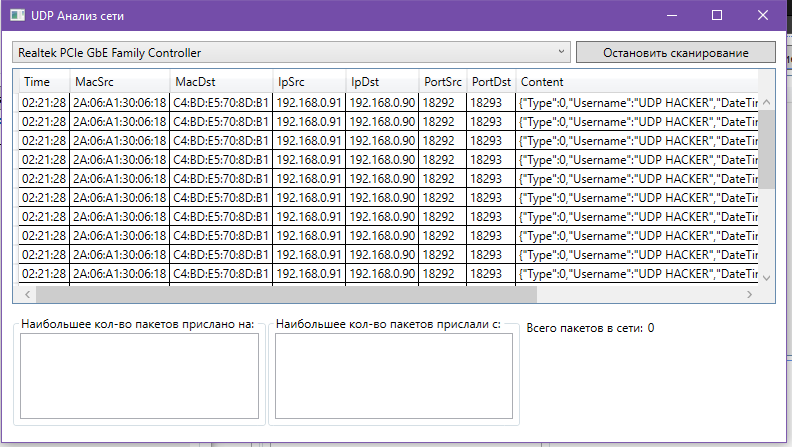


Рисунок 16 – Поведение идентификатора сразу после запуска

После ожидания того, как идентификатор соберёт данные, мы можем наблюдать следующую картину (рис. 17). Устройство, которое имеет IP-Адрес 192.168.0.91 прислало на машину, где установлен сервер чата, более 800 запросов за минуту. Можем сделать вывод, что это злоумышленник.

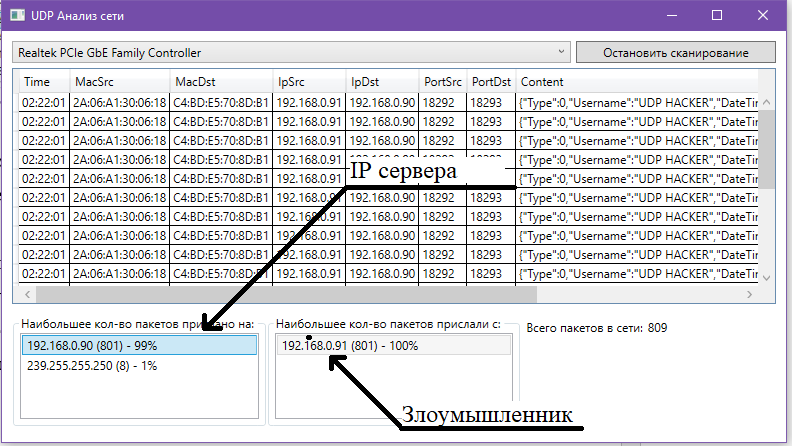


Рисунок 17 – Анализ данных, собранных идентификатором

Таким образом, мы протестировали идентификатор UDP Flood, и благодаря ему обнаружили злоумышленника, который атаковал наш чат и мешал пользователям в нём наслаждаться общением.

Можем сделать вывод, что программа-идентификатор работает корректно и выполняет те функции, которые от неё требуются.

### Глава 2.3.3: Техническая реализация

Так же, как и в главе 2.2.3 имеет смысл разбирать только ту часть программы, которая отвечает за непосредственную работу с сетью и сетевой картой.

Здесь главным классом выступает «UdpAnalyzer», который позволяет сканировать проходящий через сетевую карту трафик и сохранять. Здесь так же, как и в случае с программой для атаки применяется многопоточность.

Дополнительный поток используется для сканирования трафика без блокирования пользовательского интерфейса. Программной реализацией дополнительного потока выступает класс «UdpAnalyzerThread». В случае, если этот объект обнаруживает в сети UDP-пакет, он формирует «UdpPacket» объект и передаёт его методом обратного вызова вверх по цепочке.

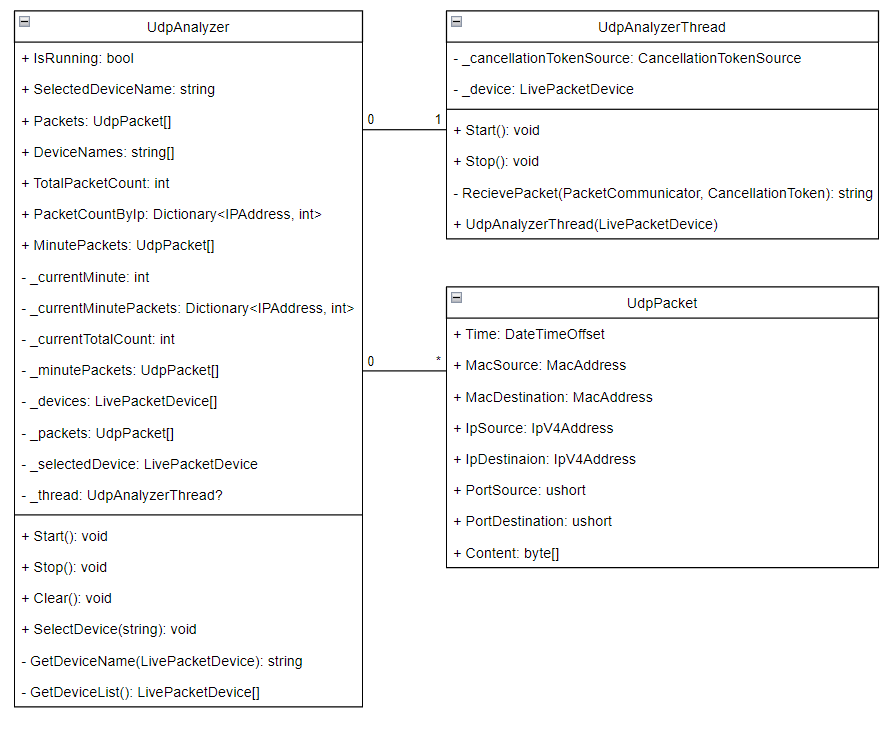


Рисунок 18 – UML идентификатора UDP Flood

У «UdpAnalyzer» есть поля для сбора ежеминутных данных, такие как «MinutePackets», «PacketCountByIp» и некоторые другие. Эти поля позволяют идентификатору собирать данные поминутно, постоянно обнуляя их.

Но есть и поля для постоянного накопления пакетов, такие как «Packet» и «TotalPacketsCount».

Общая архитектура взаимодействия пользовательского интерфейса и части ответственной за взаимодействие реализовано с помощью паттерна MVVM (Model-View-ViewModel), так как для написания пользовательского интерфейса применялась технология WPF.

Сама же программа выполнена на языке C# с использованием библиотеки PcapDotNet.

## Глава 2.4.1: Программа защиты от UDP Flood

Среди множества способов защиты от атаки вида UDP Flood, описанные в главе 1.5, я решил реализовать метод защиты через установку и настройку правил брандмауэра Windows.

Для этого мною была написана программа, которая позволяет заблокировать необходимый IP-адрес для приёма от него UDP-пакетов. То есть, мы можем установить такие правила, при которых устройство находящиеся на определённом IP-адресе будет слать пакеты на наше устройство, но мы будем эти пакеты просто отбрасывать и не принимать во внимание.

Я изначально решил, что программа будет работать только с правилами для UDP-соединений, так как этот протокол является темой курсового проекта.

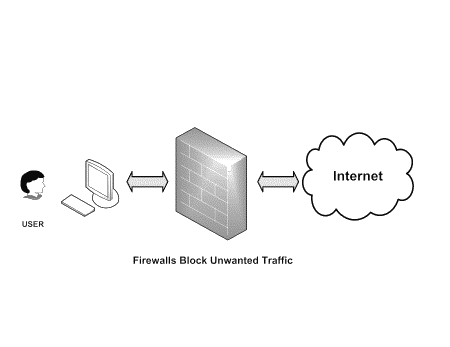


Рисунок 19 – Схема работы Firewall

### Глава 2.4.2: Руководство пользователя

После открытия приложения, которое должно защитить компьютер от атаки UDP Flood, пользователь видит окно (рис. 20), на котором имеются следующие элементы: список существующих правил, блокирующих входящие пакеты по протоколу UDP, поля ввода нового правила, кнопки блокировки и обновления списка правил.

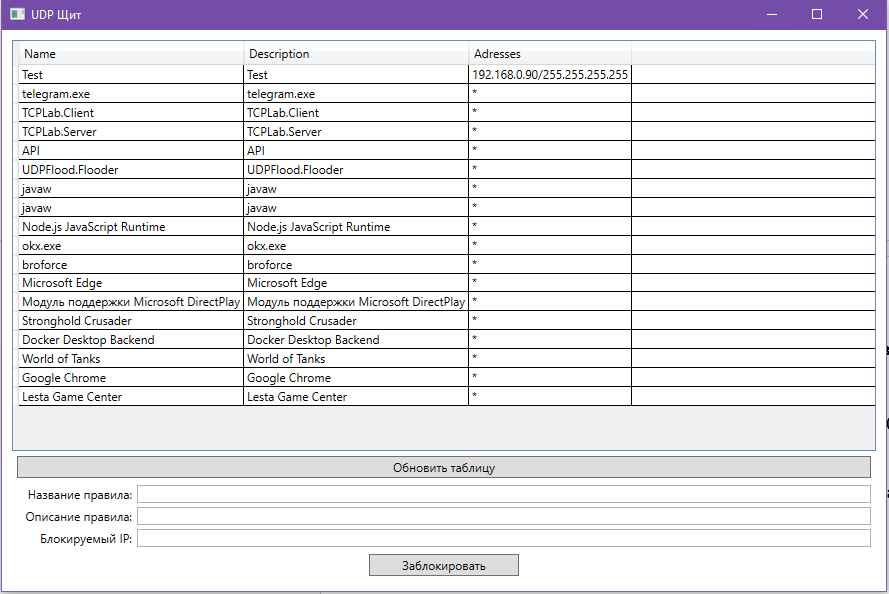


Рисунок 20 – Окно программы защиты UDP трафика

Пользователь может ввести новое правило и нажать кнопку блокировки. Если какие-то данные были введены неверно, пользователь получит сообщение об ошибке (рис. 21), иначе же новое правило добавится в список и будущие пакеты, полученные от этого IP-адреса будут отброшены.

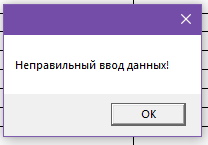


Рисунок 21 – Сообщение об ошибке при добавлении правила

### Глава 2.4.3: Тестирование приложения

Итак, вновь вернёмся к ситуации, актуальной на конец главы 2.2.3, в которой на чат происходила UDP Flood атака. В главе 2.3.3 нам удалось вычислить IP-адрес злоумышленника, и теперь с помощью программы защиты мы можем его заблокировать.

Для блокировки IP-адреса злоумышленника введём его данные в программу (рис. 21) и нажмём кнопку заблокировать.

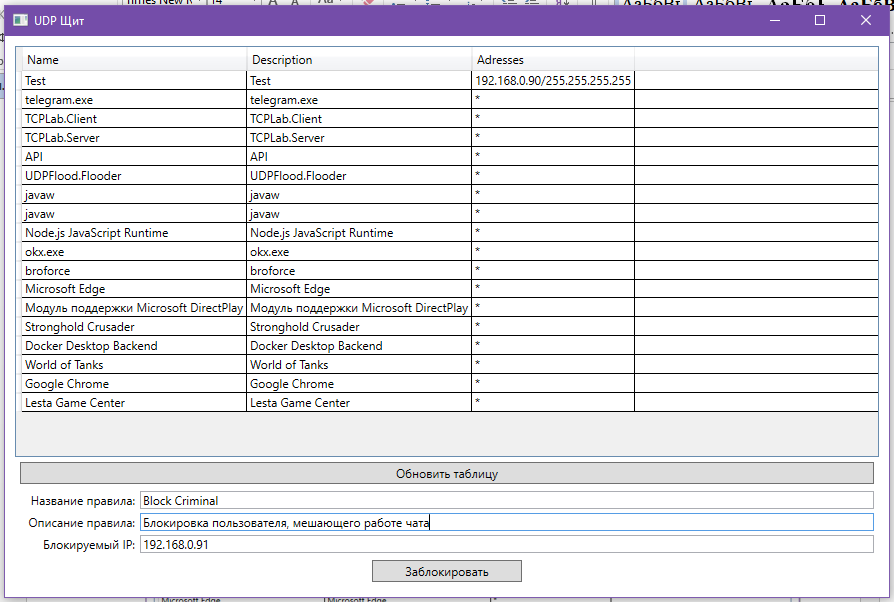


Рисунок 21 – Блокировка пользователя

После нажатия кнопки «Заблокировать» новое правило появилось в списке существующих заданных правил брандмауэра (рис. 22) для входящих подключений по протоколу UDP.

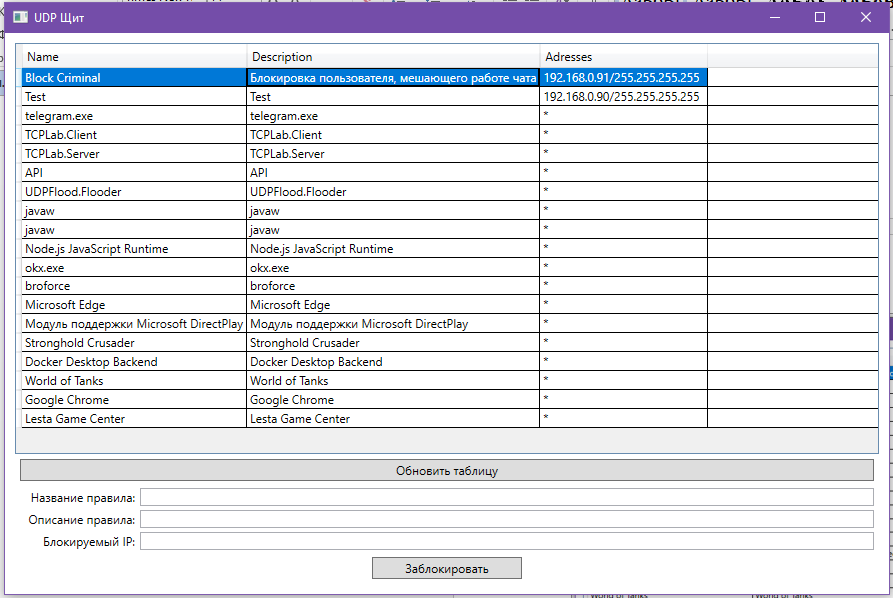


Рисунок 22 – Обновленная таблица правил после блокировки пользователя

После блокировки осталось проверить ситуацию в чате. Как можно заметить, (рис. 23) спам сообщений прекратился, и пользователи могут продолжить общение в чате, соответственно, было заблокировано верное устройство.

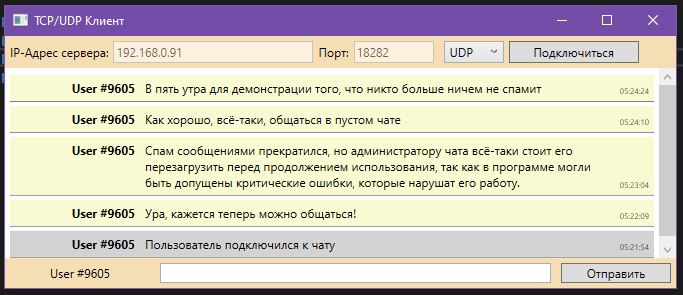


Рисунок 23 – Обстановка в чате после блокировки злоумышленника

### Глава 2.4.4: Техническая реализация

Программная реализация программы защиты от UDP Flood состоит во взаимодействии со встроенной библиотекой Windows для взаимодействия с API брандмауэра Windows.

Необходимая нам библиотека во всех версиях Windows, начиная с Windows 7, находится по пути C:\Windows\System32\FirewallAPI.dll.

После подключения указанной библиотеки, мы можем обратиться к брандмауэру Windows, например, для чтения существующих правил firewall следующим образом (рис. 24):

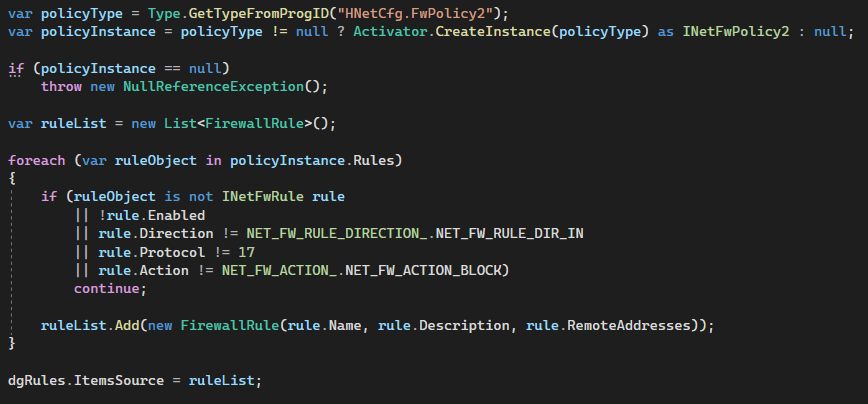


Рисунок 24 – Чтение существующих правил Firewall

Код операции чтения приведён здесь, потому что сейчас очень сложно найти документацию по взаимодействию с брандмауэром Windows средствами языка программирования C# и указанный отрезок кода является результатом суточного анализа библиотек и различных исходников.

Мною был выработан следующих алгоритм для создания правила:

1. Получение службы брандмауэра и конкретной политики, активной в данный момент;
2. Создание нового правила брандмауэра путём взаимодействия с полученной ранее службой;
3. Изменения правила фильтрации трафика под собственные нужны;
4. Внесение правила в общий список и его применение.

# ЗАКЛЮЧЕНИЕ

UDP Flood является одним из наиболее распространенных видов атак, который основывается на перегрузке сети большим количеством UDP-пакетов. Такая атака может привести к серьезным проблемам для целевой системы, включая перегрузку сетевых устройств и отказ в обслуживании.

В курсовой работе были просмотрены диаметрально разные позиции работы с UDP Flood, была занята позиция потенциального хакера, который производит вмешательство в приложение сетевого чата, а во время выполнения атаки была занята роль специалиста по безопасности, который пытается противостоять хакеру, вычислить его и заблокировать.

Благодаря такому подходу, удалось подробно разобрать тип атак UDP-Flood, а разработанное в ходе курсовой работы программное обеспечение, позволяет на необходимом уровне проводить как атаку, так и выполнять меры по идентификации и защите от атак такого уровня.

# СПИСОК ИСТОЧНИКОВ

1. «Википедия – свободная энциклопедия» UDP-флуд [Электронный ресурс] / Режим доступа: https://ru.wikipedia.org/wiki/UDP-флуд (Дата обращения: 15.05.2023)
2. «GitHub» Pcap.Net - the open-source, .NET wrapper for WinPcap [Электронный ресурс] / Режим доступа: https://github.com/PcapDotNet/Pcap.Net/wiki (Дата обращения: 02.06.2023)
3. «TechExpert TIPS» Windows Firewall - Блокировать список IP-адресов [Электронный ресурс] / Режим доступа: https://techexpert.tips/ru/windows-ru/windows-firewall-блокировать-список-ip-адресов / (Дата обращения: 03.06.2023)
4. «TechExpert TIPS» Windows Firewall - Блокировать список IP-адресов [Электронный ресурс] / Режим доступа: https://techexpert.tips/ru/windows-ru/windows-firewall-блокировать-список-ip-адресов / (Дата обращения: 03.06.2023)
5. «Damien Doumer» Allow Your App through the Firewall With C# [Электронный ресурс] / Режим доступа: https://doumer.me/allow-your-app-through-the-firewall-with-c/ / (Дата обращения: 04.06.2023)
6. Оформление курсовой работы [Электронный ресурс] / Общие требования к построению и оформлению текстовой документации ЗабГУ. – Режим доступа: https://zabgu.ru/files/html\_document/pdf\_files/fixed/Normativny'e\_dokumenty'/MI\_\_01-03-2023\_Obshhie\_trebovaniya\_k\_postroeniyu\_i\_oformleniyu\_uchebnoj\_tekstovoj\_dokumentacii.pdf (Дата обращения: 15.06.2023)