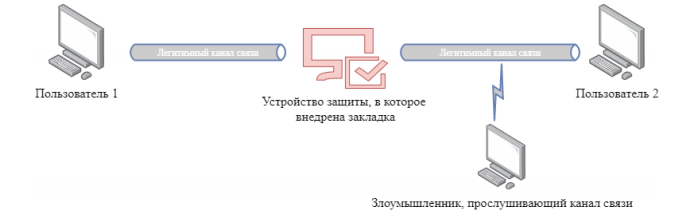
***Защита информации от утечки по скрытым каналам.***

***Отчет по лабораторной работе №1.***

***Вариант: Скрытый канал по времени на основе присутствия или отсутствия пакета в определённый промежуток времени в двустороннем консольном чате.***

***Введение***



На рисунке выше представлена принципиальная схема приложения, разработанного в ходе выполнения лабораторной работы. С помощью языка программирования python3 были реализованы три программных средства (client\_one.py, client\_two.py, firewall.py). Первое программное средство предназначено только для отправки консольных сообщений второму клиенту. Второе программное средство (firewall.py) предназначено для перенаправления сообщений с первого клиента на второго и наоборот. Также в это программное средство встроена «закладка» злоумышленника, которая позволяет отправлять бинарное сообщение определённой длины второму клиенту по скрытому каналу по времени. На втором клиенте помимо модуля, принимающего и отправляющего сообщения по открытому каналу, также реализован модуль, имитирующий работу злоумышленника. Этот модуль декодирует сообщение, отправленное ему по скрытому каналу.

***Описание работы программы***

1) client\_one.py.

В этом программном средстве предусмотрен только модуль отправки и принятия легитимных сообщений основного канала связи. Для этого используется библиотека socket и библиотека threading. С помощью библиотеки socket реализовано клиент-серверное взаимодействие между первым клиентом и фаерволом. Библиотека threading используется для использования двух потоков: первого для отправки сообщений второму клиенту в бесконечном цикле, а второй – для принятия сообщений от второго клиента также в бесконечном цикле.

2) firewall.py

В этом программном средстве используются такие же библиотеки, как и в предыдущем программном средстве. Модуль принятия и отправки сообщений работает следующим образом: если приходит пакет от первого клиента, то он пересылает его второму клиенту, а если пакет приходит от второго клиента, то наоборот. Также вторым потоком работает модуль отправки скрытого сообщения, который реализован следующим образом: пользователя просят ввести сообщение из пяти нулей и единиц, после успешного ввода начинает работу блок синхронизации, который нужен для того, чтобы второй клиент понял, что сейчас начнется передача скрытого сообщения. Для этого второму клиенту отправляются пять пакетов с межпакетным интервалом в одну секунду, причем на момент работы блока синхронизации все сообщения основного канала попадают в буфер, чтобы не нарушить блок синхронизации и если буфер не пустой, то блок использует сообщения оттуда для отправки, но если буфер пуст, то генерируется пустое сообщение и отправляется второму клиенту. После того, как отработал блок синхронизации начинается блок отправки скрытого сообщения. В этом блоке посимвольно разбивается введённое пользователем сообщение и в случае, если текущий символ – «1», то второму клиенту отправляется пакет с интервалом в 1 секунду (в этом блоке также используется буфер для сообщений из основного потока информации), а если текущий символ – «0», то канал просто «спит» одну секунду. Для синхронизации переменных между потоками для буферов используется библиотека queue. После успешной отправки скрытого сообщения, все сообщения из буфера отправляются второму пользователю.

3) client\_two.py

Данное программное средство похоже по устройству на первого клиента, однако в блок принятия сообщений встроен модуль «злоумышленника», который декодирует скрытое сообщение. Для этого, с помощью библиотеки time записывается время прихода каждого пакета, чтобы определить межпакетный интервал. Если было получено пять пакетов с межпакетным интервалом в одну секунду, то злоумышленник понимает, что прошел блок синхронизации и он начинает ждать скрытого сообщения. В блоке декодирования скрытого сообщения злоумышленник ловит приходящие пакеты, а также записывает время их прихода, после чего на основании полученных пакетов (присутствия или отсутствия в определённые моменты времени), он делает вывод о сообщении, передаваемом ему.

***Заключение***

В ходе данной лабораторной работы были получены практические навыки создания программных модулей для построения скрытых каналов, а также изучены принципы построения скрытых каналов по времени.