

<https://serveradmin.ru/audio/nastroyka-servera-telefonii-asterisk-s-nulya.ogg>

Одним из рабочих инструментов офиса, несмотря на стремительные изменения последних десятилетий, по-прежнему является телефон. Мы займемся пошаговой настройкой с нуля ATC asterisk - современного инструмента для организации телефонии в офисе на основе протокола SIP. Я подробно с примерами и описанием проведу вас по основным параметрам, необходимым для базового функционала.

Если у вас есть желание научиться строить и поддерживать высокодоступные и надежные системы, рекомендую познакомиться с онлайн-курсом **«Administrator Linux. Professional»** в OTUS. Курс не для новичков, для поступления нужно пройти вступительный тест.

## Содержание

Введение

Быстрая установка из репозитория

Centos 7

Установка из исходников

Centos 8

Centos 7

Настройка iptables, asterisk за NAT, проброс портов

Подключение абонентов и проверка внутренних звонков

Настройка sip trunk (транка) и добавление номера

Dial-plan - пример маршрутизации звонков

Приветствие и голосовое меню (ivr)

Включаем голосовую почту

Очереди (queues) входящих звонков

Учет и просмотр статистики звонков (cdr viewer)

Запись (record) разговоров

Музыка на ожидании (on hold)

Настройка конференций

Перевод, перехват, переадресация звонка

Защита asterisk с помощью fail2ban

Заключение

Помогла статья? Подписывайся на telegram канал автора

Данная статья является частью единого цикла статей про сервер Centos.

## Введение

Не буду останавливаться на описании сервера asterisk, в интернете много информации на эту тему. Да и сам я кратко рассказывал в своих предыдущих статьях про установку. Материал будет объемный, поэтому сразу перейдем к сути.

Статья планируется учебная, поэтому использовать будем голый asterisk без каких-либо web панелей для управления. Выполнять настройку asterisk будем с нуля, то есть с самой начальной установки самого сервера телефонии и всех зависимостей. Если вы разберетесь и освоите этот материал, то потом без проблем сможете настроить и поддерживать любую конфигурацию на основе астериска.

Я хочу подробно рассказать о настройке asterisk на конкретном примере, где будет собран в одном месте расширенный функционал, отвечающий практически на все запросы среднестатистического офиса.

Сразу хочу обратить внимание на очень важный момент. Я не являюсь профессионалом в настройке asterisk. Данный материал является калькуляцией всех моих знаний на текущий момент, которые я методично шаг за шагом собирал из доступных в интернете источников. По этой статье вы получите рабочий, многофункциональный сервер телефонии, работу которого я проверял на практике. Но это не означает, что все, что здесь настроено, сделано наилучшим образом. Я постоянно учусь и совершенствую свои знания. Если вы увидите ошибки, недочеты, варианты более правильной и удобной настройки, прошу об этом сообщить в комментариях, я проверю и обновлю статью.

Для примера опишем наш воображаемый офис:

- Работает 30 сотрудников. Номера будут трехзначные, от 100 до 130.
- У нас будут 3 отдела - менеджеры, техподдержка, руководство и все остальные.
- Номер секретаря 100, менеджеры 101-110, техподдержка 111-120, руководство 121-130.
- Мы будем использовать одного SIP провайдера для звонков.

Кратко получается такая картина. Я не пишу в самом начале о техническом задании, которое буду реализовывать. Каждый момент буду описывать и раскрывать в соответствующем разделе.

Для настройки я буду использовать учетную запись сервиса [zadarma.com](https://zadarma.com). Неплохой оператор voip связи. У меня их номер в аренде для этого сайта в контактах. Беру его для примера, потому что удобно использовать для тестирования конфигурации. Сразу после регистрации вам дают аккаунт, пример настроек для asterisk. Вы можете позвонить на прямой городской номер, ввести добавочный и совершить звонок на свой аккаунт. Это полностью эмулирует работу sip подключения от какого-нибудь провайдера.

В качестве операционной системы у меня выступает CentOS 8. Но для данной статьи это не имеет принципиального значения. Конфигурация астериск кроссплатформенная, без проблем переносится между системами.

Приступаем к нашей работе по настройке сервера телефонии.

## Быстрая установка из репозитория

### Centos 7

У asterisk не существует официального репозитория пакетов, поэтому предпочтительным способом установки свежей версии является сборка из исходников. Но если вы хотите быстро установить и потестить систему, то можно использовать один из сторонних репозиториев asterisk. К примеру - **Tuncy**. Я не знаю, кто ведет этот репозиторий, как часто он обновляется и насколько там актуальные версии. На момент актуализации этой статьи (март 2020 года) в репозитории не было свежих пакетов для Centos 8, только для 7-й.

Добавляем репозиторий астериск в систему. Для этого создаем файл `/etc/yum.repos.d/tuncy-asterisk-16.repo` следующего содержания.

```
[asterisk-common]
name=Asterisk Common Requirement Packages @ tucny.com
baseurl=https://ast.tucny.com/repo/asterisk-common/el${releasever}/${basearch}/
enabled=1
gpgcheck=1
gpgkey=https://ast.tucny.com/repo/RPM-GPG-KEY-dtucny

[asterisk-16]
```

```
name=Asterisk 16 Packages @ tucny.com
baseurl=https://ast.tucny.com/repo/asterisk-16/el$releasever/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://ast.tucny.com/repo/RPM-GPG-KEY-dtucny
```

Обновляем информацию о репозиториях и устанавливаем астериск.

```
# yum install asterisk
```

```
Dependencies Resolved

=====
Package                Arch                Version              Repository            Size
=====
Installing:
asterisk                x86_64              13.22.0-2.e17       asterisk-13          4.4 M
Installing for dependencies:
gmime                   x86_64              2.6.23-1.e17        epel                  193 k
gsm                     x86_64              1.0.13-11.e17       base                  30 k
ilbc                    x86_64              1.1.1-4.e17         epel                  57 k
libogg                  x86_64              2:1.3.0-7.e17       base                  24 k
libsrtplib              x86_64              1.4.4-10.20101004cvs.e17 base                  275 k
libvorbis                x86_64              1:1.3.3-8.e17.1     updates               204 k
speex                   x86_64              1.2-0.19.rc1.e17    base                  98 k

Transaction Summary
=====
Install 1 Package (+7 Dependent packages)

Total download size: 5.3 M
Installed size: 20 M
Is this ok [y/d/N]: y
```

serveradmin.ru

## Установка из исходников

### Centos 8

Вопроса установки asterisk я уже касался ранее в одной из прошлых статей. Но там я использовал связку с панелью управления freepbx. Здесь же мы будем использовать голый астериск, без обвязок. Более того, я не буду использовать никаких дополнительных плат расширения и модемов. Будет только софтовая АТС, которая легко переносится с одного сервера на другой при желании. Считаю, что такой подход наиболее эффективен и к нему стоит стремиться. Настроив виртуальную машину, вы навсегда будете отвязаны от конкретного железа и спокойно можете переносить свой сервер куда угодно, заменив только сетевые настройки.

Таким образом, нам нужно установить непосредственно asterisk и pjproject с jansson. На первоначальном этапе этого достаточно. Если вы предпочитаете сервер debian, то воспользуйтесь отдельной инструкцией по установке asterisk 16 на debian 10. После этого можете сразу же переходить на следующий этап настройки. Приступим.

Первым делом обновляем систему и отключаем SELinux, как рассказано в статье про настройку centos. Установим теперь пакеты, которые нам понадобятся для сборки. В первую очередь подключим репозиторий epel.

```
# dnf install epel-release
```

Дальше идет мета пакет **Development Tools** со всем необходимым для сборки из исходников.

```
# dnf groupinstall "Development Tools"
```

```
[root@centos8 ~]# dnf groupinstall "Development Tools"
Last metadata expiration check: 0:00:38 ago on Thu 24 Oct 2019 11:52:17 AM MSK.
Dependencies resolved.
=====
Package                               Arch                               Version                               Repository                               Size
=====
Installing group/module packages:
asciidoc                               noarch                             8.6.10-0.5.20180627gitf7c2274.e18     AppStream                               216 k
autoconf                               noarch                             2.69-27.e18                             AppStream                               710 k
automake                               noarch                             1.16.1-6.e18                             AppStream                               713 k
bison                                   x86_64                             3.0.4-10.e18                             AppStream                               688 k
byacc                                   x86_64                             1.9.20170709-4.e18                       AppStream                               91 k
ctags                                   x86_64                             5.8-22.e18                               AppStream                               170 k
diffstat                               x86_64                             1.61-7.e18                               AppStream                               44 k
flex                                    x86_64                             2.6.1-9.e18                               AppStream                               320 k
gcc                                      x86_64                             8.2.1-3.5.e18                             AppStream                               23 M
gcc-c++                                x86_64                             8.2.1-3.5.e18                             AppStream                               12 M
gdb                                      x86_64                             8.2-6.e18_0                               AppStream                               296 k
git                                     x86_64                             2.18.1-3.e18                             AppStream                               186 k
intltool                               noarch                             0.51.0-11.e18                             AppStream                               66 k
libtool                                x86_64                             2.4.6-25.e18                             AppStream                               709 k
ltrace                                 x86_64                             0.7.91-27.e18                             AppStream                               160 k
patchutils                             x86_64                             0.3.4-10.e18                             AppStream                               116 k
perl-Fedora-VSP                         noarch                             0.001-9.e18                              AppStream                               24 k
perl-generators                         noarch                             1.10-7.e18                              AppStream                               18 k
pesign                                   x86_64                             0.112-25.e18                             AppStream                               181 k
redhat-rpm-config                       noarch                             116-1.e18_0.1                             AppStream                               82 k
rpm-build                               x86_64                             4.14.2-11.e18_0                           AppStream                               166 k
source-highlight                       x86_64                             3.1.8-16.e18                             AppStream                               661 k
systemtap                               x86_64                             4.0-7.e18                                AppStream                               17 k
valgrind                                x86_64                             1:3.14.0-10.e18                           AppStream                               11 M
valgrind-devel                          x86_64                             1:3.14.0-10.e18                           AppStream                               89 k
=====
```

И еще некоторые зависимости, которые будут нужны.

```
# dnf install git wget net-tools sqlite-devel psmisc ncurses-devel libtermcap-devel newt-devel libxml2-devel libtiff-
devel gtk2-devel libtool libuuid-devel subversion kernel-devel kernel-devel-$(uname -r) crontabs cronie-anacron mariadb
mariadb-server
```

```
[root@centos8 ~]# dnf install git wget net-tools sqlite-devel psmisc ncurses-devel libtermcap-devel newt-devel libxml2-devel libtiff-devel gtk2-devel libtool libuuid-devel subversion kernel-devel kernel-devel-$(uname -r) crontabs cronie-anacron mariadb mariadb-server
Last metadata expiration check: 0:04:06 ago on Thu 24 Oct 2019 11:53:29 AM MSK.
Package git-2.18.1-3.el8.x86_64 is already installed.
Package wget-1.19.5-7.el8_0.1.x86_64 is already installed.
Package net-tools-2.0-0.51.20160912git.el8.x86_64 is already installed.
Package libtool-2.4.6-25.el8.x86_64 is already installed.
Package kernel-devel-4.18.0-80.11.2.el8_0.x86_64 is already installed.
Package kernel-devel-4.18.0-80.11.2.el8_0.x86_64 is already installed.
Package crontabs-1.11-16.20150630git.el8.noarch is already installed.
Package cronie-anacron-1.5.2-2.el8.x86_64 is already installed.
Dependencies resolved.
=====
Package Arch Version Repository Size
=====
Installing:
gtk2-devel x86_64 2.24.32-4.el8 AppStream 3.0 M
libtiff-devel x86_64 4.0.9-13.el8 AppStream 511 k
libxml2-devel x86_64 2.9.7-5.el8 AppStream 1.0 M
mariadb x86_64 3:10.3.11-2.module_el8.0.0+35+6f2527ed AppStream 6.2 M
mariadb-server x86_64 3:10.3.11-2.module_el8.0.0+35+6f2527ed AppStream 16 M
newt-devel x86_64 0.52.20-9.el8 AppStream 59 k
subversion x86_64 1.10.2-1.module_el8.0.0+45+75bba4f4 AppStream 1.1 M
libuuid-devel x86_64 2.32.1-8.el8 BaseOS 94 k
ncurses-devel x86_64 6.1-7.20180224.el8 BaseOS 527 k
psmisc x86_64 23.1-3.el8 BaseOS 151 k
sqlite-devel x86_64 3.26.0-3.el8 BaseOS 163 k
Installing dependencies:
apr x86_64 1.6.3-9.el8 AppStream 125 k
apr-util x86_64 1.6.1-6.el8 AppStream 105 k
atk-devel x86_64 2.28.1-1.el8 AppStream 192 k
cairo-devel x86_64 1.15.12-3.el8 AppStream 272 k
cmake-filesystem x86_64 3.11.4-3.el8 AppStream 40 k
fribidi-devel x86_64 1.0.4-6.el8 AppStream 63 k
```

Настройте mysql сервер, задав пароль для root.

```
# systemctl start mariadb
# systemctl enable mariadb
```

```
# /usr/bin/mysql_secure_installation
```

На этом подготовка закончена. Устанавливаем Jansson и pjsip.

```
# cd ~  
# git clone https://github.com/akheron/jansson.git  
# cd jansson  
# autoreconf -i  
# ./configure --prefix=/usr/  
# make && make install
```



```
make[4]: Leaving directory '/root/jansson/test/suites'
make[3]: Leaving directory '/root/jansson/test/suites'
make[2]: Leaving directory '/root/jansson/test/suites'
Making install in ossfuzz
make[2]: Entering directory '/root/jansson/test/ossfuzz'
make[3]: Entering directory '/root/jansson/test/ossfuzz'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/root/jansson/test/ossfuzz'
make[2]: Leaving directory '/root/jansson/test/ossfuzz'
make[2]: Entering directory '/root/jansson/test'
make[3]: Entering directory '/root/jansson/test'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/root/jansson/test'
make[2]: Leaving directory '/root/jansson/test'
make[1]: Leaving directory '/root/jansson/test'
make[1]: Entering directory '/root/jansson'
make[2]: Entering directory '/root/jansson'
make[2]: Nothing to be done for 'install-exec-am'.
/usr/bin/mkdir -p '/usr/lib64/pkgconfig'
/usr/bin/install -c -m 644 jansson.pc '/usr/lib64/pkgconfig'
make[2]: Leaving directory '/root/jansson'
make[1]: Leaving directory '/root/jansson'
[root@centos8 jansson]#
```

```
# cd ~
# git clone https://github.com/pjsip/pjproject.git
# cd pjproject
# ./configure CFLAGS="-DNDEBUG -DPJ_HAS_IPV6=1" --prefix=/usr --libdir=/usr/lib64 --enable-shared --disable-video --
disable-sound --disable-opencore-amr
```

```
# make dep && make && make install  
# ldconfig
```

```
lib/libilbccodec.so /root/pjproject/third_party/lib/libg722lcodec-x86_64-unknown-linux-gnu.a /root/pjproject/third_party/lib/libg722lcodec.so.2 /root/pjproject/third_party/lib/libg722lcodec.so /root/pjproject/third_party/lib/libwebrtc-x86_64-unknown-linux-gnu.a /root/pjproject/third_party/lib/libwebrtc.so.2 /root/pjproject/third_party/lib/libwebrtc.so /root/pjproject/pjlib/lib/libpj-x86_64-unknown-linux-gnu.a /root/pjproject/pjsip/lib/libpjsua2.so.2 /root/pjproject/pjsip/lib/libpjsua2.so /root/pjproject/pjsip/lib/libpjsua.so.2 /root/pjproject/pjsip/lib/libpjsua.so /root/pjproject/pjsip/lib/libpjsip-ua.so.2 /root/pjproject/pjsip/lib/libpjsip-ua.so /root/pjproject/pjsip/lib/libpjsip-simple.so.2 /root/pjproject/pjsip/lib/libpjsip-simple.so /root/pjproject/pjsip/lib/libpjsip.so.2 /root/pjproject/pjsip/lib/libpjsip.so /root/pjproject/pjmedia/lib/libpjmedia-codec.so.2 /root/pjproject/pjmedia/lib/libpjmedia-codec.so /root/pjproject/pjmedia/lib/libpjmedia-videodev.so.2 /root/pjproject/pjmedia/lib/libpjmedia-videodev.so /root/pjproject/pjmedia/lib/libpjmedia.so.2 /root/pjproject/pjmedia/lib/libpjmedia-audiodev.so.2 /root/pjproject/pjmedia/lib/libpjmedia-audiodev.so /root/pjproject/pjnath/lib/libpjnath.so.2 /root/pjproject/pjnath/lib/libpjnath.so /root/pjproject/pjlib-util/lib/libpjlib-util.so.2 /root/pjproject/pjlib-util/lib/libpjlib-util.so /root/pjproject/pjlib/lib/libpj.so.2 /root/pjproject/pjlib/lib/libpj.so /usr/lib64/  
mkdir -p /usr/include/  
for d in pjlib pjlib-util pjnath pjmedia pjsip; do \  
    cp -RLf $d/include/* /usr/include/; \  
done  
mkdir -p /usr/lib64/pkgconfig  
sed -e "s!@PREFIX@!/usr!" libpjproject.pc.in | \  
    sed -e "s!@INCLUDEDIR@!/usr/include!" | \  
    sed -e "s!@LIBDIR@!/usr/lib64!" | \  
    sed -e "s!@PJ_VERSION@/2.9-svn/" | \  
    sed -e "s!@PJ_LDLIBS@!!" | \  
    sed -e "s!@PJ_LDXXLIBS@!-lpjsua2 -lstdc++ -lpjsua -lpjsip-ua -lpjsip-simple -lpjsip -lpjmedia-codec -lpjmedia -lpjmedia-videodev -lpjmedia-audiodev -lpjmedia -lpjnath -lpjlib-util -lstrtp -lresample -lgsmcodec -lspeex -libccodec -lg722lcodec -lwebrtc -lpj -luuid -lm -lrt -lpthread !" | \  
    sed -e "s!@PJ_INSTALL_CFLAGS@!!" | \  
    sed -e "s!@PJ_INSTALL_CXXFLAGS@!-g -O2 -I/usr/include -DPJ_AUTOCONF=1 -DNDEBUG -DPJ_HAS_IPV6=1 -DPJ_IS_BIG_ENDIAN=0 -DPJ_IS_LITTLE_ENDIAN=1 -fPIC!"  
> //usr/lib64/pkgconfig/libpjproject.pc  
[root@centos8 pjproject]# ldconfig  
[root@centos8 pjproject]#
```

serveradmin.ru

Все готово к установке непосредственно Asterisk

Я буду устанавливать LTS версию Asterisk 16. Советую для долгосрочного использования всегда использовать LTS версии. Они в целом стабильнее и дольше срок поддержки. Идем на страницу <https://www.asterisk.org/downloads/asterisk/all-asterisk-versions> и копируем ссылку на нужную версию. Загружаем ее на сервер.

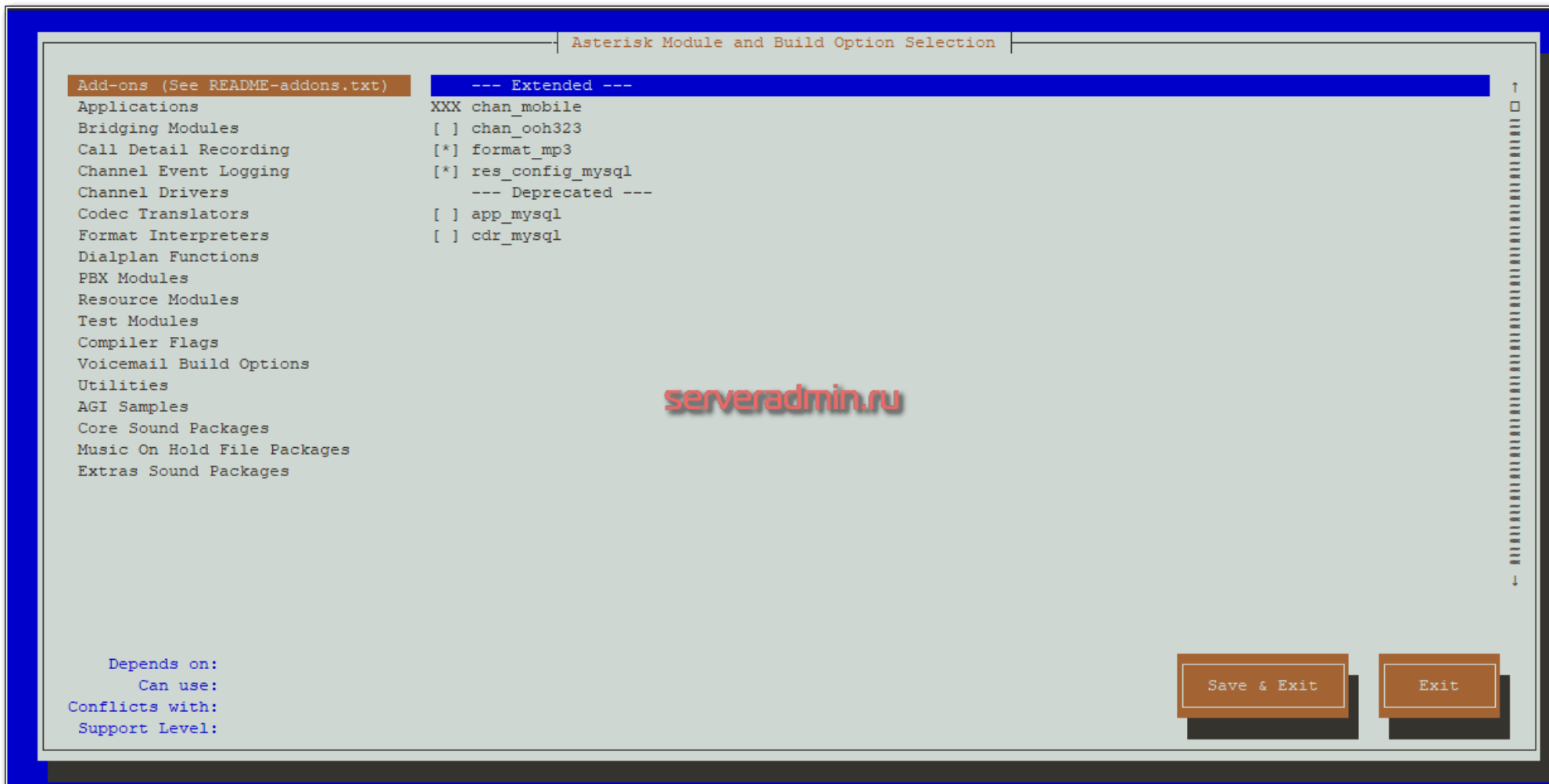
```
# cd ~
# wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-16-current.tar.gz
# tar xzf asterisk-16-current.tar.gz
# cd asterisk-16*/
# contrib/scripts/install_prereq install
# contrib/scripts/get_mp3_source.sh
```

Устанавливаем на centos 8 пакет **libedit-devel**.

```
# dnf config-manager --set-enabled powertools
# dnf install libedit-devel
```

Собираем asterisk.

```
# ./configure --libdir=/usr/lib64
# make menuselect
```



Выбирайте необходимые модули и звуки, в зависимости от того, что вам нужно. Я в общем случае указываю:

- **Add-ons:** format\_mp3, res\_config\_mysql.

- **Core Sound Packages:** русские звуки RU-WAV.
- **Music On Hold File Packages:** звук WAV.
- **Extras Sound Packages:** английский EN-WAV, русского к сожалению нет.

Все остальные настройки оставляем по-умолчанию. Ставится много модулей. Все они не нужны, но мало ли, пригодится что-то в будущем. Неиспользуемые модули можно будет потом отключить в конфигурации.

Продолжаем установку:

```
# make && make install && make samples && make config  
# ldconfig
```

По-умолчанию, asterisk установлен от root и будет запускаться от него же. Я предлагаю для этого создать отдельного пользователя и запускать астериск от него. Для этого создаем пользователя и добавляем его в некоторые группы.

```
# groupadd asterisk  
# useradd -r -d /var/lib/asterisk -g asterisk asterisk  
# usermod -aG audio,dialout asterisk  
# chown -R asterisk.asterisk /etc/asterisk /var/{lib,log,spool}/asterisk /usr/lib64/asterisk
```

Настраиваем Asterisk на запуск под этим пользователем. Для этого добавляем в конфиг `/etc/sysconfig/asterisk` параметры:

```
AST_USER="asterisk"  
AST_GROUP="asterisk"
```

Теперь добавим примерно то же самое в сам конфиг астера `/etc/asterisk/asterisk.conf`.

```
runuser = asterisk  
rungroup = asterisk
```

Пробуем запустить asterisk:

```
# systemctl start asterisk
```

Если нет сообщений об ошибке, скорее всего все в порядке. Проверяем статус службы.

```
# systemctl status asterisk
```

```
[root@centos8 asterisk-16.6.1]# systemctl status asterisk
• asterisk.service - LSB: Asterisk PBX
  Loaded: loaded (/etc/rc.d/init.d/asterisk; generated)
  Active: active (running) since Thu 2019-10-24 14:45:47 MSK; 46s ago
  Docs: man:systemd-sysv-generator(8)
  Process: 6309 ExecStart=/etc/rc.d/init.d/asterisk start (code=exited, status=0/SUCCESS)
  Main PID: 6340 (asterisk)
  Tasks: 74 (limit: 25039)
  Memory: 39.9M
  CGroup: /system.slice/asterisk.service
          └─6338 /bin/sh /usr/sbin/safe_asterisk
            └─6340 /usr/sbin/asterisk -f -vvvg -c

Oct 24 14:45:47 centos8 systemd[1]: Starting LSB: Asterisk PBX...
Oct 24 14:45:47 centos8 asterisk[6309]: Starting asterisk:
Oct 24 14:45:47 centos8 systemd[1]: asterisk.service: Can't open PID file /var/run/asterisk/asterisk.pid (yet?) after start: No such file or directory
Oct 24 14:45:47 centos8 systemd[1]: asterisk.service: Supervising process 6340 which is not our child. We'll most likely not notice when it exits.
Oct 24 14:45:47 centos8 systemd[1]: Started LSB: Asterisk PBX.
Oct 24 14:45:47 centos8 asterisk[6340]: radcli: rc_read_config: rc_read_config: can't open /etc/radiusclient-ng/radiusclient.conf: No such file or directory
Oct 24 14:45:47 centos8 asterisk[6340]: radcli: rc_read_config: rc_read_config: can't open /etc/radiusclient-ng/radiusclient.conf: No such file or directory
lines 1-19/19 (END)
```

serveradmin.ru

Asterisk запустился, но есть небольшие ошибки.

```
radcli: rc_read_config: rc_read_config: can't open /etc/radiusclient-ng/radiusclient.conf: No such file or directory
```

Связаны с тем, что в конфигах неверно указан путь к radiusclient. Сейчас исправим это.

```
# sed -i 's";\[radius\]"\[radius\]"g' /etc/asterisk/cdr.conf
# sed -i 's";radiuscfg => /usr/local/etc/radiusclient-ng/radiusclient.conf"radiuscfg => /etc/radcli/radiusclient.conf"
/etc/asterisk/cdr.conf
# sed -i 's";radiuscfg => /usr/local/etc/radiusclient-ng/radiusclient.conf"radiuscfg => /etc/radcli/radiusclient.conf"
/etc/asterisk/cel.conf
```

Перезапускаем asterisk и убеждаемся, что ошибок нет. Проверим, все ли в порядке, зайдя в консоль:

```
# asterisk -r
```

```
[root@centos8 asterisk-16.6.1]# rasterisk
Asterisk 16.6.1, Copyright (C) 1999 - 2018, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Running as user 'asterisk'
Running under group 'asterisk'
Connected to Asterisk 16.6.1 currently running on centos8 (pid = 6654)
centos8*CLI>
```

Если получили такой же вывод

команды, значит все в порядке, астериск 16 установлен. Добавим его теперь в автозагрузку.

```
# systemctl enable asterisk
```

Для тех, у кого что-то не получается или не понятно, как сделать, записал видео по приведенной инструкции. Видео подтверждает, что материал актуален и если делать по нему, то все получится. <https://youtu.be/9q9RGqBVQbA>

## Centos 7

Если вы хотите использовать для настройки Asterisk предыдущую версию системы, то смотрите мою статью по установке Asterisk 16 на Centos 7. Там нет принципиальных отличий, но имейте ввиду, что все дальнейшие действия были проделаны и проверены на 8-й версии Центос, так что я не ручаюсь, что это так же заработает и на 7-й. Полную проверку и адаптацию материала я сделал для CentOS 8.



## Настройка iptables, asterisk за NAT, проброс портов

Сразу же уделим внимание настройке iptables для работы астериск. У нас может быть 2 ситуации, которые требуют двух принципиально различных настроек:

1. Сервер телефонии имеет свой внешний ip адрес и напрямую смотрит через него в интернет.
2. Сервер стоит за шлюзом, не имеет своего внешнего адреса, доступ в интернет с помощью NAT.

В первом случае нам нужно открыть на iptables необходимые порты для работы, все остальное закрыть. Подробно вопрос настройки iptables я рассматривал в отдельной статье. Там есть примеры и пояснения, рассказан мой подход к настройке.

Я не могу привести универсальные настройки для всех случаев. У каждого будут свои нюансы. Кто-то, к примеру, будет пользоваться rhmtuadmin для настройки базы mysql для хранения статистики звонков. Потом эту же статистику будет просматривать через cdr viewer, установленный на веб сервере. Доступ к этому веб серверу можно открыть через внешний IP адрес, а можно только через локальную сеть. Настройки iptables в данном случае будут разные.

В моем примере сервер будет находиться в локальной сети офиса за nat. Доступ в интернет осуществляется через офисный шлюз, на котором установлены iptables. На нем будет сделан проброс необходимых портов для работы внешних телефонных аппаратов. В случае, если у вас все телефоны будут находиться в локальной сети офиса вместе с сервером телефонии, пробрасывать ничего не нужно. С сервисом zadarma все будет работать без проброса портов. Пиры зарегистрируются на внешнем сервере провайдера и этого будет достаточно для приема и совершения звонков.

С другими провайдерами этого может быть не достаточно. В общем случае для настройки asterisk за nat нужно будет на шлюзе пробросить порт 5060 и диапазон 10000:20000. По-умолчанию астериск использует UDP порты. Если вы не будете менять эти настройки, то пробрасывать нужно именно UDP.

В моем случае получается следующая картина. На шлюзе сделан проброс необходимых портов:

```
iptables -t nat -A PREROUTING -p udp --dst $WAN_IP --dport 5060 -j DNAT --to 192.168.1.25:5060
iptables -t nat -A PREROUTING -p udp --dst $WAN_IP --dport 10000:20000 -j DNAT --to 192.168.1.25
```

\$WAN\_IP      Внешний IP адрес на шлюзе

192.168.1.25 Локальный адрес сервера астериск

Настройкой фаервола на самом астериске я не хочу сейчас заниматься, статья и так масштабная получается. С этим без проблем можно разобраться с помощью моей статьи по iptables, ссылку на которую я привел в начале раздела. Нужно просто открыть указанные выше порты, и любые другие, которые будут использоваться на сервере.

Если вы делаете тестовую установку и настройку asterisk, можете совсем отключить firewalld:

```
# systemctl stop firewalld
# systemctl disable firewalld
# dnf remove firewalld
```

## Подключение абонентов и проверка внутренних звонков

Астериск у нас установлен, firewall настроен. Можно попробовать подключиться и протестировать работу АТС. Я для отладки использую бесплатную софтовую звонилку ЗСХPhone 6-й версии. Не знаю, где ее сейчас найти в интернете. У самого производителя давно уже вышли более новые и платные версии, которые работают только с его АТС. А эта версия универсальная. В ней отличный функционал, удобные настройки, есть дебаг режим с подробным логированием. Пользоваться программой удобно и приятно. Скачиваем ее у меня и устанавливаем.

Теперь нам нужно сделать некоторые общие настройки и добавить пользователей. Работать будем с файлом конфигурации `/etc/asterisk/sip.conf`. Файлы настроек астера хорошо закомментированы, но мне это мешает с ними работать. Они слишком большие и громоздкие, неудобно прокручивать вверх и вниз, поэтому я их полностью чищу и вношу только те настройки, которые мне нужны. Так удобнее и нагляднее получается. Сохраните на всякий случай куда-нибудь оригинальный файл `sip.conf` и начинайте новую настройку. Вот мой пример конфига для нашего случая:

```
[general]
;Внешний ip адрес
externaddr=212.78.136.18:5060
;Указываем использовать русскую озвучку
language=ru
context=default
allowoverlap=no
udpbindaddr=0.0.0.0
```

```
tcpenable=no
tcpbindaddr=0.0.0.0
transport=udp
srvlookup=yes
allowguest=no
limitonpeers=yes

[authentication]

;Создаем шаблон для телефонов менеджеров
[managers-phones](!)
type=friend
context=call-out
secret=123
host=dynamic
nat=no
qualify=yes
canreinvite=no
callgroup=1
pickupgroup=1
call-limit=1
dtmfmode=auto
disallow=all
allow=alaw
allow=ulaw
allow=g729
allow=g723
allow=g722

;Создаем пользователей менеджеров
[100](managers-phones)
callerid="Number 100" <100>
[101](managers-phones)
```

```
callerid="Number 101" <101>  
[102](managers-phones)  
callerid="Number 102" <102>  
[103](managers-phones)  
callerid="Number 103" <103>  
[104](managers-phones)  
callerid="Number 104" <104>  
[105](managers-phones)  
callerid="Number 105" <105>  
[106](managers-phones)  
callerid="Number 106" <106>  
[107](managers-phones)  
callerid="Number 107" <107>  
[108](managers-phones)  
callerid="Number 108" <108>  
[109](managers-phones)  
callerid="Number 109" <109>  
[110](managers-phones)  
callerid="Number 110" <110>
```

```
;Создаем шаблон для телефонов поддержки  
[support-phones](!  
type=friend  
context=call-out  
secret=456  
host=dynamic  
nat=no  
qualify=yes  
canreinvite=no  
callgroup=2  
pickupgroup=2  
call-limit=1  
dtmfmode=auto
```

```
disallow=all
allow=alaw
allow=ulaw
allow=g729
allow=g723
allow=g722

;Создаем пользователей техподдержки
[111](support-phones)
callerid="Number 111" <111>
[112](support-phones)
callerid="Number 112" <112>
[113](support-phones)
callerid="Number 113" <113>
[114](support-phones)
callerid="Number 114" <114>
[115](support-phones)
callerid="Number 115" <115>
[116](support-phones)
callerid="Number 116" <116>
[117](support-phones)
callerid="Number 117" <117>
[118](support-phones)
callerid="Number 118" <118>
[119](support-phones)
callerid="Number 119" <119>
[120](support-phones)
callerid="Number 120" <120>

;Создаем шаблон для телефонов топов
[top-phones](!)
type=friend
context=call-out
```

```
secret=789
host=dynamic
nat=no
qualify=yes
canreinvite=no
callgroup=3
pickupgroup=3
call-limit=1
dtmfmode=auto
disallow=all
allow=alaw
allow=ulaw
allow=g729
allow=g723
allow=g722

;Создаем пользователей топов
[121](top-phones)
callerid="Number 111" <121>
[122](top-phones)
callerid="Number 122" <122>
[123](top-phones)
callerid="Number 123" <123>
[124](top-phones)
callerid="Number 124" <124>
[125](top-phones)
callerid="Number 125" <125>
[126](top-phones)
callerid="Number 126" <126>
[127](top-phones)
callerid="Number 127" <127>
[128](top-phones)
callerid="Number 128" <128>
```

```
[129] (top-phones)
callerid="Number 129" <129>
[130] (top-phones)
callerid="Number 130" <130>
```

Я немного пояснил комментариями отдельные моменты. Чтобы сократить размер **sip.conf**, я использую шаблоны групп номеров, где задаю общие настройки для группы. Затем просто создаю пользователей и указываю их принадлежность к группе. Они берут все настройки этой группы. Если вам необходимо будет задать отдельные настройки для какого-то пользователя, как у меня, к примеру, **callerid**, то вы просто в его разделе указываете эти настройки.

У меня стоит один и тот же пароль для всей группы. Это удобно, если все телефоны стационарные и стоят в офисе, настраивают их только сисадмины. Делать каждому персональный пароль особого смысла нет. Если вам это не нужно, то указывайте персональный пароль для каждого пользователя. Я просто привожу примеры использования тех или иных настроек, но не призываю делать так же, как я. Во многих случаях так делать нельзя. Уточню еще некоторые нюансы.

- Параметры **callgroup** и **pickupgroup** задают группы перехвата звонков. Люди из одной группы могут перехватывать звонки друг друга. Удобно заводить в одну группу людей, сидящих в одной комнате. Так они видят, что человека нет на месте и перехватывают его звонок. Эти параметры можно индивидуально задать для каждого пользователя в отдельности, если разбивка по шаблонам настроек не соответствует реальной рассадке людей в офисе.
- Параметр **callerid** можно задать кириллицей, но могут возникнуть проблемы с некоторыми телефонами и точно возникнут проблемы, когда вы будете вести статистику звонков в `mysql`. Я не смог победить эту проблему с кодировками, поэтому использую только латиницу в этом параметре. Туда можно писать либо должность, либо ФИО человека.
- **call-limit=1** задает количество одновременных соединений на линию. Если у вас один человек = один телефонный аппарат, то разрешать больше одной линии на пользователя нет смысла. Ему будет идти новый звонок в тот момент, как он разговаривает. Конечно, если есть необходимость переключаться между разговорами и ставить кого-то на удержание, то можно делать и больше линий. Но мне кажется, это неудобно. Если ты разговариваешь, пусть звонящий лучше услышит занято и перезвонит.
- Я всех добавляю в один `context`. В данном примере у нас будет только один номер телефона на всех. Если у вас их будет несколько, то контекстами можно будет разводить звонки на разные номера. Эту ситуацию я рассмотрю в отдельной статье.

Сохраняем *sip.conf*. Теперь нам нужно добавить план звонков, для того, чтобы можно было совершать вызовы. Для этого как и с предыдущим конфигурационным файлом, очищаем файл *extensions.conf* и записываем туда следующую информацию:

```
[general]
static=yes
writeprotect=no
[globals]
[default]

;Вешаем трубку
[handup-sip]
exten => _X!,1,HangUp()

;Исходящие звонки
[call-out]
;Звонок на внутренний номер
exten => _XXX,1,Dial(SIP/${EXTEN})
include => handup-sip
```

Мы просто добавили один план звонков на трехзначные номера. Я сразу добавляю комментарии с заделом на дальнейшее расширение этого файла, чтобы потом было проще разбираться. Сохраняем файл.

У нас все готово для внутренних звонков через asterisk. Заходим в консоль и перезагружаем его:

```
# asterisk -r
```

```
CLI> reload
```

Вы увидите некоторые предупреждения и ошибки. Это не страшно, так и должно быть, так как мы многое еще не настроили. Загружаются модули, которым не хватает настроек. Проверим список созданных пользователей с помощью команды в консоли:



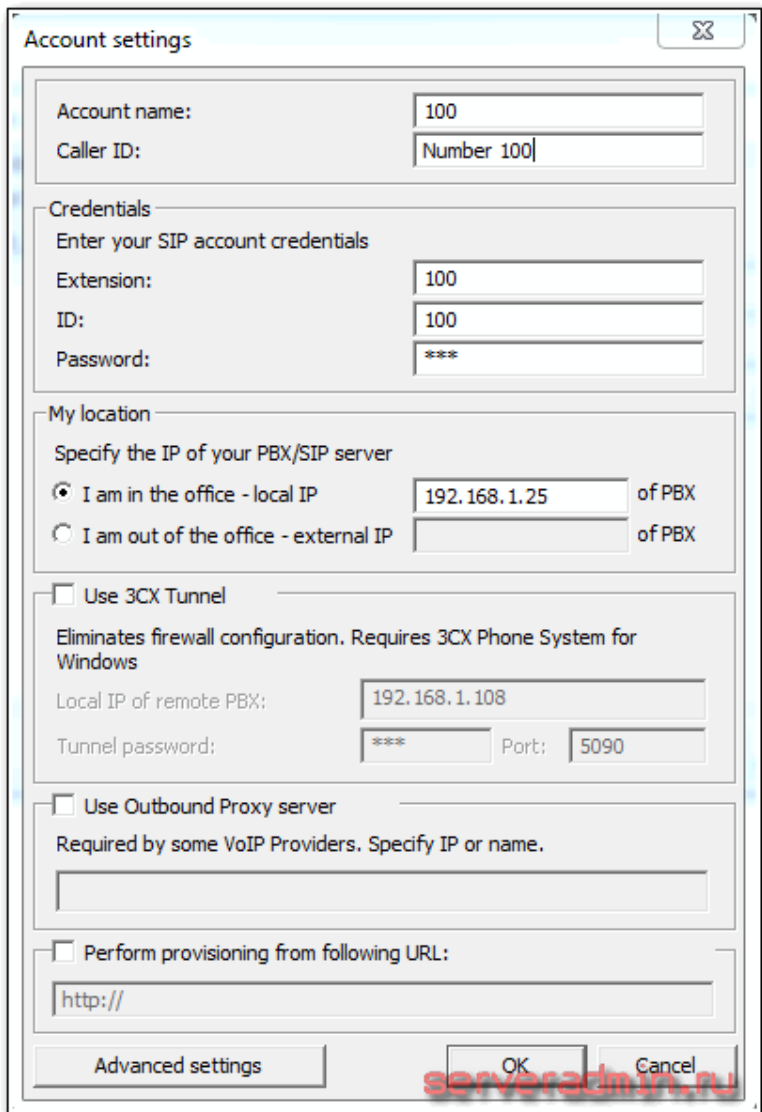
```
CLI> sip show users
```

```
asterisk*CLI> sip show users
Username                Secret                Accountcode           Def.Context           ACL Forcerport
104                     123                   call-out              call-out               No No
105                     123                   call-out              call-out               No No
106                     123                   call-out              call-out               No No
107                     123                   call-out              call-out               No No
100                     123                   call-out              call-out               No No
101                     123                   call-out              call-out               No No
102                     123                   call-out              call-out               No No
103                     123                   call-out              call-out               No No
108                     123                   call-out              call-out               No No
109                     123                   call-out              call-out               No No
126                     789                   call-out              call-out               No No
127                     789                   call-out              call-out               No No
124                     789                   call-out              call-out               No No
125                     789                   call-out              call-out               No No
122                     789                   call-out              call-out               No No
123                     789                   call-out              call-out               No No
120                     456                   call-out              call-out               No No
121                     789                   call-out              call-out               No No
128                     789                   call-out              call-out               No No
129                     789                   call-out              call-out               No No
115                     456                   call-out              call-out               No No
114                     456                   call-out              call-out               No No
117                     456                   call-out              call-out               No No
116                     456                   call-out              call-out               No No
111                     456                   call-out              call-out               No No
110                     123                   call-out              call-out               No No
113                     456                   call-out              call-out               No No
112                     456                   call-out              call-out               No No
119                     456                   call-out              call-out               No No
118                     456                   call-out              call-out               No No
130                     789                   call-out              call-out               No No
asterisk*CLI>
```

serveradmin.ru

Видим всех наших пользователей, их пароли и контекст. Список отсортирован не по порядку, не пугайтесь, если не заметите какой-то номер.

Дальше устанавливайте любую софтовую звонилку на компьютер или можете сразу использовать телефон, если он у вас под рукой. С телефонами удобнее, но не всегда они есть. В **3CXPhone** задаем следующие настройки подключения к нашей ip атс:



Account settings

Account name: 100  
Caller ID: Number 100

Credentials  
Enter your SIP account credentials  
Extension: 100  
ID: 100  
Password: \*\*\*

My location  
Specify the IP of your PBX/SIP server  
 I am in the office - local IP 192.168.1.25 of PBX  
 I am out of the office - external IP of PBX

Use 3CX Tunnel  
Eliminates firewall configuration. Requires 3CX Phone System for Windows  
Local IP of remote PBX: 192.168.1.108  
Tunnel password: \*\*\* Port: 5090

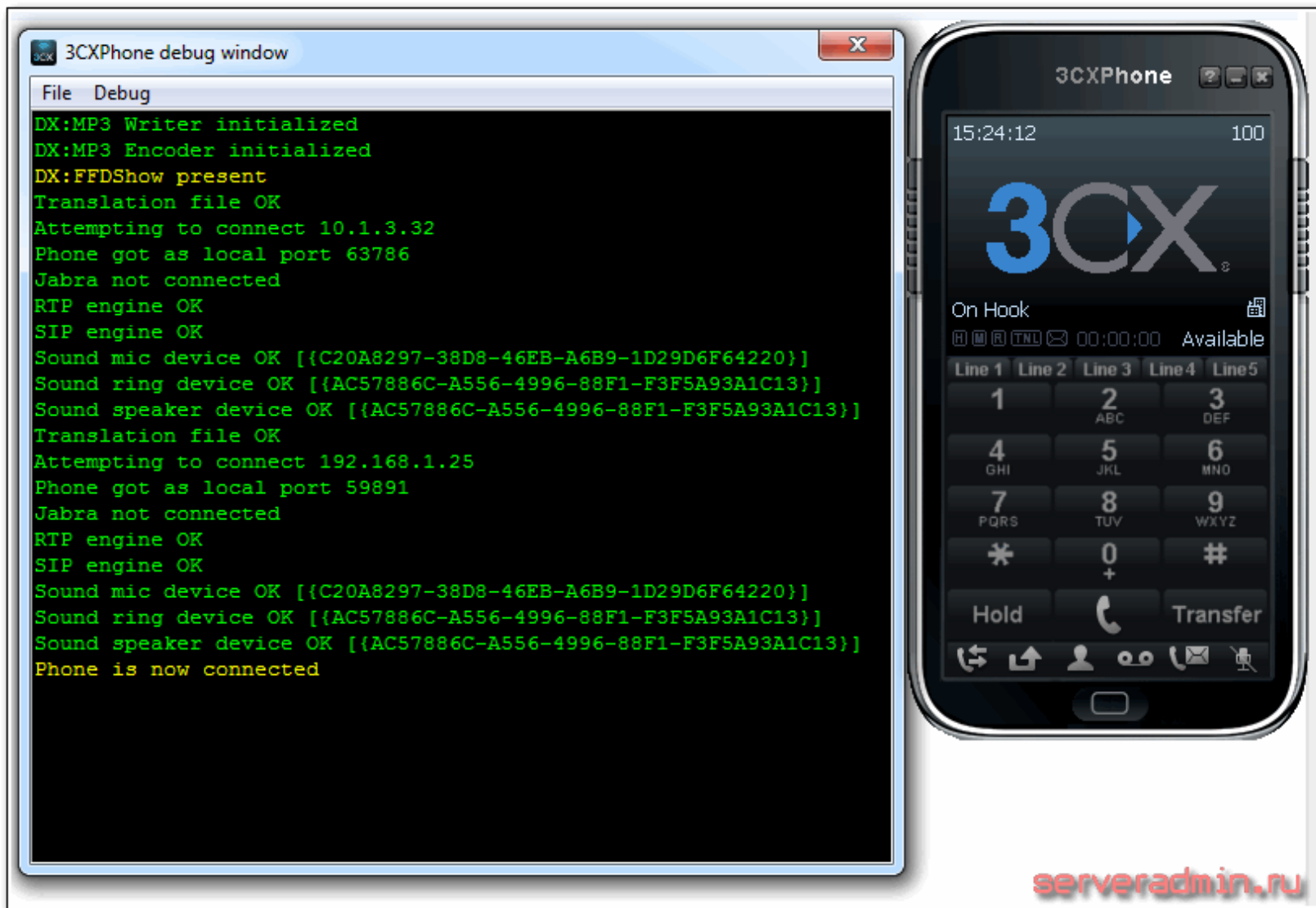
Use Outbound Proxy server  
Required by some VoIP Providers. Specify IP or name.

Perform provisioning from following URL:  
http://

Advanced settings OK Cancel

Я сразу открываю дебаг окно для отладки. Корректное подключение к серверу будет

выглядеть вот так:



Проверим на сервере список подключенных пиров с помощью команды:

```
CLI> sip show peers
```

Должна быть строка с подключенным пиром:

100/100	192.168.1.100	D	No	No	59891	OK (104 ms)
---------	---------------	---	----	----	-------	-------------

Если у вас так же, то все в порядке. Чтобы протестировать звонки, нам нужно подключить двух абонентов. Настраивайте еще один телефон или софтофон. Проверяйте в списке подключенных пиров чтобы было 2 подключения и попробуйте позвонить друг другу. Если вы все сделали правильно, то локальные звонки должны работать. Я обычно для тестирования второго клиента настраиваю на мобильном телефоне. Так можно проверить работу звонков через смартфоны.

После звонка в файле `/var/log/asterisk/cdr-csv/Master.csv` появится запись о совершенном звонке:

```
","101","100","call-out","""Number 101"" <101>","SIP/101-00000000","SIP/100-00000001","Dial","SIP/100","2020-02-27  
12:38:35","2020-02-27 12:38:45","2020-02-27 12:38:47",12,2,"ANSWERED","DOCUMENTATION","1582807115.0", ""
```

В этом файле будет накапливаться статистика звонков. Позже мы перенесем ее в mysql. Я позвонил с номера 101 на номер 100, там мне ответили. В файле отражены все основные данные этого звонка.

Один небольшой шаг по настройке voip atc asterisk мы сделали. Будем двигаться дальше.

## Настройка sip trunk (транка) и добавление номера

Регистрируемся у какого-нибудь sip провайдера и получаем настройки транков для подключения. Как я говорил выше, я буду использовать провайдера **zadarma**. После регистрации в личном кабинете в разделе **Настройки -> Подключение по SIP** я вижу свой логин, пароль для подключения и адрес сервера. Там же можно узнать пример настройки подключения для астериска и номера для тестовых звонков:

- Номер для эхо-теста: 4444.
- Информация про остаток на счету: 1111
- Прямой звонок: Москва +7 (495) 777-66-75 и внутренний номер клиента (логин)

Нам этого будет достаточно для полноценного тестирования конфигурации астериска. Добавляем необходимые настройки sip транка, чтобы выполнить его регистрацию. Редактируем *sip.conf*, добавляем в самый конец нового пира в соответствии с инструкцией провайдера:

```
[397945]
host=sip.zadarma.com
insecure=invite,port
type=friend
fromdomain=sip.zadarma.com
disallow=all
allow=alaw
dtmfmode=auto
secret=password
defaultuser=397945
trunkname=397945
fromuser=397945
callbackextension=397945
context=call-in
qualify=400
directmedia=no
nat=force_rport,comedia
```

В данном случае **397945** мой внутренний номер в сервисе, password - пароль. Сохраняем файл и даем команду астеру перечитать его:

```
CLI> sip reload
```

Тут же в консоли, если все в порядке, вы получите сообщение:

```
chan_sip.c:24403 handle_response_peerpoke: Peer '397945' is now Reachable. (55ms / 400ms)
```

Это означает, что наш пир успешно зарегистрировался у провайдера. Проверить это можно с помощью уже известной команды:

```
CLI> sip show peers
397945/397945          185.45.152.161          Auto (No)  No          5060      OK (54 ms)
```

Все, транк настроили, по сути подключили номер. Но этого не достаточно, чтобы совершать и принимать звонки. Надо отредактировать **dialplan**.

## Dial-plan - пример маршрутизации звонков

Начало построения диалплана для маршрутизации звонков мы уже положили, когда настраивали внутренние звонки. Теперь нужно дополнить **dial-plan** для совершения исходящих и приема входящих звонков. Редактируем *extensions.conf* и приводим его к следующему виду:

```
[general]
static=yes
writeprotect=no
[globals]
[default]

;Вешаем трубку
[handup-sip]
exten => _X!,1,HangUp()

;Исходящие звонки
[call-out]
;Звонок на внутренний номер
exten => _XXX,1,Dial(SIP/${EXTEN})
```



```
;Звонок на внешний номер
exten => _XXX.,1,Dial(SIP/${EXTEN}@397945)

include => handup-sip

;Входящие звонки
[call-in]
exten => 397945,1,Dial(SIP/100)
```

Я выложил полную версию файла, а не только то, что добавил. Добавленные строки выделил цветом.

В нашем примере в контекст исходящих звонков мы добавили правило набора любого номера длиннее трех символов через транк 397954. Если в локальных звонках маска экстеншена задается тремя любыми символами, то тут мы в конец добавили точку, которая означает любое количество символов.

С таким диалпланом при наборе трехзначного номера вы позвоните на локальный номер, а при наборе любого городского номера сможете позвонить на него без каких-либо добавочных цифр. Сразу набираете номер с 8 и так далее.

Контекст [call-in] описывает поведение при входящем звонке. В нашем случае все входящие звонки с транка 397945 будут направляться на номер секретаря 100. К этому контексту мы еще вернемся позже, когда будем настраивать голосовое меню.

Сохраняем dial-plan и перезагружаем астериск единой командой reload, либо отдельно перезагружаем sip и dialplan командами:

```
CLI> sip reload
CLI> dialplan reload
```

Теперь можно попробовать позвонить, к примеру, на тестовый номер zadarma - 4444 для эхотеста. Если все получилось, значит вы правильно настроили исходящие звонки. Для проверки входящего звонка, позвоните в Москве на номер +7 (495) 777-66-75 и введите свой добавочный номер в виде логина. Звонок должен переключиться на номер 100. Чтобы это произошло, необходимо, чтобы peer с номером 100 был подключен к астериску.

Я проверил, без проблем дозвонился и по исходящему номеру, и по входящему. В файле *Master.csv* появилась информация о совершенных звонках:

```
"", "100", "4444", "call-out", "" "Number 100"  
<100>", "SIP/100-000000002", "SIP/397945-000000003", "Dial", "SIP/4444@397945", "2020-02-27 14:13:18", "2020-02-27  
14:13:20", "2020-02-27 14:13:25", 6, 5, "ANSWERED", "DOCUMENTATION", "1582812798.3", ""  
"", "79689056505", "397945", "call-in", "" "79689056505"  
<79689056505>", "SIP/397945-000000004", "SIP/100-000000005", "Dial", "SIP/100", "2020-02-27 14:14:11", , "2020-02-27  
14:14:16", 4, 0, "BUSY", "DOCUMENTATION", "1582812851.6", ""
```

После звонков в консоли астера и в логе `/var/log/asterisk/messages` вы увидите множество ошибок:

```
res_hep.c: Unable to send packet: Address Family mismatch between source/destination
```

Чтобы их не было, вам нужно на сервере отключить протокол `ipv6`.

В таком виде АТС уже вполне работоспособна, можно пользоваться, но есть ряд неудобств. Например, если позвонить на трехзначный номер, которого не существует, или на номер, который не зарегистрирован в данный момент на устройстве, вы просто получите сброс звонка и не поймете, в чем проблема. Для этих событий нужна отдельная обработка. Чтобы не загружать сразу файл конфигурации длинными и непонятными конструкциями, я рассмотрю этот момент ниже, когда буду рассказывать про голосовую почту. Это не критичная настройка, в таком виде все будет замечательно работать, но не так удобно, как могло бы.

## Приветствие и голосовое меню (ivr)

Основной функционал asterisk реализован. Будем его расширять. Практически на всех АТС присутствует голосовое меню, которое встречает звонящего. Я рассмотрю настройку самого простого варианта голосового меню, как его еще называют **ivr**. Позвонив, человек услышит какое-то приветствие, далее ему будет предложено ввести внутренний номер абонента, если он его знает, либо дождаться ответа секретаря.

Для простоты настройки и отладки, поделюсь способом, который использую я. У проекта `zadarma` есть клиент под андроид - **Zadarma SIP**. Скачиваете его на телефон и регистрируете еще одну учетную запись в проекте. Логинитесь под ней в телефоне и можете звонить на свой астериск, просто набирая 6-ти значный номер аккаунта, который используете на сервере. Таким образом вы быстро, удобно и бесплатно эмулируете входящие звонки с внешних линий на свой сервер.

Начнем настройку с того, что создадим возможность для записи приветствия. Вы можете записать его где угодно и потом скопировать на сервер asterisk. Но можно поступить удобнее - записать прямо с телефонного аппарата приветствие и использовать его в голосовом меню. Чтобы это сделать, необходимо добавить в dialplan в контекст исходящих звонков, в моем примере это [call-out], в самое начало следующую конструкцию:

```
;Номер для записи звуков, окончание записи #
exten => _35X, 1, NoOp()
exten => _35X, n, Wait(2)
exten => _35X, n, Playback(beep)
exten => _35X, n, Record(/tmp/music${EXTEN:2}:wav)
exten => _35X, n, Wait(1)
exten => _35X, n, Playback(/tmp/music${EXTEN:2})
exten => _35X, n, Wait(2)
exten => _35X, n, Hangup()
```

Перезагружаем dialplan:

```
CLI> dialplan reload
```

Теперь при звонке на любой из номеров 350-359 вы услышите бип, после которого начнется запись всего, что сказано в трубку. Чтобы завершить запись, нажмите #. После этого вы прослушаете то, что было записано. Файл с записью будет сохранен в папку /tmp. Если вы позвоните на номер 351, то файл будет иметь имя music1.wav, если на 355, то music5.wav. Можно записать до 10-ти разных вариантов и потом из них выбирать.

Сохраните подходящую запись с именем *ivr-main.wav* и разместите его в какой-нибудь папке. Я положил в папку */etc/asterisk/ivr*. Добавим пример голосового меню в нашу конфигурацию asterisk. Для этого снова открываем *extensions.conf* и добавляем в него новый контекст [ivr-main] следующего содержания:

```
[ivr-main]
exten => s,1,Answer()
;Проигрываем приветствие
```

```
exten => s,2,Background(/etc/asterisk/ivr/ivr-main)
;Ждем 5 секунд ввода добавочного номера
exten => s,3,WaitExten(5)
;Звоним по введенному добавочному
exten => _XXX,1,Dial(SIP/${EXTEN})
;Если введен не существующий номер, то говорим об этом и отправляем в начало приветствия
exten => _XXX,2,Playback(privacy-incorrect)
exten => _XXX,3,Goto(ivr-main,s,1)
;Если звонящий ничего не вводит, то звоним секретарю
exten => t,1,Dial(SIP/100)
```

В комментариях я сделал все пояснения. Мы создали контекст для ivr. Теперь его надо добавить в контекст входящих звонков. Для этого изменяем существующий контекст [call-in], заменяя в нем единственную строку на новую:

```
exten => 397945,1,Goto(ivr-main,s,1)
```

Перезапускаем диал план и звоним снаружи на внешний номер. Вы должны услышать голосовое приветствие ivr, которое мы только что настроили. После этого этапа ваш файл *extensions.conf* должен выглядеть примерно так (комментарии вырезал):

```
[general]
static=yes
writeprotect=no
[globals]
[default]

[handup-sip]
exten => _X!,1,HangUp()

[call-out]
```

```
exten => _35X, 1, NoOp()
exten => _35X, n, Wait(2)
exten => _35X, n, Playback(beep)
exten => _35X, n, Record(/tmp/music${EXTEN:2}:wav)
exten => _35X, n, Wait(1)
exten => _35X, n, Playback(/tmp/music${EXTEN:2})
exten => _35X, n, Wait(2)
exten => _35X, n, Hangup()
exten => _XXX,1,Dial(SIP/${EXTEN})
exten => _XXX.,1,Dial(SIP/${EXTEN}@397945)
include => handup-sip

[call-in]
exten => 397945,1,Goto(ivr-main,s,1)

[ivr-main]
exten => s,1,Answer()
exten => s,2,Background(/etc/asterisk/ivr/ivr-main)
exten => s,3,WaitExten(5)
exten => _XXX,1,Dial(SIP/${EXTEN})
exten => _XXX,2,Playback(privacy-incorrect)
exten => _XXX,3,Goto(ivr-main,s,1)
exten => t,1,Dial(SIP/100)
```

## Включаем голосовую почту

Продолжаем наращивать функционал voip атс. В данном разделе опишу настройку голосовой почты в asterisk. Для начала пару слов о том, что это такое. Если адресат звонка долго не отвечает, мы можем предложить звонящему оставить для него голосовое сообщение. Когда получатель вернется на место, он сможет прослушать оставленные ему сообщения. При этом, после записи голосового сообщения, будет отправлено письмо с записью этого сообщения на почтовый адрес получателя.

На практике я не видел, чтобы голосовую почту активно использовали. Но для полноты картины расскажу про нее, возможно вам она пригодится. Открываем файл *voicemail.conf* на редактирование. Я не трогал настройки по-умолчанию, просто добавляем в секцию [default] ящики голосовой почты для нужных нам сотрудников в следующем виде:

```
130 => 1234,Number 130,user130@mail.ru
100 => 1234,Number 100,user100@mail.ru
```

130	внутренний номер абонента
1234	пароль доступа к ящику голосовой почты
user130@mail.ru	почтовый адрес, куда будет отправлено записанное голосовое сообщение

Для корректной отправки почтовых сообщений сразу на внешние почтовые ящики необходимо правильно настроить локальный почтовый сервер, либо использовать внешний. Я рекомендую использовать отдельный сервер, наверняка он есть в организации, либо на локальном использовать какой-то публичный с авторизацией по smtp. Пример такой настройки - отправка почты с авторизацией по smtp в linux.

В консоли перезапускаем модуль голосовой почты и проверяем пользователей:

```
CLI> voicemail reload
Reloading voicemail configuration...
asterisk*CLI> voicemail show users
Context Mbox User Zone NewMsg
default 130 Number 121 0
default 100 Number 100 0
other 1234 Company2 User 0
3 voicemail users configured.
```

Наши два добавленных пользователя и один тестовый остался из дефолтной конфигурации. Его можно удалить. Их вообще два должно быть. Одного я уже удалил, второго забыл.

Это пол дела. Теперь нам нужно добавить голосовую почту в dialplan. Причем в 2 разных места. Я буду использовать номер 500 для звонка в панель управления голосовой почтой. Позвонив на этот номер, пользователь введет свой пароль и сможет управлять голосовыми сообщениями (слушать, удалять, менять настройки). Добавим в контекст для исходящих звонков звонок на этот номер. Добавлять следует сразу за номерами для записи, которые мы ранее создали и перед правилом набора трехзначных номеров.

```
[call-out]
;Номер для записи звуков, окончание записи #
exten => _35X, 1, NoOp()
exten => _35X, n, Wait(2)
exten => _35X, n, Playback(beep)
exten => _35X, n, Record(/tmp/music${EXTEN:2}.wav)
exten => _35X, n, Wait(1)
exten => _35X, n, Playback(/tmp/music${EXTEN:2})
exten => _35X, n, Wait(2)
exten => _35X, n, Hangup()
;Управление голосовой почтой
exten => 500,1,VoiceMailMain()
;Звонок на внутренний номер
exten => _XXX,1,Dial(SIP/${EXTEN},15)
;Звонок на внешний номер
exten => _XXX.,1,Dial(SIP/${EXTEN}@397945)
```

Добавленные данные выделил цветом. Обращаю внимание на цифру **15**. Ранее этой настройки не было, сейчас я добавил. Она будет означать, что звонок будет длиться 15 секунд. Если за это время никто не ответит, он будет сброшен. До использования голосовой почты, можно было не устанавливать этот параметр, оставить его значение по-умолчанию, оно очень большое, не помню точно сколько по времени. Но сейчас нам нужно при неснятии трубки дольше 15-ти секунд, включать запись голосового сообщения.

Конкретно в контексте внутренних звонков этот параметр не принципиален, так как я не добавляю голосовую почту для звонков внутри организации, хотя можно это сделать. Не вижу в этом смысла. Я добавил сюда этот параметр, чтобы время ожидания ответа было одинаково во всех звонках. В следующих изменениях это уже будет играть принципиальное значение.

Пока мы просто добавили номер, куда можно позвонить для управления голосовой почтой. Теперь добавим непосредственно возможность записи голосовых сообщений по событиям. Остановимся на этом пункте поподробнее. Для начала уясним, в каких состояниях может пребывать номер:

1. Номера вообще не существует на сервере. В нашем случае, к примеру, это любой номер не из диапазона 100-130.
2. Номер существует, но он не зарегистрирован на АТС, то есть аппарат с этим номером не подключен.
3. Номер существует, зарегистрирован, но при звонке на него никто не отвечает.
4. Номер существует, зарегистрирован, но в данный момент занят.

Каждое из этих четырех состояний обрабатывается отдельно. Я считаю, что голосовую почту уместно будет включать по событию номер 3. Но это не обязательно, можно и на занято повесить возможность оставить сообщение. Тут на ваше усмотрение. Я покажу пример, как это делается, а вы сможете настроить так, как вам нужно.

Обработка этих событий не такая простая, как кажется на первый взгляд. Я сразу же столкнулся с трудностью следующего характера. С параметром `call-limit=1` при звонке на номер, который занят в данный момент, астериск возвращает статус `CHANUNAVAIL`, что может означать, к примеру, что канал недоступен. На статус занято `BUSY` это совсем не похоже. Для разрешения этой ситуации я воспользуюсь функцией **ChanIsAvail**, которая проверяет не статус пира, а статус канала и возвращает значение 2 или 3, когда он занят.

Для различения несуществующих и не подключенных пиров я буду использовать функцию **SIPPEER**. Если она ничего не возвращает, значит номера не существует, если значение `UNKNOWN`, значит номер не подключен. С отсутствием ответа какое-то время проще всего. Если пир возвращает статус `NOANSWER`, включаем голосовую почту. Собираем все статусы в одно место и добавляем голосовую почту. Для этого в контекст `[ivr-main]` добавляем новые параметры и приводим его к следующему виду:

```
[ivr-main]
exten => s,1,Answer()
exten => s,2,Background(/etc/asterisk/ivr/ivr-main)
exten => s,3,WaitExten(5)
exten => _XXX,1,Dial(SIP/${EXTEN},15)
;Задаем переменную для передачи в голосовую почту
exten => _XXX,n,Set(dstNUM=${EXTEN})
;Проверяем статус пира, существует или нет
exten => _XXX,n,GotoIf("${SIPPEER(${EXTEN},status)}" = "")?num-not-exist,1)
;Проверяем статус пира, подключен или нет
exten => _XXX,n,GotoIf("${SIPPEER(${EXTEN},status):0:2}" = "UN")?num-not-connected,1)
```



```
;Проверяем канал на занятость
exten => _XXX,n,ChanIsAvail(SIP/${EXTEN},s)
;Выводим в лог значение функции ChanIsAvail, нужно только для отладки, можно удалить строку
exten => _XXX,n,NoOp(===== ChanIsAvail STATUS: ${AVAILSTATUS} =====)
;Если функция возвращает 2 или 3, значит абонент занят
exten => _XXX,n,GoToIf(${AVAILSTATUS} = 2)?num-BUSY,1)
exten => _XXX,n,GoToIf(${AVAILSTATUS} = 3)?num-BUSY,1)
;Обрабатываем остальные статусы
exten => _XXX,n,Goto(num-${DIALSTATUS},1)
;Если номера не существует говорим "Ошибочный номер, попробуйте еще раз"
exten => num-not-exist,1,Wait(2)
exten => num-not-exist,n,Playback(invalid)
;Если номер не подключен, говорим "Набранный вами номер отключен, проверьте номер и повторите попытку"
exten => num-not-connected,1,Wait(2)
exten => num-not-connected,n,Playback(ss-noservice)
;Если номер занят, говорим "Занято"
exten => num-BUSY,1,Wait(2)
exten => num-BUSY,n,Playback(vm-isonphone)
;Если номер не отвечает, включаем голосовую почту
exten => num-NOANSWER,1,Wait(2)
exten => num-NOANSWER,n,Voicemail(${dstNUM},u)
;Если еще по какой-то причине будет статус CHANUNAVAIL, говорим, что номер не доступен в данный момент
exten => num-CHANUNAVAIL,1,Wait(2)
exten => num-CHANUNAVAIL,n,Playback(vm-isunavail)
;Если в голосовом меню не выбрали внутренний номер, адресуем звонок секретарю
exten => t,1,Dial(SIP/100,15)
```

Если вам нужна обработка статусов номеров для внутренних звонков, то скопируйте обработку состояний в контекст [call-out], за исключением последней строки, которая отвечает за звонок секретарю. Полностью контекст внутренних звонков будет выглядеть вот так:

```
[call-out]
```

```
;Номер для записи звуков, окончание записи #
exten => _35X, 1, NoOp()
exten => _35X, n, Wait(2)
exten => _35X, n, Playback(beep)
exten => _35X, n, Record(/tmp/music${EXTEN:2}:wav)
exten => _35X, n, Wait(1)
exten => _35X, n, Playback(/tmp/music${EXTEN:2})
exten => _35X, n, Wait(2)
exten => _35X, n, Hangup()
;Управление голосовой почтой
exten => 500,1,VoiceMailMain()
;Звонок на внутренний номер
exten => _XXX,1,Dial(SIP/${EXTEN},15)
exten => _XXX,n,Set(dstNUM=${EXTEN})
exten => _XXX,n,GotoIf(["${SIPPEER(${EXTEN},status)}" = ""]?num-not-exist,1)
exten => _XXX,n,GotoIf(["${SIPPEER(${EXTEN},status):0:2}" = "UN"]?num-not-connected,1)
exten => _XXX,n,ChanIsAvail(SIP/${EXTEN},s)
exten => _XXX,n,NoOp(===== ChanIsAvail STATUS: ${AVAILSTATUS} =====)
exten => _XXX,n,GoToIf(${AVAILSTATUS} = 2]?num-BUSY,1)
exten => _XXX,n,GoToIf(${AVAILSTATUS} = 3]?num-BUSY,1)
exten => _XXX,n,Goto(num- ${DIALSTATUS},1)
exten => num-not-exist,1,Wait(2)
exten => num-not-exist,n,Playback(invalid)
exten => num-not-connected,1,Wait(2)
exten => num-not-connected,n,Playback(ss-noservice)
exten => num-BUSY,1,Wait(2)
exten => num-BUSY,n,Playback(vm-isonphone)
exten => num-NOANSWER,1,Wait(2)
exten => num-NOANSWER,n,Voicemail(${dstNUM},u)
exten => num-CHANUNAVAIL,1,Wait(2)
exten => num-CHANUNAVAIL,n,Playback(vm-isunavail)
;Звонок на внешний номер
exten => _XXX.,1,Dial(SIP/${EXTEN}@397945)
```

```
include => handup-sip
```

Я во всех звонках установил время ожидания ответа 15 секунд. Если вы считаете, что нужно больше, измените этот параметр.

Обращаю ваше внимание, что в контексте голосового меню я не сделал обработку статусов состояния телефона секретаря, хотя это может быть нужно, если у вас будет один секретарь принимать звонки. Ему и голосовая почта может пригодиться. В следующем пункте я расскажу про случай, когда в офисе работают 2 секретаря и обработкой звонков будет заниматься очередь (queue), поэтому статусы в том виде, как они реализованы здесь будут не нужны. Вы можете использовать любую конфигурацию, которая вам подойдет. Например, использовать очередь, но с одним секретарем в ней. Настраивайте по аналогии, я даю базовый функционал. Все возможные случаи разобрать невозможно.

Перечитывайте диалплан и тестируйте конфигурацию. При звонке абоненту и его неответе, звонящий услышит в трубке сообщение о том, что номер не отвечает и предложение оставить голосовую почту. Если звонивший оставит сообщение, то получатель получит это сообщение по email и сможет его прослушать там, либо позвонить на номер 500, ввести свой номер и пароль, заданные в *voicemail.conf* и прослушать сообщение по телефону.

Я сталкивался с ошибкой, когда у пользователя в настройках отключена голосовая почта, но он все равно получал сообщения. У него мигало оповещение на телефонном аппарате. Нужно было оперативно удалить все оставленные сообщения без прослушивания. Как удалить голосовую почту в астериск я написал в отдельной заметке.

На этом настройка голосовой почты окончена. Можно пользоваться, не забывая добавлять новых пользователей в *voicemail.conf* и перечитывая конфигурацию голосовой почты.

## Очереди (queues) входящих звонков

С помощью очередей в астериске можно управлять потоком входящих звонков, перераспределяя их по определенным правилам. В нашей конфигурации asterisk мы настроим очередь (queue) для направления звонка двум секретарям одновременно. Кто первый ответит, тот и будет разговаривать со звонящим. Если один секретарь уже разговаривает, новый звонок поступит к другому. Если оба секретаря будут заняты, звонящий будет слушать мелодию и ожидать, пока кто-нибудь не освободится. Как только один из секретарей освободится, звонящего из очереди направит на освободившийся номер.

Номер первого секретаря - 100, второго - 130. Я настраиваю простейшую конфигурацию очереди в астериск для понимания принципа работы. Более

сложный вариант настройки это сделать 3 очереди для каждого отдела и в голосовом меню-приветствии сделать возможность позвонить в конкретный отдел. Эту конфигурацию я рассмотрю в отдельной статье, хотя в нем и нет ничего сложного. Делается по аналогии с приведенным примером.

Открываем файл *queues.conf* и добавляем в самый конец:

```
[secretary]
strategy = ringall
member => SIP/100
member => SIP/130
```

Все остальные настройки оставляю по-умолчанию. Параметр **strategy** может принимать следующие значения:

- ringall        вызываются все доступные участники до тех пор, пока кто-то из них не ответит на вызов (по умолчанию).
- leastrecent    Вызывается первый свободный участник, который меньше всего вызывался из этой очереди.
- fewestcalls    Вызывается первый свободный участник, который обработал наименьшее количество вызовов из данной очереди.
- random        случайным образом вызывается не занятый участник, обрабатывающий очередь.
- rrmemory      циклическое распределение с памятью, запоминается последний участник, ответивший на вызов.

Вы можете выбрать наиболее подходящую вам стратегию распределения звонков в очереди. Далее нужно добавить в *extensions.conf* в созданный нами ранее контекст с голосовым меню отправку звонка в очередь с секретарями. Для этого меняем строку в [ivr-main]:

```
exten => t,1,Dial(SIP/100,15)
```

на новую:

```
exten => t,1,Queue(secretary,t)
```

Если раньше при звонке на внешний номер, звонящий не набирал внутренний номер абонента, то через 5 секунд он перенаправлялся к секретарю с номером 100. Теперь он будет отправляться в очередь secretary, в которую мы завели 2 номера - 100 и 130. Можно добавить и больше номеров, если есть необходимость.

Перечитываем полностью конфигурацию asterisk:

```
CLI> reload
```

Состояние созданной очереди:

```
CLI> queue show secretary
secretary has 0 calls (max unlimited) in 'ringall' strategy (0s holdtime, 0s talktime), W:0, C:0, A:0, SL:0.0% within 0s
Members:
SIP/100 (ringinuse enabled) (Not in use) has taken no calls yet
SIP/130 (ringinuse enabled) (Not in use) has taken no calls yet
No Callers
```

Теперь можете позвонить на внешний номер и дождаться перенаправления на секретаря. Зазвонят оба аппарата. Если оба заняты, звонок будет висеть в очереди и ждать освобождения. После звонка статистика очереди изменится:

```
CLI> queue show secretary
secretary has 0 calls (max unlimited) in 'ringall' strategy (0s holdtime, 2s talktime), W:0, C:1, A:1, SL:0.0% within 0s
Members:
SIP/100 (ringinuse enabled) (Not in use) has taken 1 calls (last was 63 secs ago)
SIP/130 (ringinuse enabled) (Not in use) has taken no calls yet
```

## No Callers

Вот так легко организовать простую queue (очередь) в asterisk. Более сложные примеры я буду рассматривать в отдельных статьях.

## Учет и просмотр статистики звонков (cdr viewer)

Важной и нужной возможностью современной АТС на базе asterisk является сбор и просмотр статистики звонков. По умолчанию, астериск ведет статистику в файле `/var/log/asterisk/cdr-csv/Master.csv`. Разобрать этот файл и передать куда-то в обработку не очень сложно, если у вас есть что-то или кто-то, кто способен написать на каком-нибудь языке программирования обработку.

Мы будем использовать готовые бесплатные инструменты для просмотра статистики звонков. Все необходимое для этого мы установили в самом начале. Перенесем сбор статистики в mysql базу. Для этого запускаем mariadb сервер, добавляем в автозагрузку и устанавливаем пароль администратора:

```
# systemctl start mariadb
# systemctl enable mariadb.service
# /usr/bin/mysql_secure_installation
```

Подключаемся к mysql и создаем пользователя и базу данных:

```
# mysql -uroot -p
MariaDB [(none)]> create database asterisk;
MariaDB [(none)]> use asterisk;
MariaDB [asterisk]> CREATE TABLE `cdr` (
  `id` int(9) unsigned NOT NULL auto_increment,
  `calldate` datetime NOT NULL default '0000-00-00 00:00:00',
  `clid` varchar(80) NOT NULL default '',
  `src` varchar(80) NOT NULL default '',
  `dst` varchar(80) NOT NULL default '',
  `dcontext` varchar(80) NOT NULL default '',
  `channel` varchar(80) NOT NULL default '',
  `dstchannel` varchar(80) NOT NULL default '',
  `lastapp` varchar(80) NOT NULL default '',
  `lastdata` varchar(80) NOT NULL default '',
  `duration` int(11) NOT NULL default '0',
  `billsec` int(11) NOT NULL default '0',
  `disposition` varchar(45) NOT NULL default '',
  `amaflags` int(11) NOT NULL default '0',
  `accountcode` varchar(20) NOT NULL default '',
  `uniqueid` varchar(32) NOT NULL default '',
  `userfield` varchar(255) NOT NULL default '',
  PRIMARY KEY (`id`),
```

```
KEY `calldate` (`calldate`), KEY `accountcode` (`accountcode`), KEY `uniqueid` (`uniqueid`), KEY `dst` (`dst`),  
KEY `src` (`src`) ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;  
MariaDB [asterisk]> grant all on asterisk.* to 'asterisk_user'@'localhost' identified by '12345678';
```

```
asterisk      имя базы данных  
asterisk_user пользователь базы данных  
12345678     пароль пользователя бд
```

Теперь нам надо установить odbc коннектор для mysql - **mysql-connector-odbc**. С этим в Centos 8 есть некоторые трудности, так как нужных пакетов нет в базовых репозиториях. Я в итоге его взял из официальной репы mysql. Для этого скачиваем и подключаем их репозиторий.

```
# wget https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm  
# dnf localinstall mysql80-community-release-el8-1.noarch.rpm  
# dnf makecache
```

Тут я получил ошибку:

```
Failed to download metadata for repo 'mysql-tools-community'  
Error: Failed to download metadata for repo 'mysql-tools-community'
```

Не стал разбираться, почему этот репозиторий недоступен, так как он мне не нужен. Просто зашел в /etc/yum.repos.d/mysql-community.repo и закомментировал строки с этим репозиторием.

```
#[mysql-tools-community]  
#name=MySQL Tools Community  
#baseurl=http://repo.mysql.com/yum/mysql-tools-community/el/8/$basearch/  
#enabled=1
```

```
#pgpcheck=1  
#pgpkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
```

После этого благополучно обновил кэш пакетов и установил mysql-connector-odbc на centos 8.

```
# dnf install mysql-connector-odbc
```

Теперь редактируем файлы конфигурации. Добавляем в самый конец */etc/asterisk/res\_odbc.conf*:

```
[asterisk]  
enabled => yes  
dsn => MySQL-asterisk  
username => asterisk_user  
password => 12345678
```

В конец файла */etc/asterisk/cdr\_adaptive\_odbc.conf*:

```
[cdr_adaptive_connection]  
connection=asterisk  
table=cdr  
alias start => calldate
```

Создаем файл */etc/odbc.ini* следующего содержания:

```
[MySQL-asterisk]  
Description = MySQL Asterisk database
```



```
Driver = MySQL
Server = localhost
User = asterisk_user
Password = 12345678
Socket = /var/lib/mysql/mysql.sock
Database = asterisk
```

Редактируем файл `/etc/odbcinst.ini`. Я его не трогал, оставил по-умолчанию, только в самый конец секции [MySQL] добавил две недостающие строки. Я не разобрался нужны они или нет, просто подсмотрел в другой инструкции. Вот как этот файл выглядит у меня:

```
[MySQL]
Description = ODBC for MySQL
Driver = /usr/lib/libmyodbc8.so
Setup = /usr/lib/libodbcmyS.so
Driver64 = /usr/lib64/libmyodbc8a.so
Setup64 = /usr/lib64/libodbcmyS.so
FileUsage = 1
CPOutput =
CPReuse =
```

В последнем конфиге проверьте все пути. Имена файлов могут немного отличаться в зависимости от установленной версии коннектора. Актуализируйте пути и имена файлов. После этого проверьте работу `odbc connector`. Для этого запустите в консоли команду:

```
# odbcinst -q -d
```

Вы должны увидеть список всех настроенных коннекторов, в том числе [Mysql]. Настроим использование нашего часового пояса в записях `cdr`. По-умолчанию там стоит часовой пояс GMT. Для этого в файле `/etc/asterisk/cdr.conf` указываем параметр:

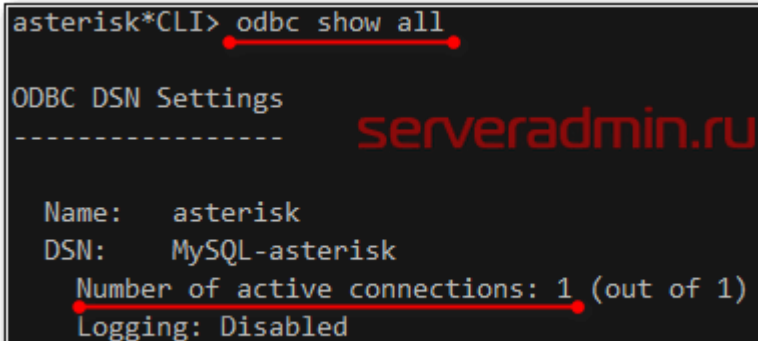
```
usegmtime=no
```

После этого перезапускаем астериск:

```
# systemctl restart asterisk
```

Заходим в консоль и проверяем подключение по odbc.

```
> odbc show all
```



```
asterisk*CLI> odbc show all  
  
ODBC DSN Settings  
-----  
Name: asterisk  
DSN: MySQL-asterisk  
Number of active connections: 1 (out of 1)  
Logging: Disabled
```

Совершаем звонок и проверяем таблицу mysql. У меня добавилась информация о звонке с номера 100 на 101, когда он был не подключен. Если сделать экспорт, то получится вот такая запись:

```
INSERT INTO `cdr` (`id`, `calldate`, `clid`, `src`, `dst`, `dcontext`, `channel`, `dstchannel`, `lastapp`, `lastdata`,
```

```
`duration`, `billsec`, `disposition`, `amaflags`, `accountcode`, `uniqueid`, `userfield`) VALUES  
(1, '2020-02-27 19:08:17', '\"Number 100\" <100>', '100', 'num-not-connected', 'call-out', 'SIP/100-0000000e', '',  
'Dial', 'SIP/101,15', 6, 4, 'ANSWERED', 3, '', '1582819697.14', '');
```

Для удобства дальнейшей работы, я настроил web сервер на базе apache. Не буду на этом подробно останавливаться. Чтобы двигаться дальше нам нужен классический web сервер на базе php. Для удобства рекомендую сразу настроить phpmyadmin или любой другой клиент к mysql. Если привыкли работать с mysql через консоль, можете обойтись без него.

Дальше нам надо настроить какую-нибудь web панель для просмотра этой статистики. Я решил сразу установить панель, которая позволяет не только смотреть статистику, но прослушивать записанные разговоры. Поэтому дальнейшая настройка панели просмотра статистики переходит в следующий раздел, в котором я расскажу, как записывать звонки.

## Запись (record) разговоров

Запись разговоров в asterisk настраивается относительно не сложно. Буквально нужно добавить несколько строк в dialplan. Но мы сразу сделаем более расширенную настройку. Мы будем не просто записывать все разговоры, но станем хранить информацию о звонках в mysql, чтобы их можно было прослушивать через удобную web панель просмотра статистики.

Хранить записи будем в **mp3**, потому нам понадобится утилита **lame**, для конвертации файлов из формата wav в mp3. Скачаем ее и установим.

```
# cd ~  
# wget http://sourceforge.net/projects/lame/files/lame/3.100/lame-3.100.tar.gz  
# tar zxvf lame-3.*  
# cd lame-3.*  
# ./configure  
# make && make install
```

В качестве web панели я буду использовать Asterisk-CDR-Viewer-Mod. Очень простая и функциональная штука. Настраивается легко, пользоваться удобно. Скачиваем исходники с github:

```
# cd ~  
# git clone https://github.com/prog-it/Asterisk-CDR-Viewer-Mod
```

Копируем содержимое репозитория в директорию cdr web сервера.

```
# mkdir /var/www/html/cdr  
# cp -R ~/Asterisk-CDR-Viewer-Mod/* /var/www/html/cdr  
# chown -R apache. /var/www/html/cdr
```

В папке docs подробная инструкция по настройке. Все дальнейшие действия делаются в соответствии с ней. Исправляются только некоторые особенности данной версии системы. Нам необходимо добавить дополнительное поле в таблицу **cdr** для хранения имени файла. Для этого либо в консоли mysql, либо через phpmyadmin выполните код:

```
alter table `cdr` add column `filename` varchar(120) DEFAULT 'none' after `userfield`;
```

Редактируем файл */etc/asterisk/cdr\_mysql.conf*, добавляя в самый конец 2 строки:

```
alias realdst => realdst  
alias filename => filename
```

Дальше редактируем диалплан. Открываем */etc/asterisk/extensions.conf* и добавляем в секцию globals переменную, соответствующую папке, где будут храниться записи разговоров.

```
[globals]
```

```
DIR_RECORDS=/mnt/calls/
```

Добавляем макрос для записи перед контекстами звонков:

```
[recording]
exten => s,1,Set(fname=${UNIQUEID}-${STRFTIME(${EPOCH},,%Y-%m-%d-%H_%M)}-${ARG1}-${ARG2})
exten => s,n,Set(monopt=nice -n 19 /usr/local/bin/lame -b 32 --silent "${DIR_RECORDS}${fname}.wav"
"${DIR_RECORDS}${fname}.mp3" && rm -f "${DIR_RECORDS}${fname}.wav" && chmod o+r "${DIR_RECORDS}${fname}.mp3")
exten => s,n,Set(CDR(filename)=${fname}.mp3)
exten => s,n,Set(CDR(realdst)=${ARG2})
exten => s,n,MixMonitor(${DIR_RECORDS}${fname}.wav,b,${monopt})
exten => s,n(no),Verbose(Exit record)
exten => s,n,Return()
```

И редактируем контексты входящих и исходящих разговоров, добавляя туда макрос на запись. Он идет самым первым, перед набором номера. Вот так у меня выглядит контекст локальных звонков и входящих через ivr меню с учетом всех сделанных ранее настроек:

```
[call-out]
;Номер для записи звуков, окончание записи #
exten => _35X, 1, NoOp()
exten => _35X, n, Wait(2)
exten => _35X, n, Playback(beep)
exten => _35X, n, Record(/tmp/music${EXTEN:2}:wav)
exten => _35X, n, Wait(1)
exten => _35X, n, Playback(/tmp/music${EXTEN:2})
exten => _35X, n, Wait(2)
exten => _35X, n, Hangup()
;Управление голосовой почтой
exten => 500,1,VoiceMailMain()
```

```
;Звонок на внутренний номер
exten => _XXX,1,GoSub(recording,s,1,(${CALLERID(num)}},${EXTEN}))
exten => _XXX,n,Dial(SIP/${EXTEN},10)
exten => _XXX,n,Set(dstNUM=${EXTEN})
exten => _XXX,n,GotoIf(["${SIPPEER(${EXTEN},status)}" = ""]?num-not-exist,1)
exten => _XXX,n,GotoIf(["${SIPPEER(${EXTEN},status):0:2}" = "UN"]?num-not-connected,1)
exten => _XXX,n,ChanIsAvail(SIP/${EXTEN},s)
exten => _XXX,n,NoOp(===== ChanIsAvail STATUS: ${AVAILSTATUS} =====)
exten => _XXX,n,GoToIf(${AVAILSTATUS} = 2]?num-BUSY,1)
exten => _XXX,n,GoToIf(${AVAILSTATUS} = 3]?num-BUSY,1)
exten => _XXX,n,Goto(num-${DIALSTATUS},1)
exten => num-not-exist,1,Wait(2)
exten => num-not-exist,n,Playback(invalid)
exten => num-not-connected,1,Wait(2)
exten => num-not-connected,n,Playback(ss-noservice)
exten => num-BUSY,1,Wait(2)
exten => num-BUSY,n,Playback(vm-isonphone)
exten => num-NOANSWER,1,Wait(2)
exten => num-NOANSWER,n,Voicemail(${dstNUM},u)
exten => num-CHANUNAVAIL,1,Wait(2)
exten => num-CHANUNAVAIL,n,Playback(vm-isunavail)
;Звонок на внешний номер
exten => _XXX.,1,GoSub(recording,s,1,(${CALLERID(num)}},${EXTEN}))
exten => _XXX.,n,Dial(SIP/${EXTEN}@397945)
include => handup-sip

[ivr-main]
exten => s,1,Answer()
exten => s,2,Background(/etc/asterisk/ivr/ivr-main)
exten => s,3,WaitExten(5)
exten => _XXX,1,GoSub(recording,s,1,(${CALLERID(num)}},${EXTEN}))
exten => _XXX,n,Dial(SIP/${EXTEN},10)
exten => _XXX,n,Set(dstNUM=${EXTEN})
```

```
exten => _XXX,n,GotoIf(["${SIPPEER(${EXTEN},status)}" = ""]?num-not-exist,1)
exten => _XXX,n,GotoIf(["${SIPPEER(${EXTEN},status):0:2}" = "UN"]?num-not-connected,1)
exten => _XXX,n,ChanIsAvail(SIP/${EXTEN},s)
exten => _XXX,n,NoOp(===== ChanIsAvail STATUS: ${AVAILSTATUS} =====)
exten => _XXX,n,GoToIf(${AVAILSTATUS} = 2)?num-BUSY,1)
exten => _XXX,n,GoToIf(${AVAILSTATUS} = 3)?num-BUSY,1)
exten => _XXX,n,Goto(num- ${DIALSTATUS},1)
exten => num-not-exist,1,Wait(2)
exten => num-not-exist,n,Playback(invalid)
exten => num-not-connected,1,Wait(2)
exten => num-not-connected,n,Playback(ss-noservice)
exten => num-BUSY,1,Wait(2)
exten => num-BUSY,n,Playback(vm-isonphone)
exten => num-NOANSWER,1,Wait(2)
exten => num-NOANSWER,n,Voicemail(${dstNUM},u)
exten => num-CHANUNAVAIL,1,Wait(2)
exten => num-CHANUNAVAIL,n,Playback(vm-isunavail)
exten => t,1,GoSub(recording,s,1,(${CALLERID(num)},${EXTEN}))
exten => t,n,Queue(secretary,t)
```

Создаем директорию для записи разговор и назначаем ей права.

```
# mkdir /mnt/calls
# chown -R asterisk. /mnt/calls
```

Перезапускаем астериск для применения всех настроек:

```
# systemctl restart asterisk
```

Осталось только указать настройки подключения к базе данных cdr viewer. Эти настройки находятся в файле *inc/config.php*. Для начала создадим его.

```
# cp /var/www/html/cdr/inc/config/config.php.sample /var/www/html/cdr/inc/config/config.php
```

Задаем там следующие параметры:

```
Mysql
$db_type = 'mysql';
$db_host = 'localhost';
$db_port = '3306';
$db_user = 'asterisk_user';
$db_pass = '12345678';
$db_name = 'asterisk';
$db_table_name = 'cdr';

$system_storage_format = 5;
$system_monitor_dir = '/mnt/calls';
```

Остальные параметры я оставил без изменений. Можно звонить и тестировать запись разговоров, просмотр статистики через Asterisk CDR Viewer Mod. Все записанные файлы складываются в одну папку /mnt/calls. В readme.txt описан пример, как настроить сортировку записей по папкам с датами. Если вам это нужно, сделайте, ничего сложного нет, все рассказано подробно. В web панель можно зайти просто набрав в браузере <http://ip/cdr/>.



### ← ДЕТАЛИЗАЦИЯ ЗВОНКОВ

Просмотр записей о совершенных звонках

**Сортировать по**

- Дата
- Входящий канал
- Номер звонящего
- Имя звонящего
- Номер назначения
- DID (если есть)
- Исходящий канал
- Код аккаунта
- Описание
- Продолжительность
- Приложение
- Статус звонка

**Условия поиска**

С 20 Май 2016 00 : 00 По 31 Май 2016 23 : 59 Выбрать период...

Не  Начинается на  Содержит  Кончается на  Равно  
 Не  Начинается на  Содержит  Кончается на  Равно  
 Не  Начинается на  Содержит  Кончается на  Равно  
 Не  Начинается на  Содержит  Кончается на  Равно  
 Не  Начинается на  Содержит  Кончается на  Равно  
 Не  Начинается на  Содержит  Кончается на  Равно  
 Не  Начинается на  Содержит  Кончается на  Равно

Продолжительность: Между  И  Секунд

Не Любое  
 Не Любой

по убыванию Группировать по: День

По всем условиям  По любому из условий

**Дополнительные опции**

Тип отчета

- Поиск в базе
- CSV файл
- График звонков
- Расход минут
- Параллельные звонки
- ASR и ACD

Плагины

- Расход средств

Кол-во строк

Детализация звонков - найдено 9 звонков

Дата	Статус	Номер звонящего	Номер назначения	Продолжительность	Тариф	Приложение	Вх. канал	Исх. канал	Файл	Описание
2016-05-20 17:20:42	● Отвечено	437678	t	00:24	0.00	Queue	SIP	SIP		
2016-05-20 16:45:52	● Отвечено	437678	t	00:29	0.00	Queue	SIP	SIP		
2016-05-20 16:43:20	● Отвечено	437678	100	00:26	0.00	Dial	SIP	SIP		
2016-05-20 16:41:09	● Отвечено	437678	100	00:19	0.00	Dial	SIP	SIP		
2016-05-20 16:33:57	● Отвечено	100	1111	00:08	0.00	Dial	SIP	SIP		
2016-05-20 16:30:14	● Отвечено	100	1111	00:08	0.00	Dial	SIP	SIP		
2016-05-20 15:31:35	● Отвечено	100	num-not-connected	00:13	0.00	Dial	SIP	SIP		

## Музыка на ожидании (on hold)

Вы можете назначить любую мелодию для проигрывания во время нахождения звонка в очереди. Можете крутить мелодию вместо длинных гудков во время вызова, хотя считаю это лишним. Уместней всего мелодию пускать, как я уже сказал, во время ожидания в очереди и во время удержания звонка по какой-то причине. По-умолчанию в эти моменты проигрывается дефолтная мелодия.

За конфигурацию **music on hold** отвечает соответствующий файл настроек - `/etc/asterisk/musiconhold.conf`. Изначально он выглядит так:

```
[general]
[default]
mode=files
directory=moh
```

Директория указана как `moh`, полный ее путь `/var/lib/asterisk/moh`, мелодии проигрываются отсюда. Я не буду менять настройки по-умолчанию. Добавим еще один класс музыки и укажем его в свойствах очереди, где будем ее проигрывать.

Я привожу пример использования. Если у вас везде будет одна и та же мелодия, то отредактируйте канал `default`. Так вам не придется каждый раз указывать добавленный канал.

Копируем любым способом mp3 файл на сервер в домашнюю директорию `root`. Создаем директорию `/var/lib/asterisk/mohmp3` и кодируем в нее мелодию из mp3 в wav:

```
# mkdir /var/lib/asterisk/mohmp3
# lame --decode /root/music.mp3 /var/lib/asterisk/mohmp3/music.wav
```

Добавляем новый класс в `musiconhold.conf` в самый конец:

```
[mp3]
mode=files
directory=mohmp3
```

Теперь добавим эту мелодию в свойства очереди. Для этого в описание очереди добавьте новый параметр:

```
[secretary]
music = mp3
strategy = ringall
member => SIP/100
member => SIP/130
```

Теперь нужно перечитать настройки очереди и мелодии. Проще перезапустить сам астериск:

```
# systemctl restart asterisk
```

Можно звонить на внешний номер и ждать попадания в очередь. Вы должны услышать добавленную мелодию во время ожидания.

Чтобы добавить эту мелодию вместо длинного гудка, необходимо отредактировать dialplan, добавив новый параметр в правило набора. Если добавляете свой класс музыки, то указываете его в свойствах:

```
exten => _XXX,1,Dial(SIP/${EXTEN},15,m(mp3))
```

Если используете музыку по-умолчанию, то достаточно написать вот так:

```
exten => _XXX,1,Dial(SIP/${EXTEN},15,m)
```

Для отладки музыки на ожидании, я рекомендую добавить в dialplan в контекст локальных звонков такую конструкцию:

```
exten => 336,1,Answer  
exten => 336,2,MusicOnHold()
```

336 - любой не занятый реальным пиром номер. При звонке на этот номер, вы услышите заданную дефолтную мелодию в соответствии с настройками в файле *musiconhold.conf*. Для проигрывания не дефолтного класса, укажите его имя в скобках. Так вы можете прослушать как звучат ваши мелодии, протестировать порядок проигрывания и т.д.

Не забывайте перезагружать диалплан после редактирования. На этом настройка music on hold закончена. Можете добавлять разные мелодии и использовать в необходимых местах диалплана или очередях.

## Настройка конференций

Полезным и востребованным функционалом офисной телефонной станции является создание конференций для одновременного разговора нескольких человек. В asterisk конференция настраивается следующим образом. В специальном файле конфигурации задаются параметры комнаты для конференций. В диалплане назначается отдельный номер для комнаты с конференцией.

Создаем конфигурацию конференции. Для этого в файл *confbridge.conf* добавляем в самый конец:

```
[confer]  
type=bridge  
max_members=20  
mixing_interval=10  
internal_sample_rate=auto
```

```
record_conference=yes
```

В контекст исходящих звонков [call-out] добавляем в самое начало:

```
exten => 999,1,Answer()  
exten => 999,n,ConfBridge(1,confer)
```

Перезапускаем астериск. Теперь при звонке на номер 999 человек попадает в комнату для конференций. Следующим участникам конференции необходимо будет так же позвонить на номер 999 для участия в совместной конференции. Если вы с кем-то уже разговариваете, хотите создать конференцию и добавить туда еще участников, то необходимо вашего собеседника переадресовать на номер конференции, затем позвонить новому абоненту и его направить туда же. После этого звоните сами на номер конференции и сможете общаться с тремя абонентами одновременно.

Как выполнять переводы и переадресации звонков я расскажу ниже.

## Перевод, перехват, переадресация звонка

Рассмотрим несколько необходимых функций и возможностей телефонной станции asterisk. Начнем с перевода звонка. Как перевести звонок на другого абонента? По-умолчанию в астериске для трансфера звонка предусмотрена клавиша #. Посмотреть, так ли это в вашей конфигурации, введите в консоли астера команду:

```
CLI> features show  
Builtin Feature          Default Current  
-----  
Pickup                   *8           *8  
Blind Transfer         #            #  
Attended Transfer  
One Touch Monitor  
Disconnect Call         *            *
```

## Park Call

Во время разговора нужно нажать **#** и набрать номер, куда вы хотите перевести звонок. Для того, чтобы трансфер состоялся, он должен быть разрешен в диалплане в команде Dial следующим образом:

```
exten => _XXX,n,Dial(SIP/${EXTEN},15,Tt)
```

T дать возможность звонящему (вызывающему) абоненту совершать перевод звонка на другой номер.

t дать возможность вызываемому абоненту сделать перевод звонка на другой номер

В предыдущих примерах я не указывал этот параметр. Сейчас пришло время рассказать о нем и добавить во все участки диалплана, если вам необходимо осуществлять перевод звонков в asterisk.

Перевод звонка можно сделать с помощью функции самого телефона. Мне еще не попадалось ни одного телефона, который бы не умел переводить звонок. Обычно для этих целей предназначена отдельная клавиша или функция в меню телефона. Для уточнения информации о переводе нужно воспользоваться инструкцией самого телефона. В софтфоне, которым я пользуюсь, есть отдельная клавиша для перевода звонка:



С включенным по-умолчанию **Blind Transfer** есть проблемы. Например, вы перенаправляете звонок на другой номер, а он не ответил или у него занято. Входящий вызов сбрасывается. В астериске есть другой режим перевода звонка, он называется **Attended Transfer**, по-умолчанию он выключен. Работает он более изящно. Сначала вы звоните тому, куда хотите перевести номер, разговариваете с ним, убеждаетесь, что он доступен и готов принять звонок. Только после этого переводите звонок на этого абонента. Чтобы включить такое перенаправление вызова, нужно раскомментировать в `features.conf` строку:

```
atxfer => *2
```

Горячую клавишу можно переназначить на любую другую. На этом о переводе звонка в астериске все.

Теперь поговорим о перехвате звонка. Хотя говорить тут особо нечего. В asterisk перехват работает из коробки и не требует никаких настроек. Для того, чтобы перехватить звонок, нужно снять трубку и нажать комбинацию **\*8**. Комбинация задается в том же файле, что и перехват - *features.conf*. Перехватывать звонки могут только абоненты в одной группе. Это задается в свойствах пользователя, я об этом рассказывал в самом начале, когда мы создавали sip аккаунты. Напомню, что речь идет о параметрах **callgroup** и **pickupgroup**.

Разберем теперь переадресацию звонка на какой-то внешний номер, например, мобильный телефон. Многие телефоны имеют встроенный функционал по перенаправлению звонка. Читаете инструкцию к телефону, смотрите, можно ли на нем установить переадресацию и как, и делаете. Работает это только если ваш телефон подключен к АТС. Он принимает звонок и сам его переадресовывает на указанный номер. Это самый простой и быстрый вариант автоматической переадресации звонка на внешний номер.

Рассмотрим другой случай переадресации на мобильный телефон. Допустим, у нас есть номер 115, который вообще отсутствует в офисе, у него нет стационарного телефона. Нам нужно, чтобы при звонке на этот номер, вызов переадресовывался на внешний номер, а конкретно на мобильный сотрудника. Для этого в dialplan нужно добавить следующую строку:

```
exten => 115,n,Dial(SIP/89151234567@397945)
```

Если вы хотите, чтобы переадресация работала только при звонках с внутренних номеров офиса, то добавить эту строку нужно в контекст [call-out] перед общим правилом набора на внутренние номера:

```
exten => 115,n,Dial(SIP/89151234567@397945)  
exten => _XXX,n,Dial(SIP/${EXTEN},15,Tt)
```

Для того, чтобы переадресация на внешний номер работала и для звонков из вне, правило перенаправления на мобильный нужно поставить в контекст с голосовым приветствием [ivr-main]. Ставим туда же, перед строкой набора на внутренние номера:



```
exten => 115,n,Dial(SIP/89151234567@397945)  
exten => _XXX,n,Dial(SIP/${EXTEN},15,Tt)
```

Вы можете поменять внешний номер для разных контекстов. К примеру, для сотрудников офиса сделать перенаправление на один мобильный, а для звонков клиентов на другой.

## Защита asterisk с помощью fail2ban

Если ваш астериск имеет внешний ip адрес, либо на него сделан проброс портов с внешнего ip, то необходимо обеспечить защиту от перебора учеток и прочих множественных подключений. Организуем защиту с помощью известного и популярного средства **fail2ban**. Сделать это не сложно, fail2ban поддерживает asterisk из коробки. Достаточно просто установить его и активировать некоторые настройки.

Устанавливаем fail2ban на сервер с астериском:

```
# dnf install fail2ban
```

Включаем запись в лог файл событий типа security. Для этого открываем файл `/etc/asterisk/logger.conf` и раскомментируем строку:

```
security => security
```

Перечитываем настройки хранения логов:

```
# asterisk -x "logger reload"
```

В папке `/var/log/asterisk` появился новый файл `security`. Его записи мы будем передавать в fail2ban для анализа. Открываем файл `/etc/fail2ban/jail.conf`, ищем там секцию `[asterisk]` и меняем путь к логфайлу и добавляем строку активации джейла:

```
logpath = /var/log/asterisk/security
enabled = true
```

Запускаем fail2ban и добавляем в автозапуск:

```
# systemctl start fail2ban
# systemctl enable fail2ban
```

Чтобы fail2ban работал, у вас должен быть запущен и настроен iptables. Отмечу также, что по-умолчанию fail2ban в centos использует команды firewalld. Если вы его отключили и используете голые iptables, то вам необходимо удалить файл `/etc/fail2ban/jail.d/00-firewalld.conf` и перезапустить fail2ban. Если этого не сделать, работать он не будет.

На этом подробная и многофункциональная настройка asterisk закончена. Я рассказал все, что запланировал, выделив наиболее востребованные функции.

## Заключение

Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоём распоряжении. Расскажи, как сделать правильно!

Время подвести итог проделанной работы. На всякий случай приведу полный конфиг основного файла, с которым мы работали - [extensions.conf](#). Я писал статью почти 2 месяца, что-то добавляя, проверяя, редактируя. Она актуальна полностью на момент написания. Можно простым копипастом переносить конфигурацию и все заработает. Я проверил перед публикацией. Со временем что-то может измениться, но не думаю, что сильно. Синтаксис конфигурационных файлов почти не меняется в разных версиях астера, это позволяет без проблем переносить настройки между версиями, и тем более между операционными системами.

Я планирую раскрыть более широко затронутую тему. У меня есть примеры интересных и полезных конфигураций. По мере возможности буду писать новые статьи по серверу телефонии. На сегодня у меня все, хороших вам настроек :-)

Напоминаю, что данная статья является частью единого цикла статьей про сервер Centos.

Другие материалы по asterisk:

- Установка и настройка астериск с нуля
- Мониторинг asterisk в zabbix
- Соединение двух серверов астериск
- Анализ SIP трафика
- Замена +7 на 8 и наоборот
- Ограничение на звонки для группы номеров

Если у вас есть желание научиться строить и поддерживать высокодоступные и надежные системы, рекомендую познакомиться с **онлайн-курсом «Administrator Linux. Professional»** в OTUS. Вы научитесь делать профессиональный подбор конфигурации, управлять процессами, обеспечивать безопасность, выполнять развертывание, настройку и обслуживание сетей. Проверьте себя на вступительном тесте и смотрите подробнее программы ссылке.

Помогла статья? Подписывайся на telegram канал автора

Анонсы всех статей, плюс много другой полезной и интересной информации, которая не попадает на сайт.