Управление удаленными устройствами - coturn, STUN, TURN ver1.3

1. Назначение протоколов

STUN (Simple Traversal of UDP through NAT) и **TURN (Traversal Using Relays around NAT)** - протоколы обеспечения связности между устройствами для передачи потоковых данных WebRTC в глобальных и локальных сетях, использующих различные типы NAT и Firewall.
В продукте SM вышеуказанные протоколы служат для организации каналов управления удаленными устройствами в реальном времени и трансляции экрана.

2. Реализация

Для реализации протоколов STUN/TURN рекомендуется использовать OpenSource проект - `Coturn` (https://github.com/coturn/coturn). Важно отметить, что данное решение не разрабатывается командой SM и не является частью продукта. Поддержка возможна только в части решения проблем с установкой и настройкой.

3. Требования к ресурсам

3.1. Требования к аппаратным ресурсам

При любом варианте развертывания `coturn` требует для корректной работы следующие ресурсы:
CPU - 2x vCpu
RAM - 4GB
HDD - 15GB
Network - 200 МБит/с для 50 одновременно активных сессий (из расчета 4-5 мбит/с на активного клиента).
OS: Linux

3.2. Требования к сетевым доступам

Для корректной работы компонент `coturn` должен располагаться в DMZ и иметь внешний маршрутизируемый IP, либо должен быть настроен DNAT по указанным ниже портам. К компоненту должен быть обеспечен сетевой доступ как со стороны APM Администратора (ПК, на котором открывается пользовательский интерфейс APM в веб-браузере), так и со стороны управляемых мобильных устройств.

Сетевые порты, необходимые для работы STUN/TURN:

- **3478:tcp/udp** - STUN-сервер;
- **49152-65535:udp** - TURN-сервер;

Требования должны быть учтены при планировании мощностей и архитектуры.

4. Развертывание

Существует два варианта развертывания:

- В docker-контейнере
- Нативным пакетом операционной системы

Установка может быть выполнена как на одной машине с продуктом SM, так и на выделенном сервере (рекомендуется).
При любом варианте установке версия `coturn` должна быть не ниже 4.6.1

Все указанные ниже примеры команд относятся к операционным системам Debian 13 / Ubuntu 24.04. Для других дистрибутивов Linux они могут отличаться.

4.1. Настройка Firewall

В обязательном порядке перед развертыванием `coturn` необходимо убедиться, что используемые порты (**п 3.2**) не блокируются firewall сервера и/или вышестоящим сетевым оборудованием.
В особенности это актуально при установке на один сервер с продуктом SM, в котором firewall настраивается автоматически. В этом случае необходимо вручную через `ufw` (предпочтительно) или `iptables` открыть необходимые порты:

UFW:

```
sudo ufw allow from any to any port 3478 proto udp
sudo ufw allow from any to any port 3478 proto tcp
sudo ufw allow from any to any port 49152:65535 proto udp
sudo ufw reload
```

IPTables:

```
sudo iptables -A INPUT -p tcp --dport 3478 -j ACCEPT
sudo iptables -A INPUT -p udp --dport 3478 -j ACCEPT
sudo iptables -A INPUT -p udp --dport 49152:65535 -j ACCEPT
```

4.2. Конфигурационный файл

Ниже пример конфигурационного файла `turnserver.conf` для варианта размещения компонента не сервере с внутренним адресом `<internal_ip>` за DNAT-ом с внешним адресом `<external_ip>`:

```
#logs
verbose

#Указывается только в случае установки нативным пакетом ОС.
log-file=/var/log/coturn/coturn.log

#network
listening-port=3478
external-ip=<external_ip>/<internal_ip>
min-port=49152
max-port=65535

#realm
realm=<domain>

#auth
lt-cred-mech
use-auth-secret
static-auth-secret=<auth-secret>


#other
no-rfc5780
no-stun-backward-compatibility
response-origin-only-with-rfc5780
```

Где:

- `<auth-secret>` - парольная фраза для аутентификации клиентов. Ее необходимо сгенерировать самостоятельно при формировании конфигурационного файла. Это же значение необходимо будет добавить в конфигурационный файл ARM `arm.yml` в секции настроек turn-сервера.

- `<domain>` - домен, к которому привязан внешний адрес компонента или DNAT. Если предполагается использование без домена, то в данном параметре необходимо указать внешний IP-адрес.
- `<external_ip>` - внешний адрес сервера в случае, если он непосредственно подключен к интернету интерфейсом. Адрес NAT - если сервер подключен к интернету через NAT.
- `<internal_ip>` - заполняется только в случае, если сервер подключен к интернету через NAT. Указывается адрес сервера в локальной сети.

4.3. Установка в docker-контейнере

1. Установить `docker` и `docker-compose` последних доступных версий из официальных репозиториев `docker` https://docs.docker.com/engine/install/ . При невозможности использовать внешние репозитории, необходимо чтобы версии компонентов были не ниже:

```
docker >= 25
docker compose >= 2.27
```

2. Создать отдельную директорию для размещения конфигурационных файлов, например `/opt/coturn` , и перейти в нее.
3. Разместить `docker-compose.yml` для запуска приложения. Также в директории `/opt/coturn/config` необходимо разместить файл конфигурации `turnserver.conf` , предварительно подготовив его согласно **п. 4.2**.

```
services:
  coturn:
    image: coturn/coturn:latest
    restart: always
    command: "turnserver -c /etc/turnserver.conf"
    volumes:
      - ./config/turnserver.conf:/etc/turnserver.conf
    network_mode: "host"
    logging:
      driver: json-file
      options:
        max-size: 10m
        max-file: 3
```

Для запуска контейнера необходимо в директории с файлом `docker-compose.yml` выполнить команду:

```
docker compose up -d
```

Статус контейнера можно узнать через команду:

```
docker ps -a
```

4.4 Установка нативным пакетом операционной системы

Ниже пример для ОС Ubuntu Linux и debian-based систем:

```
sudo apt update && apt install coturn
```

Статус сервиса можно узнать через команду:

```
systemctl status coturn
```

Далее необходимо создать директорию для log-файлов и выдать ей соответствующие разрешения. После чего перезапустить сервис:

```
sudo mkdir /var/log/coturn
sudo chown turnserver:root /var/log/coturn
sudo systemctl restart coturn
```

При редактировании или замене конфигурационного файла (по умолчанию находится `/etc/turnserver.conf`) необходимо сохранять исходные права доступа к нему. В противном случае сервис не сможет его прочитать и будет некорректно работать.
Права должны быть следующими:

```
-rw-r----- 1 root turnserver 29K Apr  1  2024 /etc/turnserver.conf
```

5. Проблемы и диагностика

5.1 Наиболее частые проблемы

Если сервис `coturn` работает некорректно или запускается с ошибками, необходимо проверить:

1. Права доступа к конфигурационному файлу turnserver.conf. Они должны соответствовать указанным в **п. 4.4** для установки нативным пакетом ОС. Для установки docker необходимо проверить, что разрешения не ниже `644`.
2. Расположение конфигурационного файла. Для нативного пакета оно должно соответствовать пути, указанном в `systemd unit`. Проверить можно командой `cat /usr/lib/systemd/system/coturn.service | grep 'ExecStart'`. Для установки в docker местоположение должно соответствовать указанному в **п 4.3.**
3. Соответствие параметра `static-auth-secret` в конфигурационном файле `turnserver.conf` и конфигурационном файле АРМ `arm.yml`.
4. Соответствие адреса и порта turn-сервера фактическому в настройках в интерфейсе АРМ.
5. Настройки firewall на сервере `coturn` и вышестоящем сетевом оборудовании.
6. Сетевая доступность с рабочего места, на котором в браузере запущен АРМ администратора и мобильных устройств.

5.2 Диагностика

Для проверки работоспособности `coturn` в составе дистрибутива имеется тестовая утилита `turnutils_uclient`. При нативной установке пакетом она идет в комплекте с самим сервисом. При установке в docker, утилиту можно развернуть на любой другой машине, находящейся в нужной сети.

Утилита эмулирует подключение устройств и проверяет надежность установленного туннеля. Пример:

```
turnutils_uclient -t -T -y -v -W <auth-secret> <external_ip_or_fqdn>
```

При успешном подключении в конце вывода команды отобразится статус проверки:

```
1: : start_mclient: msz=2, tot_send_msgs=0, tot_recv_msgs=0, tot_send_bytes ~ 0,
tot_recv_bytes ~ 0
2: : start_mclient: msz=2, tot_send_msgs=0, tot_recv_msgs=0, tot_send_bytes ~ 0,
tot_recv_bytes ~ 0
3: : start_mclient: msz=2, tot_send_msgs=5, tot_recv_msgs=5, tot_send_bytes ~ 500,
tot_recv_bytes ~ 500
3: : done, connection 0x78dc3a9df010 closed.
3: : done, connection 0x78dc3a9be010 closed.
4: : start_mclient: tot_send_msgs=10, tot_recv_msgs=10
4: : start_mclient: tot_send_bytes ~ 1000, tot_recv_bytes ~ 1000
4: : Total transmit time is 4
4: : Total lost packets 0 (0.000000%), total send dropped 0 (0.000000%)
```

```
4: : Average round trip delay 42.800000 ms; min = 29 ms, max = 59 ms
4: : Average jitter 9.800000 ms; min = 9 ms, max = 20 ms
```

При неуспешном отобразится ошибка с подробностями.

5.3 Подготовка обращения в техническую поддержку SM

Если самостоятельно решить проблему не получается, необходимо собрать диагностические данные и отправить в техническую поддержку SM.
Ниже перечень необходимых данных:

Запрос	**Отчет о выполнении**
Проверить что в подключениях к серверам указан адрес turn	Прислать скриншот
Проверить доступ с машины консоли администрирования и с устройства(нужно использовать стороннее ПО) до сервера turn по 3478 порту telnet имя сервера 3478	Да/нет
Проверить совпадение пароля в /opt/emm/config/arm.yml и на сервере turn в файле настроек /etc/turnserver.conf	Да/нет
Проверить права файла /etc/turnserver.conf -rw-r----- 1 root turnserver или /opt/coturn/config/turnserver.conf -rw-r--r-- 1 root	прислать скриншот и путь к файлу
Как настраивался сервер turn -- докером или нативно. Если докером проверить что настройки прописаны в файле /opt/coturn/config/turnserver.conf	Прислать путь к файлу настроек
Прилать ответ systemctl status coturn или docker ps -a	Прислать скриншот
Прислать лог из каталога /var/log/coturn.log или cd /opt/coturn && docker-compose logs -t coturn > /tmp/coturn.log	Лог
Прислать ответ команды ss -tulnp	Прислать скриншот
Предоставляются ли разрешения в момент включения Удаленного управлния	Прислать скриншот с устройства