Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

ОТЧЁТ
к лабораторной работе №4
на тему

РЕАЛИЗАЦИЯ СЕРВЕРА, ЗАЩИЩЕННОГО ОТ АТАКИ ПРИ УСТАНОВКЕ TCP-СОЕДИНЕНИЯ И В РАМКАХ ЗАДАННОГО ПРОТОКОЛА ПРИКЛАДНОГО УРОВНЯ

Выполнил: студент гр. 253503
Тимошевич К.С.

Проверил: ассистент кафедры информатики Герчик А.В.

Минск 2025

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ

В рамках данной лабораторной работы необходимо разработать серверное приложение, которое будет защищено от атак при установке *TCP*-соединений и обеспечит соответствие заданному протоколу прикладного уровня.

Главная цель работы заключается в освоении методов защиты сетевых сервисов от атак, таких как *SYN*-флуд, а также в изучении принципов обеспечения безопасности при обработке сетевых соединений и данных [1].

Необходимо реализовать *TCP*-сервер, который будет принимать соединения от клиентов, используя защиту от атак при установке соединения, включая контроль количества соединений от одного *IP*-адреса и тайм-ауты для обработки запросов. Сервер должен корректно обрабатывать входящие данные и правильно реагировать на ошибки, тайм-ауты соединений.

Одним из ключевых требований является защита от *SYN*-флуд-атак путем ограничения числа подключений с одного *IP*-адреса. В процессе работы необходимо провести тестирование на устойчивость к атакам, включая *SYN*-флуд и попытки создания множества соединений с одного *IP*. Результаты тестирования помогут оценить эффективность защитных механизмов, выявить уязвимости и решить, как дальше можно улучшить зашиту.

2 ХОД ВЫПОЛНЕНИЯ РАБОТЫ

В ходе выполнения лабораторной работы был разработан и протестирован *TCP*-сервер с двумя вариантами реализации: защищенным сервером и сервером без защиты от атак. Оба варианта сервера были реализованы на языке *Python* с использованием библиотеки *socket*, которая позволяет работать с низкоуровневыми сетевыми протоколами [2].

В первом варианте был реализован защищенный сервер с дополнительной функцией, предотвращающей атаки, направленные на переполнение соединений с одного *IP*-адреса. Сервер использует многопоточность, реализованную с помощью модуля *threading*, чтобы обрабатывать несколько клиентов одновременно. В процессе работы каждого клиента сервер создает отдельный поток, который обрабатывает его запросы, что позволяет не блокировать основное соединение и эффективно обслуживать несколько пользователей одновременно.

Для предотвращения атак с массовым подключением с одного *IP*-адреса была введена проверка на количество активных соединений с каждым *IP*. Если количество подключений превышает два, сервер отказывает в соединении и закрывает сокет клиента. Этот механизм защищает сервер от возможных атак типа *DoS (Denial of Service)*, таких как *SYN*-флуд.

Второй вариант – сервер без защиты – реализует базовый *TCP*-сервер, который просто принимает соединения и отправляет обратно полученные данные. В этой версии сервер не учитывает количество соединений с одного *IP*-адреса и не защищен от атак, что позволяет нагрузить сервер чрезмерным количеством соединений. В коде сервера используется также многопоточность, но без дополнительных механизмов безопасности.

В процессе тестирования было проведено моделирование атаки на сервер без защиты. Для этого был создан скрипт, имитирующий массовое подключение 100 клиентов к серверу с использованием многопоточности. Это позволило проверить устойчивость сервера к атакам типа *SYN*-флуд. Как и ожидалось, сервер без защиты не справился с большим количеством соединений, что показало необходимость реализации защиты в реальных проектах.

Защищенный сервер, в свою очередь, успешно обрабатывал только два соединения с одного *IP*-адреса, после чего новые попытки подключения с того же *IP* были отклонены. Это доказало эффективность добавленной защиты, а также улучшение стабильности работы сервера при высоких нагрузках.

Таким образом, в результате работы была реализована двухвариантная система *TCP*-серверов, одна из которых защищена от атак, а другая – нет. Это позволило на практике продемонстрировать, как защита на уровне приложения помогает избежать проблем с производительностью и безопасностью при большом числе подключений.

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы был разработан и реализован *TCP*-сервер на языке *Python*, использующий возможности сетевых функций. В ходе работы были реализованы два варианта сервера: защищенный и без защиты от атак. Основной целью работы было создание серверного приложения, способного эффективно обрабатывать множественные соединения и обеспечивать защиту от атак, таких как *SYN*-флуд.

Защищенная версия сервера продемонстрировала свою эффективность в управлении количеством подключений с одного *IP*-адреса, что предотвратило возможные атаки, направленные на перегрузку сервера. Также сервер реализует многопоточность, что позволяет обрабатывать несколько соединений одновременно без блокировки основного процесса. В сервере была предусмотрена обработка ошибок, включая тайм-ауты, что повысило его устойчивость в условиях реальных нагрузок.

Сервер без защиты, в свою очередь, стал наглядным примером уязвимости в случае атак, основанных на массовых соединениях с одного источника. Проведенное тестирование показало, что данный вариант сервера не справляется с большим количеством соединений, что подчеркивает важность защиты в реальных проектах.

Реализация защиты на уровне приложения, а также использование многопоточности, оказались ключевыми аспектами в обеспечении стабильной и безопасной работы сервера. Полученные результаты подтвердили, что правильный подход к проектированию сетевых приложений и использование механизмов защиты могут существенно повысить их устойчивость к атакам и перегрузкам.

В будущем возможно расширение функциональности сервера, добавление дополнительных методов защиты, а также оптимизация его производительности для более эффективной работы в условиях высоких нагрузок. Работа предоставила полезный опыт в разработке сетевых приложений и углубила знания в области сетевой безопасности, что является важным шагом для дальнейших исследований и практических разработок в данной области..

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Устройство TCP/Реализация SYN-flood атаки [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/782728/. – Дата доступа: 24.02.2025

[2] Протоколы семейства TCP/IP. Теория и практика [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/ruvds/articles/759988/. – Дата доступа: 24.02.2025