# Общие сведения

Цель изучения: Применение в качестве посредника в проекте автоматизации сбора данных с хостов через сеть клиентов, которые позже эти данные передают сети серверам на обработку. С помощью этой библиотекой общаются сервера и клиенты, а сводит их посредник – веб сервер.

Язык разработки: Си (С)

Поддерживаемые целевые платформы:

* Linux (Debian, Ubuntu, Mint, CentOS, Fedora, Redhat, Amazon Linux, Arch Linux, OpenSUSE)
* BSD (FreeBSD, NetBSD, OpenBSD, DragonFlyBSD)
* Solaris 11
* Mac OS X
* Cygwin (for non-production R&D purposes)

Репозиторий не имеет конкретной реализации REST API, куда можно отправлять web запрос и получать данные для аутентификации. Репозиторий лишь описывает формат общения.

Отсутствует имплементация Short Term Credentials.

# Особенности

When used as a part of an ICE solution, for VoIP connectivity, this TURN server can handle thousands simultaneous calls per CPU (when TURN protocol is used) or tens of thousands calls when only STUN protocol is used. For virtually unlimited scalability a load balancing scheme can be used. The load balancing can be implemented with the following tools (either one or a combination of them):

* DNS SRV based load balancing;
* built-in 300 ALTERNATE-SERVER mechanism (requires 300 response support by the TURN client);
* network load-balancer server.

Источник: https://github.com/coturn/coturn

# Понятия и определения

## Таблица – общие

|  |  |  |
| --- | --- | --- |
| Аббревиатура | Термин | Определение |
| STUN | Session Traversal Utilities for NAT | это сетевой протокол, который позволяет клиенту, находящемуся за сервером трансляции адресов (или за несколькими такими серверами), определить свой внешний IP-адрес, способ трансляции адреса и порта во внешней сети, связанный с определённым внутренним номером порта. |
| TURN |  | это протокол, который позволяет узлу за NAT или брандмауэром получать входящие данные через TCP или UDP соединения. Такая возможность особенно актуальна для узлов позади симметричных NAT, или брандмауэров, которые собираются стать принимающей стороной в соединении с одним конкретным узлом (peer-ом). |
| VoIP | Voice over Internet Protocol | или IP-телефония - это голосовая связь через интернет |
| ICE | Interactive Connectivity Establishment | is a technique used in [computer networking](https://en.wikipedia.org/wiki/Computer_networking) to find ways for two computers to talk to each other as directly as possible in [peer-to-peer](https://en.wikipedia.org/wiki/Peer-to-peer) networking. |
| WebRTC | Web real-time communications | (коммуникации в реальном времени) — проект с открытым исходным кодом, предназначенный для организации передачи потоковых данных между браузерами или другими поддерживающими его приложениями по технологии точка-точка. |
| OAuth |  | ткрытый протокол (схема) авторизации, который позволяет предоставить третьей стороне ограниченный доступ к защищённым ресурсам пользователя без необходимости передавать ей (третьей стороне) логин и пароль |
| ICMP | Internet Control Message Protocol | сетевой протокол, входящий в стек протоколов TCP/IP. В основном ICMP используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна, или хост, или маршрутизатор не отвечают. Также на ICMP возлагаются некоторые сервисные функции (services). |
| ALPN | Application-Layer Protocol Negotiation | is a Transport Layer Security (TLS) extension that allows the application layer to negotiate which protocol should be performed over a secure connection in a manner that avoids additional round trips and which is independent of the application-layer protocols. It is used to establish HTTP/2 connections without additional round trips (client and server can communicate over to ports previously assigned to HTTPS with HTTP/1.1 and upgrade to use HTTP/2 or continue with HTTP/1.1 without closing the initial connection). |
| SDP | Session Description Protocol | RFC 3264 |
| SIP | Session Initiation Protocol | RFC 3261 |
| TLS | transport layer security | Протокол TLS (transport layer security) основан на протоколе SSL (Secure Sockets Layer), изначально разработанном в Netscape для повышения безопасности электронной коммерции в Интернете. Протокол SSL был реализован на application-уровне, непосредственно над TCP (Transmission Control Protocol), что позволяет более высокоуровневым протоколам (таким как HTTP или протоколу электронной почты) работать без изменений. Если SSL сконфигурирован корректно, то сторонний наблюдатель может узнать лишь параметры соединения (например, тип используемого шифрования), а также частоту пересылки и примерное количество данных, но не может читать и изменять их. |
| TLV | type-length-value | широко распространённый метод записи коротких данных в компьютерных файлах и телекоммуникационных протоколах.  Метод определяет простую двоичную структуру из трёх полей: тег, длина данных и собственно данные. Первые два поля имеют фиксированную длину (обычно один или два октета на поле), длина третьего поля определяется значением второго поля (значение указывается в байтах). Тег является идентификатором данных, определяя их назначение. |
| HMAC | *hash-based message authentication code* | код аутентификации (проверки подлинности) сообщений, использующий хеш-функции, или как англ. keyed-hash message authentication code, код аутентификации сообщений, использующий хеш-функции с ключом) — в информатике (криптографии), один из механизмов проверки целостности информации, позволяющий гарантировать то, что данные, передаваемые или хранящиеся в ненадёжной среде, не были изменены посторонними лицами (см. человек посередине). |

## Таблица – Понятия и определения из стандартов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Понятие | Аббревиатура официальная | Аббревиатура не официальная | Описание |  |
| HOST TRANSPORT ADDRESS |  |  |  |  |
| TRANSPORT ADDRESS |  |  | The combination of an IP address and port | <https://datatracker.ietf.org/doc/html/rfc5766#section-2> |
| TURN SERVER TRANSPORT ADDRESS |  |  | Адрес по которому TURN сервер слушает все новые входящие соединения | <https://datatracker.ietf.org/doc/html/rfc5766#section-2> |
| Server-Reflexive Transport Address |  |  | Адрес клиента для обращения к нему из вне. Как клиента видит сервер. |  |
| Relayed Transport Address |  |  | The relayed transport address is the transport address on the server that peers can use to have the server relay data to the client.  Адрес клиентов подключенных к серверу относительно самого сервера. По этому адресу клиенты могу общаться между собой через этот сервер. |  |

# Порядок изучения

|  |  |
| --- | --- |
| Источник | Описание |
| https://github.com/coturn/coturn | Общая информация |
| https://ourcodeworld.com/articles/read/1175/how-to-create-and-configure-your-own-stun-turn-server-with-coturn-in-ubuntu-18-04 | Инструкция от заказчика |
| https://github.com/coturn/coturn/blob/master/examples/etc/turnserver.conf | Пример конфигурации |

# Принцип работы:

## Клиент

* Клиент проходит аутентификацию и авторизацию. Затем запрашивает аллокацию места на сервере.
* В ответе он получает свой внешний адрес и маршрутный адрес, по которому другие пиры отправляют ему сообщенния
* Затем клиент запрашивает у сервера проверку на подключения для сбора данных об активных пирах, которым может быть отправлены сообщения

# Documentation Research

## RFC 5245 - Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols

Описывает протокол NAT маршрутизации для UDP мультимедиа сессий, установленных по модели предожение/ответ.

<https://datatracker.ietf.org/doc/html/rfc5245#section-1>

Таблица - основные цитаты документа RFC5245

|  |  |  |  |
| --- | --- | --- | --- |
|  | Sections | Page/Indent | Описание |
| ICE allows the agents to discover enough information about their topologies to potentially find one or more paths by which they can communicate. | <https://datatracker.ietf.org/doc/html/rfc5245#section-2> | 7/1 |  |
|  |  |  |  |

Таблица – Пересказ параграфов документа

|  |  |  |
| --- | --- | --- |
| Параграф | Описание | Комментарии |
| 1. Introduction |  |  |
|  |  |  |

## RFC 5389 - Session Traversal Utilities for NAT (STUN)

Описывает некий сервис, как инструмент для других протоколов во взаимодействии с NAT маршрутизацией. Этот некий сервис используется для получения конечным узлом IP адреса и номер порта, выданный ему NAT’ом.

Обязательно познакомиться с терминологией: <https://datatracker.ietf.org/doc/html/rfc5389#section-5>

Таблица - Цитаты

|  |  |  |  |
| --- | --- | --- | --- |
| Цитата | Источник | Page | Описание |
| Although there are four message classes, there are only two types of transactions in STUN: request/response transactions (which consist of a request message and a response message) and indication transactions (which consist of a single indication message). | <https://datatracker.ietf.org/doc/html/rfc5389#section-6> | 10 | <https://datatracker.ietf.org/doc/html/rfc5389#section-10.1.1>  Есть реквесты, а есть индикации |
| The magic cookie field MUST contain the fixed value 0x2112A442 in network byte order. | <https://datatracker.ietf.org/doc/html/rfc5389#section-6> | 10 | Значение из формата сообщения должно содержать константу |
| STUN Indication: A STUN message that does not receive a response. | <https://datatracker.ietf.org/doc/html/rfc5389#section-5> | 9 | Определение |
| STUN is a client-server protocol. It supports two types of transactions. One is a request/response transaction in which a client sends a request to a server, and the server returns a response. The second is an indication transaction in which either agent -- client or server -- sends an indication that generates no response. Both types of transactions include a transaction ID, which is a randomly selected 96-bit number. | <https://datatracker.ietf.org/doc/html/rfc5389#section-3> | 5 |  |

## RFC 5766 - Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)

Является основным документом описывающим стандарт TURN

Таблица – Основные цитаты документа

|  |  |  |  |
| --- | --- | --- | --- |
| Цитата | Источник | Page/Indent | Описание |
| indications in TURN are never authenticated. | <https://datatracker.ietf.org/doc/html/rfc5766#section-4> | 20 |  |
| All requests after the initial Allocate must use the same username as that used to create the allocation, to prevent attackers from hijacking the client's allocation. | <https://datatracker.ietf.org/doc/html/rfc5766#section-4> | 20/4 |  |

Таблица – Пересказ параграфов

|  |  |  |
| --- | --- | --- |
| Параграф | Описание | Комментарий |
| 2. Overview of Operation | 1 абзац: общая информация о параграфе  2 абзац: описание типичной конфигурации среды применения стандарта  3 абзац: описание схемы  4 абзац: описание первого действия от клиента к серверу. Определение понятий HOST TRANSPORT ADDRESS и TRANSPORT ADRESS  5 абзац: описание действия клиента и указание источники адреса сервера. Определение понятия TURN SERVER TRANSPORT ADDRESS  6 абзац: определение понятия SERVER-REFLEXIVE  7 абзац: определение понятия RELAYED TRANSPORT ADDRESS. Описание аллокации на сервере  8 абзац: описание отправки клиентом данных пирам через сервер  9 абзац: определение адреса отправки от клиента к пиру  10 абзац: отношение аллокации у сервера к клиенту  11 абзац: клиент может иметь несколько аллокаций на сервере |  |
| 4. General Behavior | 1 абзац: Эта секция (документа) содержит основные правила обработки применяемые ко всем сообщениям формата TURN.  2 абзац: TURN построен на основе STUN и наследует все его правила.  3 абзац: заметки для Long-Term Credential Mechanism  4 абзац: содержание сообщений после операции Allocation  5 абзац: Как сервер валидирует входящие сообщения на предмет подлинности авторизационных данных и какие ощибки отправляет  6 абзац: рекомендации по атрибутам SOFWARE и FINGERPRINT  7 абзац: TURN не наследует механизмы обратной совместимости STUN (не поддерживает его предыдущий функционал, только новый)  8 абзац: в этой документации TURN поддерживает лишь IPv4  9 абзац: порты TURN по умолчанию  10 абзац: TURN обязан поддерживать UDP протокол и должен (SHOULD) поддерживать TCP И UDP.  11 абзац: описание механизма повторной отправки по UDP если ответ от сервера не был получен. Описывает надежность передачи  12 абзац: обработка ошибок при передаче по TCP (рассинхронизация)  13 абзац: защита от DOS атак |  |
| 5. Allocations | 1 абзац: составляющие аллокации  2 абзац: описание 5-tuple и адреса передачи (поэлементое описание предыдущего абзаца)  3 абзац: уникальность 5-tuple и адреса передачи  4 абзац: применение данных аутентификации  5 абзац: время жизни аллокации  6 абзац: ссылка на описание разрешений |  |
| 16. Detailed Example | 1 абзац: общая информация о параграфе  2 абзац: описание формата отображения атрибутов на схеме  3 абзац: описание процесса отправки данных от клиента аллокации к серверу  4 абзац: описание авторизации сервером. Описание обработки сервером первого сообщения от клиента  5 абзац: описание повтора аллокации от клиента к серверу для аутентификации  6 абзац: описание кода аутентификации в атрибуте message integrity  7 абзац: обработка сервером запроса на аллокации который был аутентифицирован  8 абзац: создание разрешения от клиента к пиру  9 абзац: получение сервером запроса на создание разрешения от клиента к пиру  10 абзац: отправка данных от клиента к пиру  11 абзац: обработка сервером индикации на отправку клиентом  12 абзац: ответ от пира к клиенту через индикацию отправки  13 абзац: привязка клиентом канала к пиру через сервер для отправки канальных сообщений  14 абзац: прием команды на привязку канала сервером от клиента  15 абзац: отправка канального сообщения клиентом к пиру  16 абзац: отправка данных через канальное сообщение от пира к клиенту  17 абзац: обновление аллокации клиентом |  |

## RFC 6544 - TCP Candidates with Interactive Connectivity Establishment (ICE)

Является расширением основного документа описывающего стандарт ICE. Определяет ICE для протокола TCP

Абзац считается от начала ссылки.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Стр/Абз |  |
| ICE works by providing a set of candidate transport addresses for each media stream, which are then validated with peer-to-peer connectivity checks based on Session Traversal Utilities for NAT (STUN). | https://datatracker.ietf.org/doc/html/rfc6544 | 1/1 |  |
| When agents perform address allocations to gather TCP-based candidates, three types of candidates can be obtained: active candidates, passive candidates, and simultaneous-open (S-O) candidates. | <https://datatracker.ietf.org/doc/html/rfc6544#section-3> | 4/ |  |

Таблица - Пересказ документации

|  |  |  |  |
| --- | --- | --- | --- |
| Параграф | Описание | Комментарий | Основные моменты |
| 1. Introduction |  |  |  |
|  |  |  |  |

# Практическое руководство

## Работа с репозиторием

## Сборка на Windows

Официально не поддерживается, но возможна по запросу.

## Сборка на Linux

## Имплементация в .Net проект

Проект выполнен на Си. Возможность использования как библиотеки не рассматривалась 2021/10/20.

## Порядок внедрения

|  |  |
| --- | --- |
| Действие | Комментарий |
| Установка зависимостей | Установить пакеты. |
| Установка Coturn | Скомпилить или скачать через пакетный менеджер. |
| Конфигурирования файла /etc/default/ coturn |  |
| Конфигурирование файла turnserver.conf |  |
| Установка и настройка БД |  |

Таблица – Действия для отладки

|  |  |
| --- | --- |
| Действие |  |
| apt install net-tools |  |

## Установка на Ubuntu

|  |  |
| --- | --- |
| Команда | Описание |
|  |  |

## Конфигурация сервера Coturn

<https://github.com/coturn/coturn/wiki/CoturnConfig>

<https://ourcodeworld.com/articles/read/1175/how-to-create-and-configure-your-own-stun-turn-server-with-coturn-in-ubuntu-18-04>

<https://github.com/coturn/coturn/wiki/turnserver>

### Список конфигураций

|  |  |  |  |
| --- | --- | --- | --- |
| Конфигурация | Описание | По умолчанию | Установлено |
| realm |  |  | localhost |
| **server-name** |  |  |  |
| **listening-ip** |  |  |  |
| **tls-listening-port** |  |  |  |
| **usercombo** | TURN REST API |  |  |
| **turn user** | TURN REST API |  |  |
| **turn password** | TURN REST API |  |  |
| **use-auth-secret** | TURN REST API |  |  |
| **lt-cred-mech** | TURN REST API |  |  |
| **dh2066** | avoid console  ERROR: set\_ctx: ERROR: cannot set DH  https://github.com/coturn/coturn/issues/629 |  |  |

Таблица – Конфигурации для разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Конфигурация | Описание | По умолчанию | Установлено |
| web-admin-ip |  |  | 127.0.0.1 |
| web-admin-port |  |  | 8080 |
| no-auth | Выключить авторизацию |  |  |
| log-file |  | log-file =/var/tmp/turn.log |  |

### Установка сертификата локально для разработки

|  |  |
| --- | --- |
| sudo apt install libnss3-tools -y  or  apt-get install wget libnss3-tools | <https://kifarunix.com/how-to-create-self-signed-ssl-certificate-with-mkcert-on-ubuntu-18-04/>  <https://www.howtoforge.com/how-to-create-locally-trusted-ssl-certificates-with-mkcert-on-ubuntu/?__cf_chl_captcha_tk__=pmd_ihIT2PmDOwG7Udhsdt7oo.LhBQ.HQFW_HwwEn4K.Gw4-1634631367-0-gqNtZGzNAyWjcnBszQjl> |
| wget <https://github.com/FiloSottile/mkcert/releases/download/v1.4.3/mkcert-v1.4.3-linux-amd64> |  |
| sudo cp mkcert-v1.4.3-linux-amd64 /usr/local/bin/mkcert |  |
| sudo chmod +x /usr/local/bin/mkcert | Сделать файл mkcert исполняемым |
| mkcert -install  mkcert -install testcoturn.com | Создание сертификатов root и доменный |
| mkcert -CAROOT | Показать директорию сертификата |

\*Пока не работает этот сертификат с coturn

### Настройка базы данных

Создание схемы БД:

<https://github.com/coturn/coturn/blob/master/turndb/schema.sql>

Инструкции:

<https://github.com/coturn/coturn/wiki/turnserver#databases>

<https://github.com/coturn/coturn/wiki/turnserver#user-database-settings>

|  |
| --- |
| By default, coturn uses SQLite database for users and settings. That database is automatically created (empty) when the process turnserver starts the first time.  Источник: <https://github.com/coturn/coturn/wiki/CoturnConfig#coturn-installation>  Default path:  /var/lib/turn/turndb |

|  |  |
| --- | --- |
| Операция | Описание |
| Создать схему БД если используется конфигурация не по умолчанию (например используется MySQL) |  |
| Создать в БД пользователя для создания подключений и дать соответствующие права для возможности подключения и редактирования записей в БД |  |
| sudo bin/turnadmin -A -u admin -p 1234 – не работает  sudo turnadmin -A -u admin -p 1234 | Add the admin user(s) to the database, with turnadmin utility. If you added those users, then will be able to connect to the TURN server ports over HTTPS from your browser and perform admin tasks with the web interface.  Источник: <https://github.com/coturn/coturn/wiki/CoturnConfig#coturn-installation>  Добавляет админа в web-admin панель  <https://localhost:8080/> |
| sudo turnadmin -A -u admin -p 1234 --mysql-userdb="host=192.168.0.3 dbname=coturn user=ngoc password=W7jGdybD port=3306 connect\_timeout=30 read\_timeout=30" | Добавть пользователя для web-admin в MySql БД |

#### PostgreSQL

Файл схемы находится:

<https://github.com/coturn/coturn/blob/master/turndb/schema.sql>

Далее в pgAdmin4 в выбранной базе данных через контекстное меню выбрать Query Tool и выполнить код из файла схемы

### Настройка пользователя

https://github.com/coturn/coturn/wiki/turnadmin

### Конфигурация web-admin panel

<https://github.com/coturn/coturn/blob/master/README.turnadmin>

|  |  |
| --- | --- |
| Команда | Описание |
| turnadmin -P |  |

### Конфигурация контейнера Docker

<https://hub.docker.com/r/coturn/coturn>

|  |  |
| --- | --- |
| docker ps -a | Получить container\_id для подключения в следующей команде |
| docker exec -u root -t -i b20777f58ad6 /bin/bash | b20777f58ad6 – container\_id из предыдущей команды  подключение с правами администратора в командную оболочку контейнера |
| cd etc/coturn |  |
| touch turnserver.conf | Создание файла конфигурации |
| apt-get update |  |
| apt-get install nano |  |
| nano turnserver.conf | Редактировать файл конфигурации. Пример без авторизации:  psql-userdb="host=host.docker.internal port=5432 dbname=coturn user=postgres password=postgres connect\_timeout=60"  no-auth  realm=izihard  verbose |
|  | [Создать БД по схеме](#_PostgreSQL) |
|  | Перезапустить контейнер |

## Работа с Rest API

<https://github.com/coturn/coturn/wiki/turnserver#turn-rest-api>

Таблица - Последовательность взаимодействия

|  |  |
| --- | --- |
| Действие | Описание |
| Получение Get запросом данных для входа | англ. - get credentials. |

## Соединение двух точек (TurnClient) через Turn сервер

## Сбор клиентом списка активных пиров

Таблица – Источники для решения задачи

|  |  |  |
| --- | --- | --- |
| Источник | Содержание | Описание |
| <https://datatracker.ietf.org/doc/html/rfc5768#section-1> | ICE defines procedures by which agents gather a multiplicity of addresses, include all of them in an SDP offer or answer, and then use peer-to-peer Session Traversal Utilities for NAT (STUN) [RFC5389] connectivity checks to determine a valid address. | Для UDP подключений |
| <https://datatracker.ietf.org/doc/html/rfc6544#section-7.1> | TCP Candidates with Interactive Connectivity Establishment  Interactive Connectivity Establishment (ICE) defines a mechanism for NAT traversal for multimedia communication protocols based on the offer/answer model of session negotiation. | Тут только теория и описание |
|  |  |  |

Задача: Получить список пиров подключенных к серверу в формате для возможности отправки им сообщения.

Проблема: Где искать? Где конкретная реализация метода/протокола? В каком конкретно документе указан способ?

Общий принцип на уровня обывателя: Когда клиент подключается к серверу, и фиксируется на динамически выделенном порту, то запись о подключении попадает в общий реестр. Должен быть способ получения этих данных. Если сервер с авторизацией, то получение списка возможна только после авторизации.

## Тестирование

<https://datatracker.ietf.org/doc/html/rfc5769#section-2.4>

Рекомендации:

|  |  |
| --- | --- |
| /etc/turnserver.conf - полностью очистить, чтобы не заблудиться в конфиге. |  |
| Стартовать сервер не через  systemctl start coturn  а через:  turnserver -a  и каждый конфиг добавлять построчно |  |
| В консоль показать документацию:  man turnadmin  man -M man turnadmin |  |

Как интерпретировать вывод консоли сервера

<https://github.com/coturn/coturn/issues/224>

sp = Sent Packets

sb = Sent Bytes

rp = received Packets

rb = Received Bytes

STALE CREDENTIALS

INTEGRITY CHECK FAILURE

MISSING USERNAME

USE TLS

SERVER ERROR

GLOBAL FAILURE

FORBIDDEN

ALLOCATION MISMATCH

WRONG CREDENTIALS

UNSUPPORTED TRANSPORT PROTOCOL

ALLOCATION QUOTA REACHED

INSUFFICIENT CAPACITY

## Полная инструкция

Замены:

<YOUR\_DOMAIN> => coturn.softorium.pro

|  |  |
| --- | --- |
| Действие | Описание |
| sudo apt-get -y update  sudo apt-get install coturn  systemctl stop coturn |  |
| nano /etc/default/coturn | Редактировать  TURNSERVER\_ENABLED=1 |
| sudo apt-get -y install certbot  sudo certbot certonly --standalone  cd /etc/letsencrypt/live/coturn.softorium.pro/  sudo openssl dhparam -out dhparam.pem 2048 | Установить certbot и с помощью мастера создать сертификат.  Данные для мастера:  Почта: tran@softorium.pro  Домен: coturn.softorium.pro |
| sudo nano /etc/turnserver.conf | cert=/etc/letsencrypt/live/coturn.softorium.pro/cert.pem  pkey=/etc/letsencrypt/live/coturn.softorium.pro/privkey.pem  dh-file=/etc/letsencrypt/live/coturn.softorium.pro/dhparam.pem  cipher-list="ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA38"  realm=coturn.softorium.pro  server-name=coturn.softorium.pro  dh2066  listening-ip=95.181.45.166  listening-ip=127.0.0.1  listening-port=3478  tls-listening-port=5349  #Long Term Credentials  lt-cred-mech  user1=username1:password1  user2=username2:password2  #debug  web-admin  cli-password=qwerty |
| turnserver -a | Тестируем и вручную исправляем конфигурацию по логам.  Проверяем чтобы не было пробелов в конце конфигов |
| turnserver -a -v --log-file=stdout | Тестируем и смотрим подклчюения |
| sudo ss -lnpt | grep turnserver | Смотрим статистику сокетов |

Возможные ошибки:

|  |  |  |  |
| --- | --- | --- | --- |
| Ошибка с вывода | Пробовал, не помогло | Пробовал, помогло | Источник |
| bind: Cannot assign requested address  Cannot bind local socket to addr: Cannot assign requested address  0: Cannot bind TLS/TCP listener socket to addr 95.181.45.166:3478  0: Trying to bind TLS/TCP listener socket to addr 95.181.45.166:3478, again...  0: Trying to bind fd 18 to <95.181.45.166:3478>: errno=99 | Не лечится:  sudo ufw allow 3478/tcp  sudo ufw allow 3478/udp | Указал в конфиге  external-ip=95.181.45.166/192.168.212.63  убрал все  listening-ip  Возможно также что coturn уже запущен и нужно его отключить  systemctl stop coturn | https://github.com/coturn/coturn/issues/21 |
|  |  |  |  |

# Разработка

## Клиент

Таблица – Особенности разработки на .Net

|  |  |
| --- | --- |
|  |  |
| Примитивные типы в C# - Little Endian, а формат заголовков сервера – Big Endian. | Это значит, что при побайтовой записи значения, например, типа и класса STUN header в UInt16 при сравнении отображаемых значений в последнем будет не 0x09 и 0x00, а 0x00 затем 0x09 |

### Получение списка активных пиров для установления соединения

Документ [RFC 5245](https://datatracker.ietf.org/doc/html/rfc5245#section-2.1) определяет процесс сбора кандидатов (UDP). [rfc6544](https://datatracker.ietf.org/doc/html/rfc6544) для TCP по протоколу ICE.

После поиска в <https://datatracker.ietf.org/doc/html/rfc5245#section-2> прихожу в выводу, что нет внутреннего механизма для получения списка активных пиров. Сам ICE – это протокол, который позволяет установить самый короткий путь между двумя известными TURN клиентами (или STUN). Он комбинирует все активные подключения одного узла со всеми активными подключениями другого узла и выявляет самую эффективную комбинацию. Ключевой момент – оба узла уже знают друг о друге.

Для того чтобы получить список клиентов, подключенных к TURN серверу, необходимо написать приложение, работящее с базой данных TURN сервера. Если посмотреть в [приложение](#_Как_выглядит_панель) то мы увидим, что в базе хранятся данные о активных сессиях. Разрабатываемое приложение должно по запросу с соответствующим фильтром выдавать данные о подходящих пирах. В ответе обязательно должен быть RELAYED TRANSPORT ADDRESS каждого пира, для того чтобы клиент мог отправить им данные через сервер.

Так как структура базы данных у coturn сервера определена, то лучше всего выполнить импорт схемы через EF Core.

|  |  |
| --- | --- |
| Coturn поддерживает старую версию протокола ice – RFC 5245 | Поиски решения по сбору данных об активных подключения начинаются отсюда |
|  |  |

### Проблемы в ходе разработки

<https://datatracker.ietf.org/doc/html/rfc5389#section-10.1.1>

Хоть в документации и написано что должно содержаться USERNAME and MESSAGE-INTEGRITY аттрибута а про другие не сказано, в coturn реализации не будет понят USERNAME если не будет аттрибута REALM. То есть он по факту тоже обязателен.

## Сервер

# Проблемы

## Динамические IP адреса у Turn клиентов

Суть проблемы: у клиента и сервера устанавливается устойчивое подключение. Вопрос в том, что у клиента может измениться адрес, что делает его недоступным для принятия данных (не уверен). Если бы только можно было решить вопрос через периодический опрос клиента на состояние и обновления данных об этом клиенте на сервере в случае изменении адреса, просто убив подключение и пересоздав его.

<https://github.com/coturn/coturn/issues/469#issuecomment-561095175>

<https://github.com/processone/ejabberd/issues/3278>

# Альтернативные STUN/TURN решения

|  |  |
| --- | --- |
| Источник | Описание |
| <https://eturnal.net/> |  |
| <https://github.com/processone/ejabberd> | ejabberd is a distributed, fault-tolerant technology that allows the creation of large-scale instant messaging applications. The server can reliably support thousands of simultaneous users on a single node and has been designed to provide exceptional standards of fault tolerance. As an open source technology, based on industry-standards, ejabberd can be used to build bespoke solutions very cost effectively. |
| <https://github.com/sipsorcery-org/sipsorcery> | This fully C# library can be used to add Real-time Communications, typically audio and video calls, to .NET applications. |
| <https://github.com/HMBSbige/NatTypeTester> |  |
| <https://github.com/sq5gvm/JabberNet-2010> | Jabber-NET 2010 |

# Источники

## Теория и инуструкции

Таблица - Теория и инструкции

|  |  |
| --- | --- |
| Ссылка | Описание |
| <https://github.com/coturn/coturn> | Репозиторий проекта |
| <https://datatracker.ietf.org/doc/html/draft-uberti-behave-turn-rest-00> | Источник с <https://github.com/coturn/coturn>.  DTLS versions 1.0 and 1.2.  Datagram Transport Layer Security (DTLS) as Transport for Traversal  Using Relays around NAT (TURN) |
| <https://ru.wikipedia.org/wiki/STUN> | STUN |
| <https://ru.wikipedia.org/wiki/Traversal_Using_Relay_NAT> | TURN |
| <https://www.3cx.ru/voip-sip/stun-server/> | STUN Server |
| <https://en.wikipedia.org/wiki/Interactive_Connectivity_Establishment> | ICE |
| <https://ru.wikipedia.org/wiki/WebRTC> | WebRTC |
| <https://www.youtube.com/watch?v=4dLJmZOcWFc> | What are STUN and TURN Servers? (WebRTC Tips from WebRTC.ventures) |
| <https://ru.wikipedia.org/wiki/OAuth> | OAuth |
| <https://www.youtube.com/watch?v=JrxSDIJv5xs> | Swatting NATs with TURN and WebRTC |
| <https://gabrieltanner.org/blog/turn-server>  <https://arstech.net/how-to-install-turn-server-on-ubuntu-for-webrtc/>  <https://www.linuxbabe.com/linux-server/install-coturn-turn-server-spreed-webrtc>  <https://ourcodeworld.com/articles/read/1175/how-to-create-and-configure-your-own-stun-turn-server-with-coturn-in-ubuntu-18-04>  <https://docs.bigbluebutton.org/admin/setup-turn-server.html> | Как настроить конфигурацию coturn  How to set up and configure your own TURN server using Coturn |
| <https://community.letsencrypt.org/t/how-to-create-certificates-for-a-development-environment-in-a-local-network/122910> | Получить сертификат для разработки в локальной сети |
| <https://github.com/FiloSottile/mkcert> | Генерация сертификата для разработки |
| <https://kifarunix.com/how-to-create-self-signed-ssl-certificate-with-mkcert-on-ubuntu-18-04/> | Генерация сертификата для разработки |
| <https://www.py4u.net/discuss/1737497>  Нет ответа:  <https://stackoverflow.com/questions/35766382/coturn-how-to-use-turn-rest-api>  <https://progi.pro/coturn-kak-ispolzovat-api-turn-rest-4890305> | CoTURN: How to use TURN REST API? |
| <https://github.com/marcplouhinec/webrtcdemo/blob/master/infrastructure/10_stun_turn_server/resources/install_stunturn_server.sh>  Вроде есть ответ:  <https://stackoverflow.com/questions/64362321/webrtc-coturn-authentication-using-turn-rest-api-flag-use-auth-secret-based> | Пример coturn + mkcert |
| <https://developer.mozilla.org/ru/docs/Web/API/WebRTC_API/Protocols> | Введение в протоколы WebRTC |
| <https://ru.wikipedia.org/wiki/Порядок_байтов> | Порядок байтов |
| <https://ru.wikipedia.org/wiki/HMAC> | HMAC |
| <https://csharp.hotexamples.com/examples/StringPrep/SASLprep/-/php-saslprep-class-examples.html> | SASLerp examples |
| <https://datatracker.ietf.org/doc/html/rfc5769#section-2.4> | Тестирование STUN |
| <https://misi.github.io/webrtc-c0d3l4b/#0> | WebRTC |
| <https://www.ietf.org/proceedings/87/slides/slides-87-behave-10.pdf> | TURN REST Server API  Схема авторизации |
| <http://www.stunprotocol.org/> | STUNTMAN is an open source implementation of the STUN protocol (Session Traversal Utilities for NAT) as specified in RFCs 5389, 5769, and 5780. It also includes backwards compatibility for RFC 3489. Source code distribution includes a high performance STUN server, a client application, and a set of code libraries for implementing a STUN client within an application. |
| <https://www.html5rocks.com/en/tutorials/webrtc/infrastructure/> | What is signaling? |
| https://russianblogs.com/article/55901107430/ | Создайте сервер ICE в Ubuntu (Coturn) |

## Стандарты

Таблица – Стандарты

|  |  |  |  |
| --- | --- | --- | --- |
| Код | Источник | Заголовки и содержание | Ремарки |
| DRAFT | <https://datatracker.ietf.org/doc/html/draft-uberti-behave-turn-rest-00> | Документация: A REST API For Access To TURN Services  draft-uberti-behave-turn-rest-00 | Описание взаимодействия |
| 5245 | [https://datatracker.ietf.org/doc/html/rfc5245](https://datatracker.ietf.org/doc/html/rfc5245#section-2.1) | Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols | <https://datatracker.ietf.org/doc/html/rfc5245#section-15.1>  Описание взаимодействия, но только UDP  Указаны новые атрибуты и ошибки для STUN |
| 8445 | <https://www.rfc-editor.org/rfc/rfc8445.html#section-20.1> | Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal | Заменяет 5245 (coturn не работает по 8445)  Указаны новые атрибуты и ошибки для STUN |
| 5389 | <https://datatracker.ietf.org/doc/html/rfc5389> | Session Traversal Utilities for NAT (STUN) | Описание формата сообщений  Перечень атрибутов и методов  Аутентификация  Базовый протокол процедур |
| 5766 | <https://datatracker.ietf.org/doc/html/rfc5766> | Traversal Using Relays around NAT (TURN):  Relay Extensions to Session Traversal Utilities for NAT (STUN) | * UDP * Более детальное описание процесса передачи сообщений между пирами через STUN * Описание операций и формата сообщений и атрибутов * Указаны новые методы, атрибуты и коды ошибок для STUN |
| 5780 | <https://www.rfc-editor.org/rfc/rfc5780.html> | NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN)  This specification defines an experimental usage of the Session Traversal Utilities for NAT (STUN) Protocol that discovers the presence and current behavior of NATs and firewalls between the STUN client and the STUN server. | Указаны новые атрибуты, |
| 5928 | <https://datatracker.ietf.org/doc/html/rfc5928> | Traversal Using Relays around NAT (TURN) Resolution Mechanism  This document defines a resolution mechanism to generate a list of server transport addresses that can be tried to create a Traversal Using Relays around NAT (TURN) allocation. | Описание взаимодействия |
| 6062 | <https://datatracker.ietf.org/doc/html/rfc6062> | Traversal Using Relays around NAT (TURN) Extensions for TCP Allocations. | Описание подключение к серверу  Описание поведения Turn клиента  Описание поведения Turn сервера |
| 6544 | <https://datatracker.ietf.org/doc/html/rfc6544> | TCP Candidates with Interactive Connectivity Establishment (ICE)  Interactive Connectivity Establishment (ICE) defines a mechanism for NAT traversal for multimedia communication protocols based on the offer/answer model of session negotiation. ICE works by providing a set of candidate transport addresses for each media stream, which are then validated with peer-to-peer connectivity checks based on Session Traversal Utilities for NAT (STUN). ICE provides a general framework for describing candidates but only defines UDP-based media streams. This specification extends ICE to TCP-based media, including the ability to offer a mix of TCP and UDP-based candidates for a single stream. | Описание взаимодействия |
| 7443 | <https://datatracker.ietf.org/doc/html/rfc7443> | Application-Layer Protocol Negotiation (ALPN) Labels for Session Traversal Utilities for NAT (STUN) Usages | Описание взаимодействия |
| 4013 |  |  |  |
| DRAFT | <https://datatracker.ietf.org/doc/html/draft-reddy-behave-turn-auth-04> | Problems with STUN Authentication for TURN |  |

## Документация

|  |  |
| --- | --- |
| <http://interoperability.blob.core.windows.net/files/MS-TURN/%5BMS-TURN%5D.pdf> | Traversal Using Relay NAT (TURN) Extensions  Copyright © 2021 Microsoft Corporation |
|  |  |

## Таблица - Инструменты

|  |  |
| --- | --- |
| URI | Описание |
| https://webrtc.github.io/samples/src/content/peerconnection/trickle-ice/ | Тестирование Turn серверов |

## Обсуждения репозитория

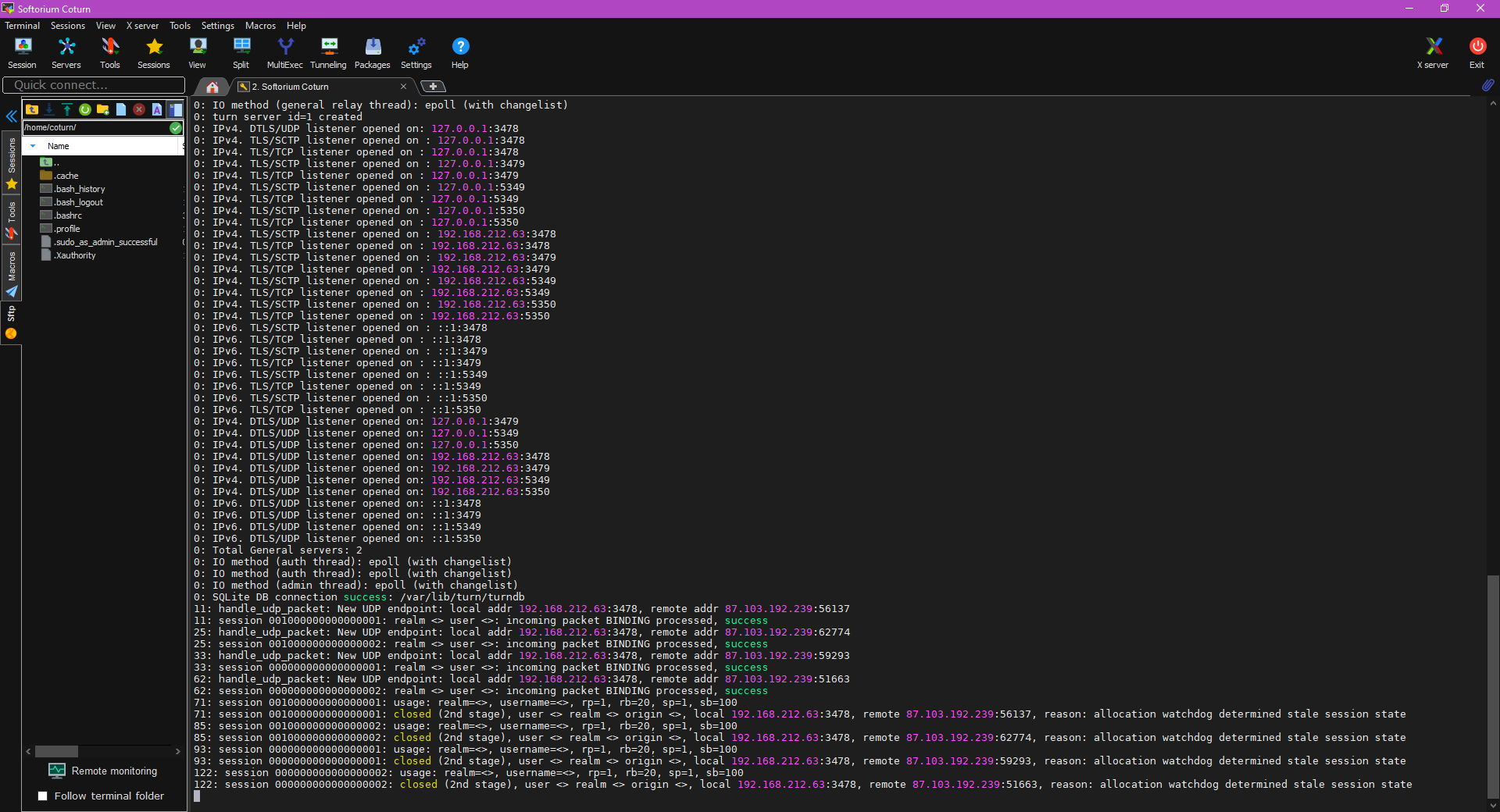
|  |  |
| --- | --- |
|  |  |
| <https://github.com/coturn/coturn/issues/399> | TURN server doesn't response TCP relay candidate |
| <https://github.com/coturn/coturn/issues/543> | Unable to connect to TURN server |
| <https://github.com/coturn/coturn/issues/364> | Clarify docs about auth methods |
|  |  |

## Source Code

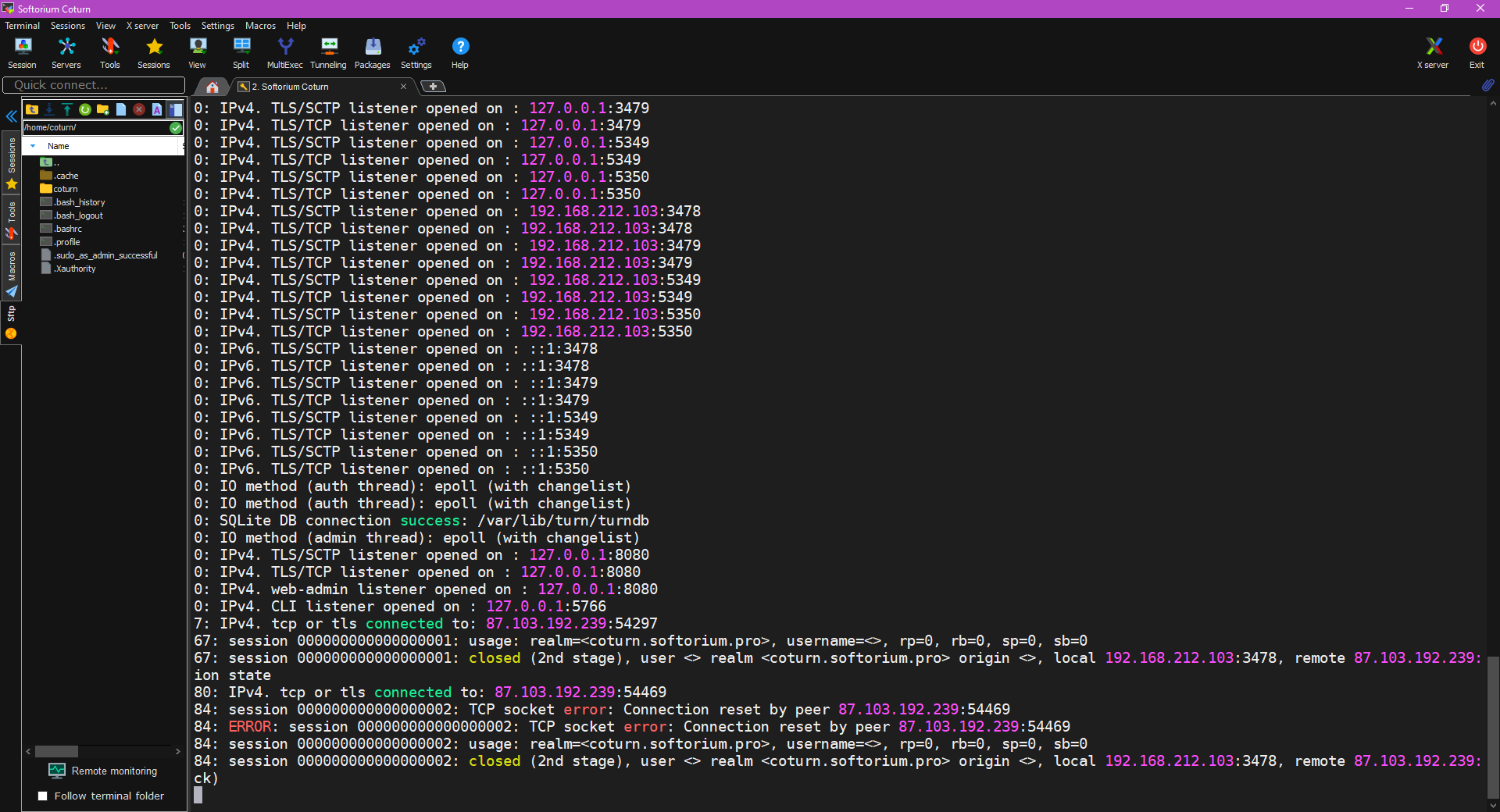
|  |  |  |
| --- | --- | --- |
| Uri | Search substring | Description |
| <https://github.com/coturn/coturn/blob/6492f51a63501162748ab055015ff0da473f44f0/src/server/ns_turn_server.c> | check\_stun\_auth |  |

# Приложения

Пример обращения с <https://webrtc.github.io/samples/src/content/peerconnection/trickle-ice/>



Разработка: Первое подключение. С командой Bind. Отключение. Повторное подключение. TransanctionID разные. Обратить внимание на 84 строку



## Как выглядит панель админа после аллокации

