**Разработка инструмента определения IP-адреса собеседника при P2P-звонке.**

**Цель** - создать работающее решение, которое в режиме, близком к реальному времени, определяет и отображает в веб-интерфейсе публичный IP-адрес удаленного участника P2P-звонка, установленного через приложение мессенджер Chattti (<https://www.chattti.com/>).

**Ключевое требование**: Решение должно работать извне целевого мессенджера, без необходимости его модификации или нарушения лицензионных соглашений. Взаимодействие с приложением мессенджера недопустимо.

Для реализации задания была выбрана open-source библиотека gopacket, которая позволяет перехватывать трафик с определенного устройства (device) компьютера. Идея заключалась в том, чтобы перехватывать STUN пакеты, которыми обмениваются собеседники во время звонка, и оттуда узнавать айпи адрес собеседника.

Был написан файл capture.go, который, используя библиотеку gopacket перехватывает пакеты. Для этого изначально определяется устройство, на котором будут перехватываться пакеты. Имя устройства (вида \Device\NPF\_...) получается либо из env файла, либо, если строка пустая, определяется автоматически.

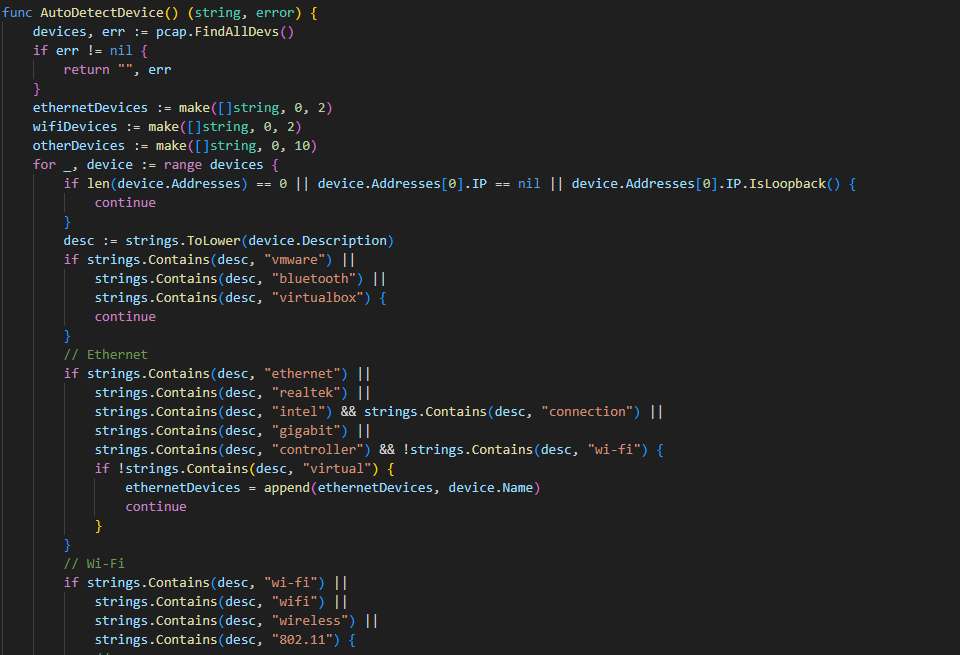


Рисунок 1 - Автоматическое определение устройства

Функция отбрасывает виртуальные устройства и в первую очередь выбирает Ethernet-адаптеры и Wi-Fi-адаптеры. Если таковых нет выбирается первый из списка. Поиск происходит по названию устройства, поэтому рекомендуется использовать определенное устройство, название которого можно найти используя функцию printDevices() (За нее отвечает LIST\_DEVICES=true в файле env).

После выбора устройства gopacket начинает перехватывать пакеты, приходящие и исходящие с данного айпи. Пакеты фильтруются, чтобы получать только UDP пакеты. Из них достается вся важная информация, такая как айпи и порт как отправителя, так и получателя, и полезная нагрузка.

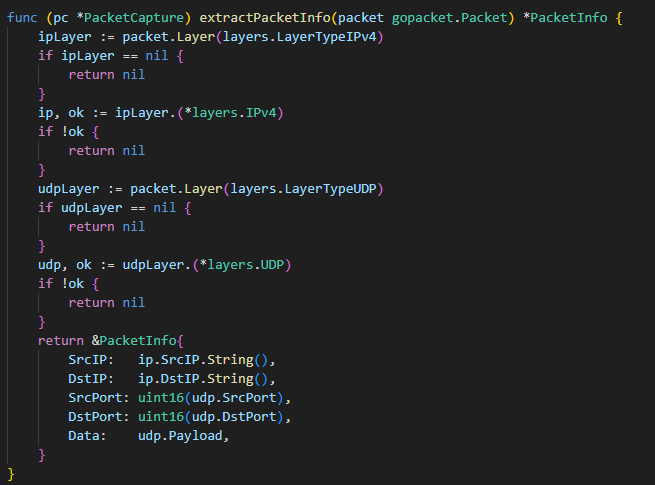
**

Рисунок 2 - Информация из пакета

После чего информация пакета отправляется в анализатор (analyzer.go), который анализирует содержимое пакета и отбрасывает ненужное, оставляя только пакеты между собеседниками.

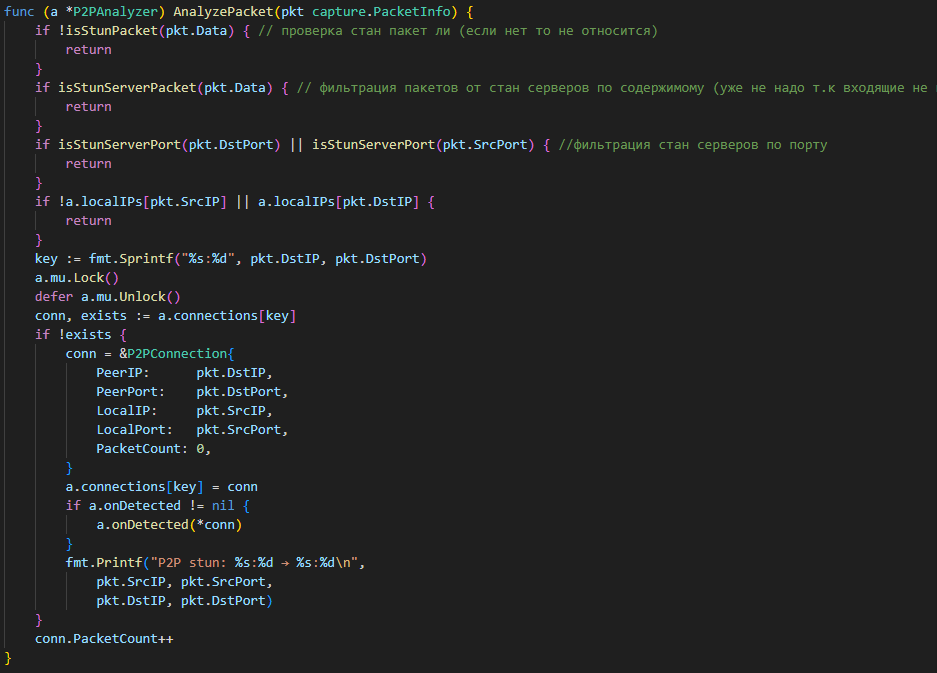


Рисунок 3 – Анализатор

Изначальная идея заключалась в том, чтобы анализировать пакеты, отбрасывая пакеты от STUN-серверов по Data из пакетов, однако, во время написания приложения, Chattti перестал работать без VPN, из-за чего стало невозможно получить входящие сообщения. Поэтому анализ свелся к проверке только отправляемых пакетов.

Сначала проверяется что пакет — это STUN-пакет. Для этого проверяется содержимое пакета, а именно 4-7 байты, в которых лежит Magic Cookie – фиксированное значение 0x2112A442. Далее фильтруются STUN-сервера по порту. Обычные порты STUN-серверов это 443, 3478, 5349, 80, а также отбрасываются все порты ниже 1024, т.к при звонке используются высокие порты.

Далее проверяется что пакет направляется нелокальному айпи-адресу, после чего фильтрация заканчивается, и информация добавляется в структуру.

После анализа пакет отправляется на сервер (server.go) и отображается на localhost.

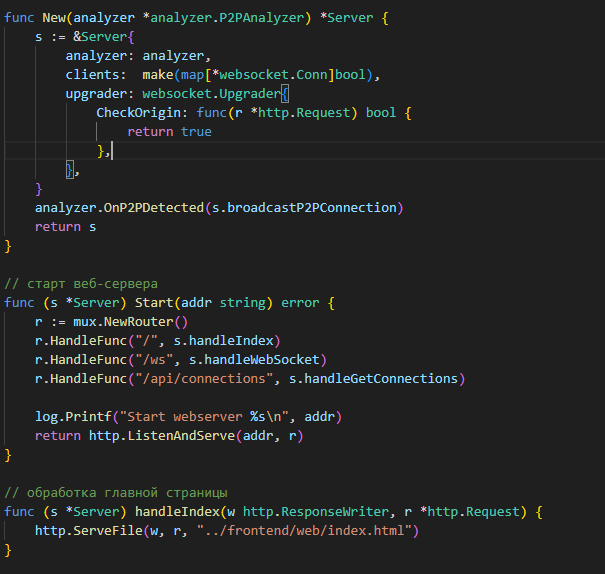


Рисунок 4 – Сервер

Однако, из-за использования VPN, p2p соединение не может нормально установиться, поэтому пакеты от одного собеседника к другому идут не напрямую, а через TURN-сервера, из-за чего полученный айпи-адрес в большинстве случаев принадлежит не собеседнику, а TURN-серверу.

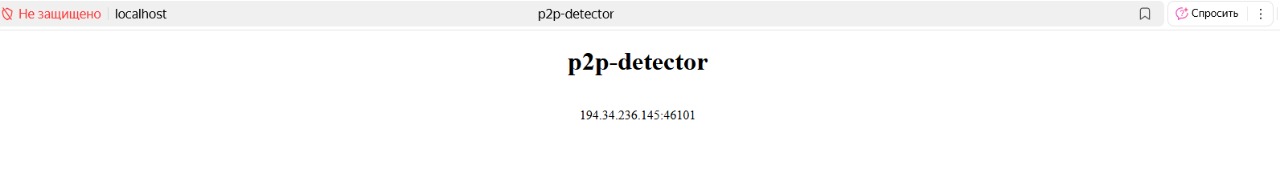


Рисунок 5 - Результат работы приложения

**Запуск приложения.**

Были написаны Docker-файлы для бэка и для фронта, также был написан docker-compose файл, поэтому для запуска приложения необходим .env файл с параметрами для запуска.



Рисунок 6 - env файл

DEVICE – название устройства, если пустое, то используется автоопределение.

LIST\_DEVICES – по стандарту true – булевая переменная для вывода всех устройств.

WEB – адрес и порт для запуска веба.

Чтобы запустить приложение используйте команду docker compose up –build –d. Логи, такие как список всех девайсов записываются в docker compose logs –f backend. Страница с вебом доступна на localhost.

Однако из-за того, что docker на windows запускает приложение на виртуальной машине, а программе нужен доступ к хосту, приложение на windows, запущенное через докер, работать не будет. Для Linux в docker-compose указан network\_mode:host, который дает доступ к хосту. Чтобы приложение работало на windows необходимо запускать приложение из папки backend, для этого нужно поменять первые 3 строки в функции main файла main.go на соответствующие значения из env-файла (deviceName и webAddr – string, listDevices - bool) и далее запускать приложение с помощью go run cmd\detector\main.go. В таком случае логи будут выписываться прямо в консоль.