Никоноров М.Ю, Ефременко А.В

Телефония Asterisk с нуля

Подробное пошаговое руководство

Приложение к одноименному видеокурсу

Оглавление

1. Устанавливаем ОС Linux сборки CentOS

В моем случае это была Linux сборка CentOS 6.4. Использовать будем дистрибутив minimal - он без графической оболочки Gnome. То есть работать будем полностью через интерфейс CLI, то есть через командную строку

a) Скачиваем дистрибутив отсюда: http://vault.centos.org/

b) Устанавливаем Linux

c) Сразу после установки пишем команду:

```
ifconfig -a  (показывает сетевые адаптеры)
```

Если ОС видит сетевой адаптер, то он отобразится. У меня он назывался eth0. Но беда - нет ip адреса.

Для того, чтобы ОС прицепила ip адрес к сетевой карте пишем команду:

```
ifup eth0 (определяет ip интерфейса eth0)
```

Снова пишем команду ifconfig -a и видим свой ip адрес для интерфейса eth0.

Внимание – после перезагрузки придется снова определять ip адрес.

Для того, чтобы ip адрес цеплялся автоматически при старте CentOS, мы выполняем следующие действия:

а) устанавливаем текстовый редактор nano (на подобии блокнота в Windows)

```
yum install nano
```

б) Затем пишем следующую команду:

```
nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

и в появившемся файле переменную ONBOOT=”no” меняем на ONBOOT=”yes”

Как пользоваться редактором nano: http://habrahabr.ru/post/106554/

2. Установка Putty

Итак, мы установили операционную систему, определили ip адрес, теперь нужно поставить саму систему.

Удобнее будет работать через putty, ибо приятнее шрифт и можно работать с буфером обмена.

Скачиваем, запускаем putty и цепляем его к нашей centOS. Скачать можно здесь:

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

3. Установка Asterisk

Полная статья по установки Asterisk здесь:

http://www.voip-info.org/wiki/view/Asterisk+11+Installation+on+CentOS+6

Устанавливать будем Asterisk 11.0.0

Кратко:

a) Отключаем улучшенную систему безопасности SELinux:

```
sed -i s/SELINUX=enforcing/SELINUX=disabled/g /etc/selinux/config
```

b) Установка необходимых компонентов для установки Asterisk:

```
yum install -y make wget openssl-devel ncurses-devel newt-devel libxml2-devel kernel-devel gcc gcc-c++ sqlite-devel
```

c) Загружаем исходный код Asterisk. Для этого переходим в папку:

```
cd /usr/src/
```

и загружаем с помощью команды wget:

```
wget http://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-complete-current.tar.gz
wget http://downloads.asterisk.org/pub/telephony/libpri/libpri-1.4-current.tar.gz
wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-11-current.tar.gz
```

d) Распаковываем скаченные архивы:

```
tar zxvf dahdi-linux-complete*
tar zxvf libpri*
tar zxvf asterisk*
```

e) Устанавливаем LibPRI

```
cd /usr/src/libpri*
make && make install
```

f) Переходим в директорию, в которую распаковался Aterisk:

```
cd /usr/src/asterisk*
```

Кстати, находясь в /usr/src/ можно набрать ls и посмотреть как конкретно называется директория, в которую распаковался asterisk

g) Запускаем конфигурационные скрипты для Asterisk. Для этого, сначала узнаем какой битности наш Asterisk.

Набираем:

```
uname -a
```

Если ответ: 2.6.18-238.12.1.el5 #1 SMP Tue May 31 13:23:01 EDT 2011 i686 i686 i386 GNU/Linux то значит 32 бита

Если ответ: 2.6.18-238.19.1.el5 #1 SMP Fri Jul 15 07:31:24 EDT 2011 x86_64 x86_64 x86_64 GNU/Linux то значит 64 бита.

В зависимости от того, какова битность Asterisk, запускаем конфигурационный скрипт:

Для 32 бита:

```
./configure && make menuselect && make && make install
```

Для 64 бита:

```
./configure --libdir=/usr/lib64 && make menuselect && make && make install
```

Внимание! Может возникнуть проблема, при которой после выполнения последней команды возникнет ошибка. Будет ругаться на .xml файл. Тогда необходимо добавить строчку к этой команде, после чего для 64-бита будет выглядеть так:

```
./configure --(команда, которую предлагает астериск) --libdir=/usr/lib64 && make menuselect && make && make install
```

Об успешной установки свидетельствует синие окно

h) Добавляем поддержку звонков. Дело в том, что при такой конфигурации Asterisk вроде бы как работает, но звонки совершаться не будут. Будет возникать ошибка

[Apr 27 21:35:51] ERROR[1225][C-00000009]: rtp_engine.c:259 ast_rtp_instance_new: No RTP engine was found. Do you have one loaded?

Поэтому, делаем следующее:

```
yum install uuid uuid-devel libuuid libuuid-devel uuid-c++
```

после этого:

```
./configure
make menuselect
```

и потом

```
make
make install
```

(полная статья про это дело здесь: http://forums.asterisk.org/viewtopic.php?f=1&t=86518

i) Далее устанавливаем образцы. Без установки этих образцов у нас не появятся конфигурационные файлы sip.conf и extensions.conf

```
make samples
make config
```

j) пишем

```
cd
```

и затем пишем

```
reboot
```

k) После перезагрузки пишем

```
asterisk
```

В результате Asterisk должен запуститься. Поздравляю! Мы запустили asterisk.

4. Конфигурация Asterisk для совершения звонков между внутренними абонентами

a) Теперь необходимо отключить фаерволл в самой CentOS. Без этого софтовый телефон X-Lite не хочет цепляться к серверу Asterisk.

Полная статья по этому делу: http://www.sl-s.ru/kak-otklyuchit-firewall-v-centos-redhat/

Для этого пишем следующие команды:

```
service iptables save
service iptables stop
chkconfig iptables off
```

b) Переходим непосредственно к редактированию sip.conf:

```
nano /etc/asterisk/sip.conf
```

У нас открывается файл. Пишем свои конфиги в самое начало файла. В моем случае это определение двух sip клиентов (телефонов):

```
[1001]
type=friend
regexten=1001
secret=1234
context=outcoling
host=dynamic
callerid="1001" <1001>
disallow=all
allow=alaw
allow=ulaw
language=ru
callgroup=1
pickupgroup=1
qualify=yes
```

```
canreinvite=yes
call-limit=4
nat=no

[1002]
type=friend
host=dynamic
insecure=invite
username=1002
secret=45678
context=outcoling
disallow=all
allow=alaw
```

После этого находим секцию [general] и удаляем её. Так же удаляем надпись «context=public» после надписи [general].

Обращаем внимание на контексты.

Для телефонов (sip клиентов) [1001] и [1002] это outcoling.

Что такое контекст и зачем он нужен? Контекст связывает файл sip.conf с файлом extensions.conf. То-есть если у [1001] прописан контекст outcoling, то [1001] будет искать правило в extensions.conf под названием outcoling.

Нажимаем ctrl+x, нажимаем y и нажимаем enter. Файл сохранен.

Подробная статья по этому делу:

http://wiki.zadarma.com/index.php/Asterisk (настройка транка для задармы)

http://habrahabr.ru/post/122898/ - простая настройка sip клиентов

c) Переходим к редактированию extensions.conf

```
nano /etc/asterisk/extensions.conf
```

В конце файла пишем наш диал план:

```
[outcoling]
exten => _XXXX,1,Dial(SIP/${EXTEN},,m)
```

После чего сохраняем файл.

Статьи по этому делу:

http://www.asterisk.by/node/153 (основы основ Dial плана)

http://habrahabr.ru/post/122898/ (пример простого экстеншена) в середине файла

http://asterisk-pbx.ru/wiki/doku.php/asterisk_dialplan (пример простого экстеншена)

http://asterisk.ru/knowledgebase/Asterisk+config+extensions.conf

d) Далее пишем

```
asterisk -r
```

Это мы из управления Linux перешли в управление Asterisk (как будто бы открыли окно программы и начали с ней работать)

Пишем

```
core reload
```

Тем самым мы заставили asterisk пересчитать все конфигурационные файлы и принять изменения.

Все. Коннектим софтфон к астериску и пробуем совершить первый телефонный звонок между двумя внутренними абонентами.

5. Конфигурация Asterisk на работу через транк (принимаем звонки с внешних телефонов)

a) Заключаем договор с sip провайдером (провайдером ip телефонии). Получаем от него данные. В моем случае это zadarma. Как это делается показано в видео уроке.

b) Заходим в файл sip.conf

```
nano /etc/asterisk/sip.conf
```

И над нашими sip клиентами [1001] и [1002] пишем следующий код:

```
[general]
register => 00000:password@sip.zadarma.com/00000

[zadarma]
type=friend
username=00000
secret=password
fromuser=00000
fromdomain=sip.zadarma.com
host=sip.zadarma.com
nat=yes
insecure=invite
context=incoming
canreinvite=no
```

После того, как мы вставили этот текст, ниже найдем еще один [context] и удалим его. Так после [context] будет строка: context=public – удалим её.

c) Сохраняем файл и заходим в файл extensions.conf

```
nano /etc/asterisk/extensions.conf
```

К контексту [outcoling] добавляем следующие:

```
exten => _XXXXXXXXXX,1,Dial(SIP/zadarma/${EXTEN})
```

Кроме того, после контекста [outcoling] добавляем еще один контекст:

```
[incoming]
exten => _X.,1,Dial(SIP/1001&SIP/1002,60,m,tT)
```

Все вместе это будет выглядеть так:

```
[outcoling]
exten => _XXXXXXXXXX,1,Dial(SIP/zadarma/${EXTEN})
exten => _XXXX,1,Dial(SIP/${EXTEN},,m)

[incoming]
exten => _X.,1,Dial(SIP/1001&SIP/1002,60,m,tT)
```

d) Сохраняем файл и пишем:

```
asterisk –r
```

далее пишем команду:

```
core reload
```

после чего напишем:

```
sip show registry
```

Таким образом мы проверяем «поднялся ли транк», то есть определяем соединился ли наш Asterisk с провайдером ip телефонии. Если все ОК, в ответ на команду sip show registry мы получим ответ:

```
1 SIP REGISTRATION
```

e) Пробуем совершить звонок из Asterisk например на мобильный телефон и с мобильного телефона на Asterisk.

6. Реализация функций Asterisk

6.1 Установка музыки вместо гудка.

Для того, чтобы человек, который нам звонит слышал музыку, а не гудок, нам необходимо взять MP3 файл и перекодировать его в WAV кодеком G.711 U-Law. Должен быть 8-bit моно. Для этого:

a) Скачиваем и устанавливаем программу Ease Audio Converter. Скачать можно отсюда: http://audiotool.net/EaseAudioConverter/index.htm

b) В программе нажимаем кнопочку Setting, в появившемся окне выбираем формат wav и выставляем:

частота: 8000

каналы: моно

Выходной формат: 16 bit WAVE (PCM)

Выбираем mp3 файл, который хотим перекодировать и перекодируем.

Статья по этому делу: http://www.ask.com/explore/convert-mp3-g711-ulaw-3865

Наглядно этот процесс продемонстрирован в видео

c) Закачиваем полученный файл в Asterisk. Для этого:

1) скачиваем программу pscp http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

2) Кидаем программу на рабочий стол, но запускать её не надо

3) Запускаем cmd.exe в windows и пишем:

```
desktop\pscp.exe D:\test.txt remote_user@remote_host:/path_to_destination
```

path_to_destination пускай будет папка mymusic, которую мы предварительно создадим в linux:

```
mkdir /var/lib/asterisk/moh/mymusic
```

Так что если наш mp3 файл называется Jessi.wav и лежит в корне диска D, то команда будет такой:

```
desktop\pscp.exe D:\Jessi.wav root@ip адрес CentOS:/var/lib/asterisk/moh/mymusic
```

В результате файл Jessi.wav из Windows должен упасть в CentOS по адресу:

```
 /var/lib/asterisk/moh/mymusic
```

Статья по этому делу: http://alegenk.livejournal.com/19231.html

d) Теперь уже с помощью Asterisk нам необходимо перекодировать файл с помощью кодека U-Law. Для этого, в консоли пишем:

```
asterisk-r
```

далее

```
file convert youraudio.wav youraudio.ulaw
```

т.е для нашего конкретного случая будет выглядеть так:

```
file convert /var/lib/asterisk/moh/mymusic/Jessi.wav Jessi.ulaw
```

Статья по этому делу: http://striker24x7.blogspot.ru/2012/02/wavmp3-to-g729-ulaw-alaw-gsm-converter.html

e) Теперь настраиваем конфиги Asterisk и редактируем файл musiconhold.conf

```
nano /etc/asterisk/musiconhold.conf
```

Внутри файла находим следующее:

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

Меняем значение directory на

```
directory=moh/mymusic
```

f) Сохраняем файл

Статья по этому делу: http://cs.stu.cn.ua/post/413/

Если мы положим в папку mymusic и другие файлы, они будут проигрываться по очереди, то один, то другой.

Если мы хотим для определенного экстеншена указать одну конкретную песню, мы можем воспользоваться статьей по этому делу:

http://www.hilik.org.ua/asterisk-%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%B0-dial-misic-on-hold/

Поскольку в нашем случае у нас только один музыкальный файл в папке mymusic, то больше ничего не требуется.

g) Последнее, что осталось, это настроить extensions.conf. Если вы воспользовались моим примером:

```
[outcoling]
exten => _XXXXXXXXXXX,1,Dial(SIP/zadarma/${EXTEN})
exten => _XXXX,1,Dial(SIP/${EXTEN},,m)

[incoming]
exten => _X.,1,Dial(SIP/1001&SIP/1002,60,m,tT)
```

то музыка уже будет проигрываться. За это отвечает буква m

6.2 Создание интерактивного (голосового) меню.

a) Прежде всего необходимо записать голосовое сообщение и получить mp3 файл. Например с помощью Nero WaveEditor

b) Затем необходимо выполнить перекодировку файла с помощью программы Ease Audio Converter, подобно тому, как мы делали это для установки музыки вместо гудка

c) Затем загружаем полученный файл после конвертации в папку, но предварительно создаем её

```
mkdir /var/lib/asterisk/moh/voicemail
```

d) Перекодируем загруженный в CentOS файл с помощью Asterisk:

```
localhost*CLI> file convert /var/lib/asterisk/moh/voicemail/название файла.wav название файла.ulaw
```

e) Создаем новый номер 7777, к которому не будет подключен никакой телефон, но который будет использоваться для того, чтобы эмитировать звонок с внешки (дабы не тратить деньги):

```
nano  /etc/asterisk/sip.conf
```

и добавляем:

```
[7777]
type=friend
host=dynamic
insecure=invite
username=7777
secret=1213
context=outcoling
disallow=all
allow=alaw
```

f) Теперь займемся самым главным - extensions.conf

```
nano  /etc/asterisk/extensions.conf
```

затираем все, что делали там ранее, и вместо это пишем:

```
[incoming]
exten => _X.,1,Goto(menu,s,1)  ;если нам кто-то звонит, то входящий звонок из файла sip.conf поступает на этот контекст. После чего звонок переадресовывается с помощью функции Goto на котекст menu


[outcoling]
exten => _XXXXXXXXXX,1,Dial(SIP/zadarma/${EXTEN})
exten => _XXXX,1,Dial(SIP/${EXTEN},,m)
exten => 7777,1,Goto(menu,s,1) ;если мы изнутри позвоним на этот номер, то мы сможем проверить работу нашего голосового меню. Благодаря этой строчки нет необходимости для проверки звонить постоянно с внешки


[menu]
exten => s,1,Background(/var/lib/asterisk/moh/voicemail/voicemenu) ;здесь ловится звонок из контекста incoming и проигрывается записанное нами приветствие. Не надо указывать расширение файла, достаточно указать само имя файла с записанным голосом
exten => 1,1,Dial(SIP/1001,,m) ;если человек нажал цифру 1, то звоним нашему внутреннему абонентку 1002
exten => 2,1,Dial(SIP/1002,,m) ;если человек нажал цифру 2, то звоним нашему внутреннему абонентку 1005
exten => s,n,Wait(5) ;если человек не нажал ничего, ждем 5 секунд и
exten => s,n,Dial(SIP/1001&SIP/1002,,m) ; тогда звоним сразу двум абонентам
```

То, что выделено синим – комментарии.

6.3 Перенаправление звонков

Случается так, что например секретарь получил звонок и секретарю этот звонок нужно направить, например менеджеру. Для включения этой функции выполняем следующее действие:

в extensions.conf в Dial добавляем параметр t. Этот параметр означает, что для этого Dial плана разрешено перенаправление звонков.

У меня это выглядит так:

```
exten => _XXXX,1,Dial(SIP/${EXTEN},,t&m,)
```

Здесь мы видим параметр t&m. То-есть сразу два параметра - t для перевода звонков и m для музыки.

Теперь нужно разобраться в понятиях blind transfer и attended transfer.

Blind transfer используется для слепого перевода звонков и работает по умолчанию. Что означает слепой перевод звонков?

Это когда секретарь переводит звонок менеджеру и секретарю все равно, что случится со звонком дальше. К ней этот звонок уже никогда не вернется. То есть это простая переадресация, без обратной связи.

Когда кто-то позвонил и БЫЛО УСТАНОВЛЕННО СОЕДИНЕНИЕ с секретарем, то секретарь нажимает на #, вводит внутренний или любой внешний номер телефона, на который хочет перевести звонящего абонента (ну например на менеджера) и все.

Другое дело attended transfer. Attended transfer позволяет секретарю не просто перевести звонок, но и контролировать успешность его перевода. Представим ситуацию: секретарь перенаправляет звонок менеджеру. Если менеджер не отвечает в течении заданного количества времени, или менеджер просто нажал на красную трубочку (сбросил), то звонок возвращается обратно к секретарю. А там секретарь уже скажет - извините, менеджера сейчас нет или он занят.

Давайте реализуем. Итак, в контексте для Dial мы задали параметр t. Теперь у нас работает слепой трансфер (blind transfer) через #.

Для Attended transfer все сложнее.

1. Переходим к редактированию файла features.conf

```
nano /etc/asterisk/features.conf
```

2. Находим строчки:

```
;atxfernoanswertimeout = 15     ; Timeout for answer on attended transfer default is 15 seconds.
;atxferdropcall = no            ; If someone does an attended transfer, then hangs up before the transferred
                                ; caller is connected, then by default, the system will try to call back the
                                ; person that did the transfer.  If this is set to "yes", the callback will
                                ; not be attempted and the transfer will just fail.
                                ; For atxferdropcall=no to work properly, you also need to
                                ; define ATXFER_NULL_TECH in main/features.c.  The reason the
                                ; code is not enabled by default is spelled out in the comment
                                ; block near the top of main/features.c describing ATXFER_NULL_TECH.
;atxferloopdelay = 10           ; Number of seconds to sleep between retries (if atxferdropcall = no)
;atxfercallbackretries = 2      ; Number of times to attempt to send the call back to the transferer.
                                ; By default, this is 2.
```

и раскомментируем их (убираем двоеточие в начале файла).

Кроме того, раскомментируем строчку:

```
;atxfer => *2
```

Эта строчка разрешает использование Attended transfer

Теперь пробуем совершить звонки и переадресацию.

1) Звоним с мобильника на астериск на номер секретаря.

2) Устанавливаем соединение. Секретарь говорит собеседнику, чтобы подождал, пока она переключит

3) Секретарь набирает на телефоне *2 и номер, на который она хочет перекинуть. Ну например 1002

4) Номер 1002 берет трубку. Секретарь спрашивает у номера 1002 хочет ли он разговаривать. Если хочет, то секретарь кладет трубку

5) После того, как секретарь положил трубку, звонок уже долбится к менеджеру (1002). Если менеджер не ответит в течении 15 секунд или повесит трубку, нас снова отошлют к секретарю.

Обратите внимание на пункт 4. Если номер 1002 не ответил или сбросил вызов, секретарь снова начнет разговаривать с мобильником (скажет что менеджер сейчас занят или отсутствует)

Сложно для восприятия!? Просто поэкспериментировать и все станет понятно!

И не забываем, что так можно перенаправлять не только на внутренние номера, но и на любые внешние, например на мобильники.

Статьи на эту тему:

http://subscribe.ru/archive/comp.soft.linux.cfgsoftunix/201004/19084218.html - перенаправление звонков

http://asterisk-support.ru/question/39046/attended-transfer-i-blind-transfer-odnoi-knopkoi/ - некие дискуссии по поводу blind transfer и attended transfer

http://asterisk-support.ru/question/36797/ne-vozvrashchaiutsia-pereadresovannye-vyzovy/ - дискуссия на тему "не возвращаются перенаправленные звонки"

http://igorg.ru/2008/03/29/za-transfer-zamolvite-slovo/ - а это перенаправление звонков не с помощью встроенной функции Asterisk, регулирующийся через features.conf, а через

Dial plan (В старых версиях астера не было встроенной функции перевода звонков)

6.4 Запись разговоров

а) Первым делом, необходимо определить папку, куда будут записываться и складываться звонки. Для того, чтобы лучше видеть структуру папок в вашем Linux, предлагаю установить Midnight Commander.

Для этого:

```
yum install mc
```

После чего, для запуска:

```
mc
```

Появляется программа, очень похожая на Norton Commander. Теперь мышкой можно лазить по папкам как в Windows и подумать, куда лучше складывать звонки. Я решил на корневом диске создать папку records. Поэтому, в Norton Commander переходим на самый верхний уровень и под Norton Commander пишем команду:

```
mkdir records
```

Наглядная работа с Norton Commander (что означает перейти на самый верхний уровень) показана в видео уроке.

Видим, что у нас появилась папка records. Заходим туда. Далее в папке records создадим еще одну папку callrecords

```
mkdir callrecords
```

Все. Сюда мы и будем складывать наши записанные разговоры.

b) Запись звонков настраивается все в том же dial плане, все в том же extensions.conf. Пример экстеншена для записи звонков выглядит следующим образом:

```
exten => _8.,1,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN}) ;для всех звонков, начинающихся с 8 (входящих или исходящих, не важно, здесь это не указано) начинается определение переменной fname. В неё закладывается текущий год, месяц и число, а так же кто звонит и кому звонит

exten => _8.,2,MixMonitor(/home/share/monitor/${fname}.wav) ;функция MixMonitor начинает запись звонка и сохраняет файл по указанному пути

exten => _8.,3,Dial(SIP/prov1) ;совершается звонок.
```

Это простой общий пример. Ниже будет приведет мой конкретный Dial план:

```
[incoming]

exten => _X.,1,Goto(menu,s,1)


[outcoling]

exten => _X.,1,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN}) ;_X., означает, что для ЛЮБЫХ исходящих номеров начинает определятся название файла

exten => _X.,2,MixMonitor(/records/callrecords/${fname}.wav) ; _X., означает, что для ЛЮБЫХ исходящих номеров начинается запись файла и сохраняется по пути, который мы создали в нашем linux: /records/callrecords/

exten => _XXXXXXXXXX,3,Dial(SIP/zadarma/${EXTEN})

exten => _XXXX,3,Dial(SIP/${EXTEN},,t&m,)

exten => 7777,1,Goto(menu,s,1);


[menu]

exten => s,1,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN}) ;буква s в данном случае означает, что нет точного определения в каком конкретном случае начнется определение имени файла. Эта строка просто начинает работать сама по себе как только вызывается экстеншен [menu]

exten => s,2,MixMonitor(/records/callrecords/${fname}.wav)

exten => s,3,Background(/var/lib/asterisk/moh/voicemail/voicemenu)

exten => 1,1,Dial(SIP/1001,30,m&t)

exten => 2,1,Dial(SIP/1002,30,m&t)

exten => s,4,Wait(5)

exten => s,5,Dial(SIP/1001&SIP/1002,30,t&m)
```

То есть мы здесь прописали запись разговоров для 2-х случаев:

1) Когда мы звоним (контекст outcoling)

2) И когда нам звонят (контекст menu). А контекст menu, в свою очередь вызывается из контекста incoming

c) Теперь нам необходимо прослушать эти разговоры. Лучше всего будет прослушать из Windows. Для того, чтобы мы могли прослушивать полученные файлы из Windows, нам необходимо расшарить папку records, которую мы создали в Linux.

Для того, чтобы расшарить папку в Linux, необходимо установить и настроить сервер Samba, который и будет управлять протоколом Samba (именно этот протокол Windows использует, когда мы расшариваем папки. Но в Windows это все уже установлено по дефолту, а вот в Linux надо установить принудительно)

1) Пишем команду:

```
yum install samba
```

2) После установки правим конфигурационный файл сервера samba:

```
nano /etc/samba/smb.conf
```

Удаляем оттуда все, и пишем то, что предлагаю я:

```
# smb.conf is the main Samba configuration file. You find a full commented
# version at /usr/share/doc/packages/samba/examples/smb.conf.SUSE if the
# samba-doc package is installed.
# Date: 2008-06-06
[global]
workgroup = WORKGROUP
server string = Samba Mega Server %v
hosts allow = ALL
# ———————— Logging Options ————————————
log file = /var/log/samba/%m.log
# max 50KB per log file, then rotate
max log size = 1024
# ————————— Standalone Server Options ————————
security = share
#encrypt passwords = yes
socket options = TCP_NODELAY SO_SNDBUF=8192 SO_RCVBUF=8192 IPTOS_LOWDELAY
# ————————— Browser Control Options ————————
local master = yes
os level = 255
```

```
preferred master = yes
# ————————————— Name Resolution ——————————————
dns proxy = yes
# ——————————————Charsets———————————————————-
unix charset = utf8
dos charset = cp1251
display charset = cp1251
# ——————————————-Share Definitions ——————————————
[share]
comment = records
path = /records #здесь указывается папка, которую мы расшариваем
browseable = yes
writable = yes
guest ok = yes #позволяет подключаться к папке кому угодно, без аутентификации
```

(если не будет доступа к папке – удалите комментарии)

3) Стартуем сервер:

```
/etc/init.d/smb start
```

4) Добавляем его в автозагрузку:

```
chkconfig smb on
```

d) Теперь в Windows запускаем приложение "Выполнить" и пишем \\ip_нашего_Linux_сервера. Напоминаю, что ip нашего Linux можно узнать набрав в Linux команду ifconfig

Все! Теперь мы заходим через Windows в нашу расшаренную папку и видим там все наши записанные разговоры в папке callrecords

Кстати, кроме MixMonitor, есть функция просто Monitor. Благодаря ей, голос двух собеседников (того кто звонит и того кому звонят) можно записывать в разные файлы (в MixMonitor голоса двух собеседников записываются в один файл)

Но мне кажется это бессмысленно - тебе нужно прослушать разговор, ты открываешь один файл и слушаешь сразу двух собеседников). В статье "запись телефонных разговоров", ссылка на которую приведена ниже, есть информация по функции Monitor.

Статьи на эту тему:

http://bloglinux.ru/2011/06/26/kak-rassharit-papki-na-mashine-s-linux-dlya-se/ - установка Samba

http://www.samba.org/samba/docs/using_samba/ch09.html - более глубокая настройка Samba сервера. Назначение прав доступа на папки, добавление пользователей (аналог локальным политикам и группам

в Windows, только через конфиг)

http://sys.dmitrow.com/node/189 - запись телефонных разговоров

6.5 Простой автоответчик

Этот пункт напрямую опирается на предыдущий пункт о записи разговоров, поэтому, до выполнения этого пункта рекомендуется выполнить пункт о записи разговоров.

а) Создадим еще одну подпапку в нашей расшаренной папки. Туда будут складываться записанные файлы автоответчика

```
cd /records
mkdir voicemail
```

b) Так же создадим папку, в которую положим сообщение-приветствие автоответчика:

```
cd /var/lib/asterisk/moh
mkdir voicebox
```

d) Запишем приветствие автоответчика, что-то типа "в настоящее время все специалисты заняты. Пожалуйста, оставьте сообщение после сигнала". Записать можно, например в Nero Wave Editor и получить на выходе .mp3 файл

e) Для того, чтобы чел, который нам звонит слышал записанное сообщение о том, что все специалисты заняты, нем необходимо взять MP3 файл и перекодировать его в WAV кодеком G.711 U-Law. (нельзя просто так взять и... :)) Должен быть 8-bit моно. Для этого:

1) скачиваем и устанавливаем программу Ease Audio Converter. Скачать можно отсюда: http://audiotool.net/EaseAudioConverter/index.htm

2) В программе нажимаем кнопочку Setting, в появившемся окне выбираем формат wav и выставляем:

частота: 8000

каналы: моно

Выходной формат: 16 bit WAVE (PCM)

Выбираем mp3 файл, который хотим перекодировать и перекодируем.

f) Закачиваем полученный файл в Asterisk. Для этого:

1) скачиваем программу pscp http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

2) Кидаем программу на рабочий стол, но запускать её не надо

3) Запускаем cmd.exe в windows и пишем:

```
desktop\pscp.exe D:\test.txt remote_user@remote_host:/path_to_destination
```

path_to_destination это путь, по которому мы хотим кинуть файл. В этом конкретном случае /var/lib/asterisk/moh/voicebox

g) Теперь уже с помощью Asterisk нам необходимо перекодировать файл с помощью кодека U-Law. Для этого, в консоли пишем:

```
asterisk-r
```

далее

```
file convert youraudio.wav youraudio.ulaw
```

т.е для нашего конкретного случая будет выглядеть так:

```
file convert /var/lib/asterisk/moh/voicebox/название вашего файла.wav название вашего файла.ulaw
```

h) Теперь настроем Dial план в файле extensions.conf. Затираем все то, что мы писали ранее и пишем так:

```
[incoming]
exten => _X.,1,Goto(menu,s,1)

[outcoling]
exten => _X.,1,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN})
exten => _X.,2,MixMonitor(/records/callrecords/${fname}.wav,b)
exten => _XXXXXXXXXX,3,Dial(SIP/zadarma/${EXTEN})
exten => _XXXX,3,Dial(SIP/${EXTEN},,t&m,)
exten => 7777,3,Goto(menu,s,1,t&m)

[menu]
exten => s,1,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN})
exten => s,2,MixMonitor(/records/callrecords/${fname}.wav)
exten => s,3,Background(/var/lib/asterisk/moh/voicemail/voicemenu)
exten => 1,1,Dial(SIP/1001,30,m&t)
exten => 1,2,Goto(autoanswer,s,1) ;Если 1001 не ответил или сбросил вызов, перенаправляем на автоответчик
```

```
exten => 2,1,Dial(SIP/1002,30,m&t)

exten => 2,2,Goto(autoanswer,s,1) ;Если 1002 не ответил или сбросил вызов, перенаправляем на автоответчик

exten => s,4,Wait(5)

exten => s,5,Dial(SIP/1001&SIP/1002,30,t&m) ;если в течении 30 секунд ни 1001 ни 1002 не ответили или сбросили вызов, то вызывается контекст autoanswer (автоответчик)

exten => s,6,Goto(autoanswer,s,1)



[autoanswer]


exten => s,1,Background(/var/lib/asterisk/moh/voicebox/название нашего файла приветствия без расширения) ;проигрывается наше записанное приветствие. Мол все заняты

exten => s,2,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN}) ;здесь выполняется определение имени файла, в которое будет записан голос чувака, оставляющего сообщение на автоответчик

exten => s,3,Record(/records/voicemail/${fname}.wav,0,15,X) ;теперь записывается сам файл. При начале выполнения этой строчки, чувак на том конце слышит бииб.

exten => s,4,Hangup
```

Как видим, здесь у нас добавился по сравнение с предыдущим пунктом экстеншен autoanswer, который вызывается в случае, если SIP/1001&SIP/1002 заняты или не ответили.

Функция Record очень похожа на функцию MixMonitor, но она при начале записи воспроизводит сигнал beep, поэтому используется именно для автоответчика.

i) Теперь можно через windows подключаться к нашей расшаренной папки: \\records, переходить в папку voicemail и слушать сообщения, оставленные на автоответчик

Статьи про простой автоответчик:

http://i-wanna-think.ru/delaem-sobstvennyj-exotest-avtootvetchik-dlya-asterisk/

http://asterisk-support.ru/question/14890/avtootvetchik-privetstvie-i-zapis-soobshcheniia/

Более сложный автоответчик:

http://lsoft.daraba.ru/content/%D1%81%D0%BE%D0%B1%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9-%D0%B4%D0%B8%D0%B0%D0%BB%D0%BF%D0%BB%D0%B0%D0%BD-%D0%B0%D0%B2%D1%82%D0%BE%D0%BE%D1%82%D0%B2%D0%B5%D1%82%D1%87%D0%B8%D0%BA%D0%B0-%D0%B8-%D0%BF%D1%80%D0%BE%D1%81%D0%BB%D1%83%D1%88%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D0%B9-%D0%B2-asterisk-18%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D1%83%D1%8F

6.6 Улучшение работы автоответчика и записи звонков

Существует возможность воспроизводить записанные файлы автоответчика и записи разговоров через web интерфейс.

При этом можно осуществлять выборку записей по дате. Представьте, что за полгода накопилось миллион записей звонков. Если зайти в папку callrecords, то мы там просто утонем.

Выборка позволяет выбрать конкретный файл для прослушивания, да и это более удобно.

Для реализации этой функции, мы полностью отказываемся от использования samba. Теперь все наши записанные разговоры мы будем просматривать прямо из браузера.

Для этого:

1. Установка и настройка веб сервера Apache:

а) устанавливаем Apache:

```
yum -y install httpd mod_ssl
```

б) прописываем Apache в автозагрузку:

```
chkconfig httpd on
```

в) запускаем Apache

```
service httpd start
```

г) устанавливаем PHP с его дополнительными компонентами

```
yum -y install php php-common php-gd php-mysql php-xml php-mbstring
```

д) Перезапускаем Apache

```
service httpd restart
```

е) Теперь наберем в браузере http://ip_нашего_Linux и видим приветствие Apache. Это означает, что Apache установлен корректно. Напоминаю, что ip можно узнать, набрав ifconfig в консоли Linux

2. Apache работает с файлами по пути /var/www/html

Нам необходимо создать 2 новых папки в директории html и загрузить туда файл index.php, который и будет отвечать за выборку и воспроизведение наших записанных разговоров и файлов автоответчика.

Для этого:

а) Создаем 2 папки

```
cd /var/www/html
mkdir callrecords
cd /var/www/html
mkdir voicemail
```

б) Создаем файл в первой папке

```
nano /var/www/html/callrecords/index.php
```

Открывается редактор nano. Пишем туда код:

```
<?php


$file_list = glob("*.wav");


$q[]="";
$q[]="января";
$q[]="февраля";
$q[]="марта";
$q[]="апреля";
$q[]="мая";
$q[]="июня";
$q[]="июля";
$q[]="августа";
$q[]="сентября";
$q[]="октября";
$q[]="ноября";
$q[]="декабря";


$dlina=count($file_list);


echo "Количество файлов = ".$dlina."<br>";
?>
   <form name="test" method="post" action="index.php">
   Введите год месяц число, например (20110228)<input name="date" type="text" value="<?php echo
$_POST['date']; ?>"size="10">
   <input type="submit" value="Отправить">
   </form>
<?php


 if ($_POST['date']<>"") {
$day=substr($_POST['date'], 6, 2);
```

```
$month=substr($_POST['date'], 4, 2);
$year=substr($_POST['date'], 0, 4);
echo "Звонки записанные ".$day." ".$q[$month]." ".$year."<br>";
$datelist=$_POST['date'];
echo "<pre>";
for ($i=0;$i<=count($file_list);$i++)
{
$day=substr($file_list[$i], 6, 2);
$month=substr($file_list[$i], 4, 2);
$year=substr($file_list[$i], 0, 4);
$time=substr($file_list[$i], 8, 4);
$napravlenie=substr($file_list[$i], 13, 20);
$timeq=$time[0].$time[1].":".$time[2].$time[3];

$string=substr($file_list[$i], 0, strlen($datelist));
if ($string==$datelist) echo "<a href=".$file_list[$i].">".$day." ".$q[$month]." ".$year." в ".$timeq."
".$napravlenie."</a>\n";

}
echo "</pre>";
}
?>
```

Сохраняем файл (Ctr+X)

в) Создаем файл во второй папке

```
nano /var/www/html/voicemail/index.php
```

г) Открывается редактор nano. Пишем туда точно такой же код, как и в файле для первой папки

д) Сохраняем

3. Теперь по старой доброй традиции отредактируем файл extensions.conf. В этот раз нам нужно всего лишь поменять пути для трех записей

```
exten => _X.,2,MixMonitor(/records/callrecords/${fname}.wav,b)
exten => s,2,MixMonitor(/records/callrecords/${fname}.wav)
```

и

```
exten => s,3,Record(/records/voicemail/${fname}.wav,0,15,X)
```

на

```
exten => _X.,2,MixMonitor(/var/www/html/callrecords/${fname}.wav,b)
exten => s,2,MixMonitor(/var/www/html/callrecords/${fname}.wav)
```

и на

```
exten => s,3,Record(/var/www/html/voicemail/${fname}.wav,0,15,X)
```

Это мы поменяли пути. То есть раньше записанные разговоры и автоответчик сохранялись в папку records, и мы их просматривали через шару records, а теперь файлы сохраняются в директорию web сервера)

Полный экстеншен в моем случае выглядит следующим образом:

```
[incoming]
exten => _X.,1,Goto(menu,s,1)

[outcoling]
exten => _X.,1,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN})
exten => _X.,2,MixMonitor(/var/www/html/callrecords/${fname}.wav,b)
exten => _XXXXXXXXXXX,3,Dial(SIP/zadarma/${EXTEN})
exten => _XXXX,3,Dial(SIP/${EXTEN},,t&m,)
exten => 7777,3,Goto(menu,s,1,t&m)
```

```
exten => 9999,3,Goto(autoanswer,s,1,t&m)



[menu]
exten => s,1,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN})
exten => s,2,MixMonitor(/var/www/html/callrecords/${fname}.wav)
exten => s,3,Background(/var/lib/asterisk/moh/voicemail/voicemenu)
exten => 1,1,Dial(SIP/1001,30,m&t)
exten => 1,2,Goto(autoanswer,s,1)
exten => 2,1,Dial(SIP/1002,30,m&t)
exten => 2,2,Goto(autoanswer,s,1)
exten => s,4,Wait(5)
exten => s,5,Dial(SIP/1001&SIP/1002,30,t&m)
exten => s,6,Goto(autoanswer,s,1)



[autoanswer]
exten => s,1,Background(/var/lib/asterisk/moh/voicebox/busy)
exten => s,2,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN})
exten => s,3,Record(/var/www/html/voicemail/${fname}.wav,0,15,X)
exten => s,4,Hangup
```

4. Теперь совершим несколько звонков, оставим сообщение на автоответчик и прослушаем наши записи через web сервер:

а) Для записей разговоров напишем в браузере: http://ip_нашего_Linux/callrecords/

б) Для автоответчика напишем в браузере: http://ip_нашего_Linux/voicemail/

Когда это сделаем, у нас появится окошко. В это окошку можно написать дату. Пишем дату и у нас появляются все записи записанные указанной датой. Нажав на конкретную запись её можно прослушать.

P.S еще одна небольшая фишечка для удобства по этому делу есть в видео

Статьи:

http://admin.nnov.ru/zapis-razgovorov-v-asterisk-1-6.html

http://i-leon.ru/%D1%83%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0-%D0%B8-%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0-apache-php-mysql-%D0%BD%D0%B0-centos-pma-%D0%B8-ftp/

6.7 Установка системы просмотра статистики звонков

Случается так, что необходимо просмотреть кто, когда и кому звонил. По дефолту, Asterisk пишет логи просто в файл и если открыть его редактором nano то там будет технический код.

Чтобы разобраться в этом файле кто, куда и кому звонил - уйдут месяцы.

Именно поэтому существует специальные способы обрабатывать этот лог-файл и выводить нормальную информацию. В общем случае система работает так:

-устанавливается веб сервер

-устанавливается php модуль к этому серверу

-устанавливается MySQL

-астериск настраивается на работу с этим MySQL (начинает записывать логи звонков не в лог-файл, а в базу MySQL)

-на веб сервер загружается некий сайт, который и собирает статистику из базы MySQL и выводит подробную информацию в нормальном виде.

Итак, веб сервер с php у нас уже установлен (делали это в предыдущем пункте). Поэтому, начнем с установки MySQL:

1. Установка MySQL

а) Устанавливаем:

```
yum -y install mysql mysql-server
```

б) Добавляем в автозапуск:

```
chkconfig mysqld on
```

в) Запускаем

```
service mysqld start
```

г) Устанавливаем пароль:

```
mysqladmin -u root password 'new-password'
```

где new-password - пароль, который вы хотите установить. Для простоты предлагаю установить пароль root). Поэтому, будет выглядеть так:

```
mysqladmin -u root password 'root'
```

д) Проверяем MySQL. Создаем проверочный файл:

```
nano /var/www/html/mysqltest.php
```

и пишем туда вот этот код:

```
<?php
   $dblocation = "localhost";
   $dbname = "test";
   $dbuser = "root";
   $dbpasswd = "ваш пароль который вы поставили на сервер MySQL";

   $dbcnx = @mysql_connect($dblocation, $dbuser, $dbpasswd);
   if (!$dbcnx){
      echo "<p>К сожалению, не доступен сервер mySQL</p>";
      exit();
   }
   if (!@mysql_select_db($dbname,$dbcnx)){
      echo "<p>К сожалению, не доступна база данных</p>";
      exit();
   }
   $ver = mysql_query("SELECT VERSION()");
   if(!$ver){
      echo "<p>Ошибка в запросе</p>";
      exit();
   }
   echo mysql_result($ver, 0);
?>
```

e) Сохраняем и заходим через браузер по адресу: http://IP-адрес_сервера/mysqltest.php

Если появится номер версии MySQL - значит MySQL установлен корректно. Рекомендуется удалить файл mysqltest.php после проверки. Сделать это будет удобнее всего через mc (запустить Midnight Commander)

2. Теперь необходимо заставить Asterisk сохранять логи не просто в файл, как он это делает сейчас, а в MySQL базу. Для этого:

а) логинимся в сервер MySQL:

```
mysql -uroot -p
```

После чего нам предложат ввести пароль. Пароль у нас root

Появится сообщение:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.77 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

б) Создаем базу:

```
mysql> create database asterisk; (строку mysql> копировать не надо)
```

в) Создаем в базе “asterisk” таблицу “cdr”, вот с такой структурой

```
mysql> use asterisk;

mysql>  CREATE TABLE `cdr` (
 `id` int(11) 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=latin1;
```

ответом должно быть:

```
Query OK, 0 rows affected (0.04 sec)
```

г) Теперь даем доступ для пользователя “asterisk_user” с паролем “Some_Pass_Aster01? к базе “asterisk” только с локалхоста.

```
mysql> grant all on asterisk.* to 'asterisk_user'@'localhost' identified by 'Some_Pass_Aster01';

mysql> flush privileges;
```

д) Теперь указываем Asterisk писать CDR (так называется система логов) в базу MySQL. Для этого откроем файл для редактирования:

```
nano /etc/asterisk/cdr_mysql.conf
```

и вместо записи [global] которая там есть (вместо неё) пишем:

```
[global]
hostname=localhost
dbname=asterisk
table=cdr
password=Some_Pass_Aster01
user=asterisk_user
sock=/var/lib/mysql/mysql.sock
```

е) Сохраняем файл.

3. Мы установили MySQL и указали Asterisk писать свои логи CDR в базу. НО. Asterisk все еще не умеет работать с MySQL. Поэтому, сначала установим дополнительную "зависимость" (libmysqlclient library)

```
yum install mysql-devel
```

4. Поддержка MySQL находится в пакете Asterisk-addons. Asterisk до 1.8 не имел этого пакета на борту, его приходилось докачивать. Наш же астериск уже имеет его на борту (входит в состав его ядра). Теперь необходимо пересобрать Asterisk с поддержкой MySQL. Для этого:

а) переходим в директорию, в которую мы некогда, когда-то очень давно распаковывали Asterisk:

cd /usr/src/asterisk* (в моем случае это asterisk-11.5.1)

```
cd /usr/src/asterisk-11.5.1
```

б) Даем команду переконфигурации:

Для 32 бита

```
./configure && make menuselect && make && make install
```

Для 64 бита:

```
./configure --libdir=/usr/lib64 && make menuselect && make && make install
```

как выяснить какая битность? Набираем:

```
uname -a
```

Если ответ: 2.6.18-238.12.1.el5 #1 SMP Tue May 31 13:23:01 EDT 2011 i686 i686 i386 GNU/Linux то значит 32 бита

Если ответ: 2.6.18-238.19.1.el5 #1 SMP Fri Jul 15 07:31:24 EDT 2011 x86_64 x86_64 x86_64 GNU/Linux то значит 64 бита.)

В результате мы увидим графическое меню. В этом меню необходимо поставить звездочки в соответствии с этим:

— extended —

XXX chan_mobile

[ ] chan_ooh323

[ ] format_mp3

[*] res_config_mysql

— deprecated —

[*] app_mysql

[ ] app_saycountpl

[*] cdr_mysql

Нажимаем save & exit и ждем завершение сборки.

в) После сборки ОБЯЗАТЕЛЬНО перезагружаемся:

```
reboot
```

(я очень долго мучился с этим пунктом. Когда нажимал save & exit, сыпались ошибки и астер ну никак не хотел добавлять в себя поддержку MySQL. Оказалось, что необходимо было поставить libmysqlclient library (му её уже поставили))

г) После перезагрузки подключаемся к Asterisk:

```
asterisk -r
```

и пишем:

```
cdr mysql status
```

Если ответ:

```
Connected to asterisk on socket file /var/lib/mysql/mysql.sock using table cdr for 5 hours, 22 minutes, 7 seconds.
  Wrote 5 records since last restart.
```

То все ок - Астериск видит созданную нами базу. Если ответ

```
No such command ‘cdr mysql status’ (type ‘core show help cdr mysql’ for other possible commands)
```

то что-то не так. Например, Вы не перезагрузились или произошла какая-то неведомая...

Все перепроверяем и гуглим по ошибкам.

5. Теперь осталось прикрутить web интерфейс, который и будет выводить данные из базы MySQL.

а) Качаем сайт (веб-интерфейс) по ссылке: https://code.google.com/p/asterisk-cdr-viewer/ (там слева есть раздел downloads)

В результате получим скаченный архив.

б) Для того, чтобы спокойно загрузить файлы сайта на наш веб сервер в Linux, настроем сервер samba (samba мы уже устанавливали)

заходим в конфиг:

```
nano /etc/samba/smb.conf
```

и изменяем там последний раздел share definitions на:

```
# ——————————-Share Definitions ——————————
[share]
comment = share
path = /var
browseable = yes
writable = yes
guest ok = yes
read only = no
directory mask = 0777
force create mode = 0777
```

Не забываем перезагрузить сервер samba чтобы он перечитал конфиги:

```
/etc/init.d/smb restart
```

Теперь опять же в Linux установим права на папку html

```
chmod 777 /var/www/html
```

Все. Теперь заходим через Windows \\ip_нашего_сервера и переходим в папку html, после чего распаковываем скаченный архив с сайтом в папку html (у нас уже там есть папки callrecords

и voicemail - их не трогаем)

в) Переходим в папку include (работаем с распакованными файлами нашего сайта) и открываем файл config.inc.php (мышкой кликаем по нему :))

Там вместо

```
$db_type = 'mysql';
$db_host = 'localhost';
$db_port = '3306';
$db_user = 'cdrasterisk';
$db_pass = 'astcdr123';
$db_name = 'cdrasterisk';
$db_table_name = 'cdr';
```

прописываем

```
$db_type = 'mysql';
$db_host = 'localhost';
$db_port = '3306';
$db_user = 'asterisk_user';
$db_pass = 'Some_Pass_Aster01';
$db_name = 'asterisk';
$db_table_name = 'cdr';
$db_options = array();
```

и сохраняем файл

6. Проверяем работу нашего сервера CDR статистики. Для этого вводим в окно браузера: http://ip_нашего_сервера

Появится сайт. Нажимаем на кнопку Search и видим звонки, которые мы совершали (предварительно нужно позвонить кому-нибудь, ибо записываться они будут туда именно с того момента, как мы заставили Asterisk писать логи в MySQL базу)

Статьи на эту тему:

http://asterisk-system.ru/asterisk/asterisk-1-8-pishem-cdr-v-mysql-bazu.html - Asterisk 1.8 пишем CDR в MySQL базу

https://npmjs.org/package/mysql-libmysqlclient - установка зависимости libmysqlclient library

http://i-leon.ru/%D1%83%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0-%D0%B8-%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0-apache-php-mysql-%D0%BD%D0%B0-centos-pma-%D0%B8-ftp/ - установка MySQL (в середине файла)

https://code.google.com/p/asterisk-cdr-viewer/ - веб интерфейс asterisk-cdr-viewer

http://www.cdr-stats.org/ - альтернативный, более продвинутый веб-интерфейс для анализа и сбора статистики звонков

http://atsip.ru/resheniya-dlya-call-tsentrov/asternic-call-center-stats-rasshirennaya-statistika-dlya-call-tsentrov-na-asterisk - еще один модуль анализа и сбора статистики звонков

6.8 Усовершенствуем голосовую почту. Голосовая почта на каждый телефон с отправкой уведомления по e-mail.

Несколько разделов назад мы рассматривали возможности голосовой почты, при которой если никто не отвечал или сбрасывали трубку, мы могли оставить голосовое сообщение В ОБЩЕЕ ХРАНИЛИЩЕ, которое доступно любому сотруднику организации.

Теперь давайте сделаем так, чтобы голосовое сообщение можно было оставлять конкретному сотруднику. Например, если мы набрали внутренний номер 2 и попали на сотрудника Александра и он не ответил, мы могли бы оставить сообщение лично ему, а не для всех, как это было сделано несколько разделов назад. При этом, когда Александр вернется с обеда, он увидит на своем телефоне восклицательный знак, говорящий о том, что ему было оставлено голосовое сообщение. Кроме того, Александр еще получит на свой рабочий e-mail уведомление об оставленной голосовой почте с прикреплением этого голосового сообщения. То-есть он может прослушать голосовое сообщение нажав кнопку на телефоне, либо открыв прикрепленный файл к его e-mail уведомлению.

Для реализации задуманного, нам необходимо выполнить несколько пунктов:

1. Настройка почтового сервера Postfix
2. Настройка sip.conf
3. Настройка extensions.conf
4. Настройка voicemail.conf
5. Настройка софтфона X-Lite или аппаратного телефона

6. Проверка работоспособности созданной системы

7. Русификация голосового меню автоответчика

Приступим:

1. Настройка почтового сервера Postfix. На самом деле это не почтовый сервер, а только агент передачи почты (MTA — mail transfer agent). То-есть это некая Linux программа, которая будет отвечать за то, чтобы на почту сотрудника приходили уведомления о том, что ему оставили голосовое сообщение.

а) Регистрация почты на yandex. Нам необходимо зарегистрировать любую почту на yandex.ru. Потом мы прицепим Postfix к этому почтовому ящику и Postfix будет отправлять уведомления именно через этот почтовый ящик. Делайте почтовый ящик именно на yandex.ru. Ни на google (gmail), и ни в коем случае не на mail.ru. Я пробовал цеплять Postfix и к gmail и к mail.ru но у меня не получилось. Возможно, если покопаться и порыть, можно все же заставить Postfix работать с этими почтовыми серверами, но для того, чтобы избежать плясок с бубном, делаем почту на yandex.ru

Итак, мы зарегистрировали почту и получили адрес почтового ящика, например maycal2008@yandex.ru и пароль.

Теперь перейдем к непосредственно к настройке Postfix

б) Postfix по умолчанию уже установлен в CentOS, но нам необходимо установить некий пакет SASL. Расшифровывается это как Simple Authentication and Security Layer — метод для добавления поддержки аутентификации в протоколы соединения. Это необходимо для того, что Postfix смог залогиниться в yandex.ru (грубо)

```
yum install cyrus-sasl-plain
```

в) открыв файл /etc/postfix/main.cf выполнив команду

```
nano /etc/postfix/main.cf
```

в самый конец файла добавляем следующие строчки:

```
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/mailpasswd
smtp_sasl_security_options = noanonymous
```

```
smtp_sasl_type = cyrus
smtp_sasl_mechanism_filter = login
smtp_sender_dependent_authentification = yes
sender_dependent_relayhost_maps = hash:/etc/postfix/sender_relay
sender_canonical_maps = hash:/etc/postfix/canonical
smtp_generic_maps = hash:/etc/postfix/generic
```

г) Создаем файл /etc/postfix/mailpasswd выполнив команду

```
nano /etc/postfix/mailpasswd
```

и пишем туда следующие строчки

```
[smtp.yandex.ru] www@some.ru:password (это шаблон. В моем конкретном случае это выглядит так:)
[smtp.yandex.ru] maycal2008@yandex.ru:пароль
```

где [smtp.yandex.ru] - адрес smtp сервера yandex.ru; maycal2008@yandex.ru - адрес почты на Yandex, которую Вы зарегистрировали; пароль - пароль к почтовому ящику, который Вы зарегистрировали

д) Создаем файл /etc/postfix/sender_relay выполнив команду

```
nano  /etc/postfix/sender_relay
```

В этом файле мы указываем привязку доменов и конкретных отправителей к внешним службам. Для этого пишем туда следующие строчки:

```
@some.ru [smtp.yandex.ru]
www@some.ru [smtp.yandex.ru] (это шаблон. В моем конкретном случае это выглядит так:)

@yandex.ru [smtp.yandex.ru]
maycal2008@yandex.ru [smtp.yandex.ru]
```

е) Заходим в файл /etc/postfix/canonical выполнив команду

```
nano /etc/postfix/canonical
```

и в самом конце файла пишем

```
@some.ru www@some.ru
```

Это шаблон. В моем конкретном случае это выглядит так:

```
@yandex.ru maycal2008@yandex.ru
```

этим мы указываем агенту Postfix для домена через какой аккаунт отправлять

ж) Выполняем команду postmap для всех созданных и отредактированных файлов:

```
postmap /etc/postfix/canonical
postmap /etc/postfix/sender_relay
postmap /etc/postfix/mailpasswd
postmap /etc/postfix/generic
```

Главная задача команды postmap заключается в построении индексированных карт на основе обычных текстовых файлов. То-есть postfix работает не с конфигурационными файлами, а их индексированными картами. Поэтому, создав файл mailpasswd мы выполняем для него команду postmap и этот файл превращается в индексированную карту. Если будут внесены изменения в какой-либо файл, для него снова придется выполнить команду postmap.

Теперь пишем последовательно следующие команды (либо их можно скопировать в putty все разом)

```
chkconfig saslauthd on
service saslauthd restart
chkconfig postfix on
service postfix restart
```

Все! Мы сконфигурировали почтовый агент postfix и заставили его отсылать почту через наш созданный почтовый ящик maycal2008@yandex.ru. (тобишь CentOS теперь может отсылать почту кому угодно. Это типа как мы настроили Outlook в Windows) Теперь это нужно проверить.

Проверяем:

з) Устанавливаем поддержку команды mail

```
yum install mailx
```

и) Теперь пишем команду

mail (почтовый ящик, куда хотим написать письмо)

В реальной ситуации это выглядит так:

```
mail darkmaycal@gmail.com
```

Далее нам предлагают ввести тему письма

Subject: пишем сюда что-нибудь, но только обязательно на английской раскладке и нажимаем enter

Далее нам предлагают написать тело письма. Пишем что-нибудь, нажимаем enter

Далее, когда мы закончили писать тело письма нажимаем enter, ставим точку и снова нажимаем enter. Точка говорит о том, что мы закончили писать письмо:

enter

.

enter

Если все успешно, мы получим сообщение EOT

к) Теперь через windows открываем почту на которую мы послали сообщение, в моем случае это darkmaycal@gmail.com и смотрим, пришло ли туда сообщение. Если пришло - все ОК, мы все сделали правильно. Если нет, то посмотрим логи Postfix:

```
nano /var/log/maillog
```

там мы можем выяснить, в чем проблема и почему сообщение не было отправлено. Возможно, вы задали не правильный пароль к почте через которую Postfix отправляет сообщения или же допустили еще какую-либо ошибку в конфигурационных файлах.

Когда все ОК, лог файл выглядит так:

```
Nov  4 12:59:21 localhost postfix/pickup[2127]: A82BB102D29: uid=0 from=<root>

Nov  4 12:59:21 localhost postfix/cleanup[2136]: A82BB102D29: message-id=<20131104085921.A82BB102D29@localhost.localdomain>

Nov  4 12:59:21 localhost postfix/qmgr[2043]: A82BB102D29: from=<root@localhost.localdomain>, size=443, nrcpt=1 (queue active)

Nov  4 12:59:22 localhost postfix/smtp[2138]: connect to gmail-smtp-in.l.google.com[2a00:1450:4008:c01::1a]:25: Network is unreachable

Nov  4 12:59:24 localhost postfix/smtp[2138]: A82BB102D29: to=<darkmaycal@gmail.com>, relay=gmail-smtp-in.l.google.com[173.194.71.26]:25, delay=2.8, delays=0.21/0.01/0.67/2, dsn=2.0.0, status=sent (250 2.0.0 OK 1383555571 q8si1228583lag$

Nov  4 12:59:24 localhost postfix/qmgr[2043]: A82BB102D29: removed
```

Кстати, не пытайтесь отправить почту, которая зарегистрирована на mail.ru - не отправляет. Лог выводит ошибку:

```
relay=mxs.mail.ru[94.100.176.20]:25, delay=0.32, delays=0.28/0.01/0.02/0.01, dsn=5.0.0, status=bounced (host mxs.mail.ru[94.100.176.20] said: 550 Unroutable sender address (in reply to MAIL FROM command))
```

Если порыть, то можно заставить ходить почту на mail.ru, но в данном случае это не имеет смысла.

2. Итак, мы настроили postfix. Теперь перейдем непосредственно к теме автоответчика для каждого сотрудника. Зайдем в файле sip.conf

```
nano /etc/asterisk/sip.conf
```

Если вы делаете по моему примеру, то у Вас там описание для двух sip клиентов. [1001] и [1002]

Для первого добавляем:

```
mailbox=1001@default
language=en
```

для второго тоже самое, но не 1001, а для 1002

```
mailbox=1002@default
language=en
```

В моем конкретном случае это выглядит так:

```
[1001]
type=friend
host=dynamic
insecure=invite
username=1001
secret=1234
context=outcoling
disallow=all
allow=alaw
mailbox=1001@default ;это ГОЛОСОВОЙ ПОЧТОВЫЙ ЯЩИК куда будет записываться голос
language=en

[1002]
type=friend
host=dynamic
insecure=invite
username=1002
secret=45678
context=outcoling
disallow=all
allow=alaw
mailbox=1002@default ;это ГОЛОСОВОЙ ПОЧТОВЫЙ ЯЩИК куда будет записываться голос
language=en
```

Так же зарегистрируем еще одного sip клиента. Его назначение станет ясно позже:

```
[700]
type=friend
host=dynamic
insecure=invite
username=700
secret=5555
context=outcoling
disallow=all
allow=alaw
```

и сохраним файл

3. Перейдем к редактированию файла extensions.conf

```
nano /etc/asterisk/extensions.conf
```

и добавим там новые строки:

Для контекста [outcoling]:

```
exten => 700,1,VoiceMailMain()
```

для контекста [menu]

```
exten => внутренний номер абонента,приоритет,Voicemail(внутренний номер абонента@default)
```

В моем конкретном случае, весь файл extention.conf будет выглядеть следующий образом (добавлены те строки, у которых есть комментарий):

```
[incoming]
exten => _X.,1,Goto(menu,s,1)

[outcoling]
exten => _X.,1,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN})
exten => _X.,2,MixMonitor(/var/www/html/callrecords/${fname}.wav,b)
exten => _XXXXXXXXXX,3,Dial(SIP/zadarma/${EXTEN})
exten => _XXXX,3,Dial(SIP/${EXTEN},,t&m,)
exten => 7777,3,Goto(menu,s,1,t&m)
exten => 9999,3,Goto(autoanswer,s,1,t&m)
exten => 700,1,VoiceMailMain() ;здесь если позвонить на номер 700 мы сможем прослушать свою голосовую почту


[menu]
exten => s,1,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN})
exten => s,2,MixMonitor(/var/www/html/callrecords/${fname}.wav)
exten => s,3,Background(/var/lib/asterisk/moh/voicemail/voicemenu)
exten => 1,1,Dial(SIP/1001,30,m&t)
exten => 1,2,Voicemail(1001@default) ;здесь работает так: если SIP/1001 не ответил в течении 30 секунд или сбросил звонок, попадаем на его личный автоответчик 1001@default
exten => 2,1,Dial(SIP/1002,30,m&t)
exten => 2,2,Voicemail(1002@default) ;здесь работает так: если SIP/1002 не ответил в течении 30 секунд или сбросил звонок, попадаем на его личный автоответчик 1002@default
exten => s,4,Wait(5)
exten => s,5,Dial(SIP/1001&SIP/1002,30,t&m)
exten => s,6,Goto(autoanswer,s,1)

[autoanswer]
exten => s,1,Background(/var/lib/asterisk/moh/autoanswer/busy)
exten => s,2,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN})
exten => s,3,Record(/var/www/html/voicemail/${fname}.wav,0,15,X)
exten => s,4,Hangup
```

Общую работу всего контекста можно описать следующим образом:

1. Звоним в организацию с мобильника (или же можно позвонить, набрав 7777 - это имитация звонка с внешки, так у нас сделано в extensions.conf)

2. Попадаем на контекст menu

3. Выбираем внутреннего сотрудника, например 2

4. После того, как мы нажали 2, звонок идет абоненту 1002

5. Если абонент 1002 не ответил или сбросил звонок мы попадаем на его автоответчик и оставляем приветствие лично ему

Вариант работы контекста номер 2:

1. Звоним в организацию с мобильника (или же можно позвонить, набрав 7777 - это имитация звонка с внешки, так у нас сделано в extensions.conf)

2. Попадаем на контекст menu

3. Никакого сотрудника не выбираем, а просто ждем

4. Звонок идет сразу на 1001 и 1002

5. Если никто не отвечает или сбрасывает вызов, то попадаем на общий автоответчик, который мы делали несколько разделов назад.

4. Теперь настроим файл voicemail.conf

```
nano /etc/asterisk/voicemail.conf
```

и в самый конец файла пишем:

```
[default]
(внутренний номер абонента) => (пароль к голосовому почтовому ящику, любой, какой придумаете),(имя сотрудника),(почта на которую должно прийти уведомление)
```

Выше был приведен шаблон. А вот как это выглядит в моем конкретном случае:

```
[default]
1001 => 123, Mikhail, maycal593@gmail.com
1002 => 456, Alexander, darkmaycal@gmail.com
```

где 1001 - внутренний номер сотрудника

123 - пароль к голосовому почтовому ящику

Mikhail - произвольное придуманное мной имя

maycal593@gmail.com - почта, на которую приходит уведомление о том, что было оставлено голосовое сообщение

5. Перейдем к настройке софтфона X-lite. Думаю, что и для аппаратного телефона это будет справедливо. Только там мы все это делаем через web интерфейс самого телефона

а) Откроем софтфон и нажмем на кнопку "Softfone". Далее нажмем на Account Settings

б) В появившемся окне перейдем на вкладку Voicemail и в окно напротив "Number to dial for checking voicemail" поставим 700

Все. Теперь проверяем как работает наш автоответчик.

6. Проверка работоспособности созданной системы

а) Берем мобильный телефон и звоним на Asterisk.

б) Попадаем в интерактивное меню и нам говорят, что для того, чтобы связаться например с Александром, нажмите 2. Нажимаем 2.

в) Слышим, что пошел звонок и звонит телефон Александра. При этом не отвечаем и ждем 30 секунд

г) По истечении 30 секунд слышим на английском о том, что нас просят оставить сообщение

д) После звукового сигнала говорим что-нибудь и вешаем трубку

е) Видим, что у Александра на телефоне говорит кнопка о не прослушанных голосовых сообщениях

ж) Проверяем почту darkmaycal@gmail.com, туда должно было прийти уведомление с файлом. Уже в почте мы можем прослушать оставленное нам голосовое сообщение

Итак, мы можем прослушать сообщение, оставленное Александру через почту. Но есть альтернативный способ прослушать сообщение оставленное Александру

а) В X-lite софтфоне нажимаем на значок информирующий о том, что есть не прослушанные голосовые сообщения

б) X-lite автоматически звонит на номер 700 (помните, в sip.conf мы сделали этот 700-тый номер, а в extensions.conf для контекста outcoling сделали строку exten => 700,1,VoiceMailMain())

в) Женщина на английском нас просит ввести номер сотрудника, для которого мы хотим прослушать сообщение. Набираем 1002. Далее она просит ввести пароль. Пароль мы задавали в файле

voicemail.conf (1002 => 456, Alexander, darkmaycal@gmail.com).

г) Поэтому вводим пароль 456 и далее следуем инструкциям. Чтобы прослушать сообщение нажмите 1. Нажимаем 1 и прослушиваем сообщение.

То-есть как оно все работает еще раз (коротко о главном)

а) В sip.conf мы указали именно ГОЛОСОВЫЕ почтовые ящики для [1001] и [1002] (mailbox=1001@default и mailbox=1002@default для каждого из них соответственно). И кроме того, сделали

номер 700 для прослушивания почты

б) В файле extensions.conf мы заставили записывать голос человека на голосовые почтовые ящики 1001@default если не дозвонились до 1001 и 1002@default если не дозвонились до 1002

в) в файле voicemail.conf мы описали правила, по которым мы можем прослушать голосовое сообщение (какой пароль для какого внутреннего номера и на какую почту для какого внут.

номера отсылать уведомление) (1002 => 456, Alexander, darkmaycal@gmail.com)

г) Теперь, когда мы звоним на номер 700, мы вызываем внут. функцию астериска VoiceMailMain() и далее, когда вводим номер, например 1001, то у нас начинает работать файл

voicemail.conf, а именно строчка в нем: 1001 => 123, Mikhail, maycal593@gmail.com. Мы вводим пароль 123 и слушаем сообщение.

Прошу Вас не путать вот эти вещи: почтовый ящик 1001@default это ВНУТРЕННИЙ ЯЩИК Asterisk, на который ЗАПИСЫВАЕТСЯ ГОЛОС. А ящик maycal593@gmail.com - это Ваш ящик, на который приходит уведомление.

7. Русификация голосового меню автоответчика

Сейчас когда просят оставить сообщение, или когда мы звоним по номеру 700 чтобы прослушать оставленное сообщение нам говорят по-английски. Сотрудников фирмы это может повергнуть в шок.

Поэтому русифицируем.

а) Создадим папку ru по пути /var/lib/asterisk/sounds. Для этого напишем в консоли Linux:

```
mkdir /var/lib/asterisk/sounds/ru
```

б) Зайдем в эту папку:

```
cd /var/lib/asterisk/sounds/ru
```

в) Скачаем пакет русской локализации, написав:

```
wget http://downloads.asterisk.org/pub/telephony/sounds/asterisk-core-sounds-ru-alaw-current.tar.gz
```

г) Распакуем скаченный пакет:

```
tar xfz asterisk-core-sounds-ru-alaw-current.tar.gz
```

д) Зайдем в файл sip.conf

```
nano /etc/asterisk/sip.conf
```

и для ВСЕХ sip клиентов (даже для [zadarma]) поменяем значение language=en на значение language=ru (если нет параметра language - допишем его)

ж) Все. Теперь если позвоним на номер 700 мы услышим русские инструкции. И так же звонящий человек услышит русские инструкции о том, что он может оставить голосовое сообщение.

Статьи по этому делу:

http://dev.1c-bitrix.ru/community/webdev/user/8078/blog/nastroyka-postfix-dlya-otpravki-pochty/?commentId=29664 - Настройка Postfix для отправки почты через yandex

http://mypostfix.ru/34-utility-komandnoj-stroki.html - подробнее о postfix

http://www.opennet.ru/openforum/vsluhforumID1/94903.html - реализация системы "голосовая почта на каждый телефон"

http://xgu.ru/wiki/%D0%A0%D1%83%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_Asterisk - русификация автоответчика Asterisk

6.9 Перехват звонков. Pickup

В Asterisk существует способ перехватить звонок. Представим себе ситуацию, что мы находимся в одном отделе. И видим, что у нашего коллеги звони телефон, но его нет на месте.

Если мы на своем телефоне нажмем звездочку, то мы перехватим этот звонок (переведем его на свой телефон)

Рассмотрим простейший пример, при котором у нас существует только один отдел и мы перехватываем звонки в пределах этого отдела.

Для реализации задуманного, перейдем к редактированию файла sip.conf:

```
nano /etc/asterisk/sip.conf
```

и для [1001] и [1002] допишем следующие:

```
callgroup=2
pickupgroup=2
```

То есть теперь внутренние номера (сотрудники с номерами) 1001 и 1002 состоят в колл группе 2 и пикап группе 2

Теперь перейдем к редактированию файла features.conf

```
nano /etc/asterisk/features.conf
```

Найдем там строчку (это можно сделать, нажав сочетание клавиш ctrl+w и введя pickup, после чего нажав enter)

```
;pickupexten = *8
```

раскомментируем его (убираем в начале строки точку с запятой)

и значение с *8 поменяем на *

получится:

```
pickupexten = *
```

Сохраним файл и сделаем core reload.

Теперь позвоним например на 1001. 1001 звонит, но трубку не берем. Теперь на 1002 нажимаем * и кнопку "позвонить". И видим, что звонок перехвачен.

В более сложной реализации, когда есть несколько отделов, мы можем сделать так, чтобы 1001 мог перехватывать звонки, которые пришли на 1002, а 1002 не мог перехватывать звонки, которые пришли на 1001 (чтобы например только начальник отдела мог перехватывать звонки или чтобы два совершенно разных отдела в большой организации не перехватывали звонки друг друга). Это регулируется параметрами callgroup=2 и pickupgroup=2

Статьи по этому делу:

http://asteriskpbx.ru/pages/viewpage.action?pageId=1737104

http://www.hilik.org.ua/asterisk-pickup/

http://www.greenspider.ru/index.php/tekhnichka/asterisk-call-pickup

7. Усиливаем безопасность Asterisk. 17 шагов которые сохранят Ваши деньги.

В настоящее время атак, а главное, успешных взломов IP-АТС Asterisk немыслимое множество.

Как правило, взламывают российские Asterisk из других стран и начинаю совершать международные звонки. Сейчас это очень распространенно. После таких взломов, организациям приходят счета на десятки и даже сотни тысяч рублей. Бывали убытки и на миллионы. Причем жертвой может стать как и крупная организация (что не факт), так и маленькая. В основном это мелкие организации, где безопасности Asterisk оказывается минимальное внимание. Сисадмин думает - "это может случиться с кем угодно, но только не со мной. Взлом? Какой взлом, господи, о чем вы говорите?"

Но на самом деле пока вы читаете эти строки, тысячи людей по всему миру сидят и сканируют интернет в поисках очередной жертвы. Получив доступ к Asterisk, они могут сажать целые организации на ваш аккаунт и совершать международные звонки за ваши деньги.

Вот типичный алгоритм взлома:

1. Злоумышленник сканирует интернет на наличие систем с открытым портом 5060

2. Злоумышленник, найдя такую систему (Asterisk), начинает искать имеющиеся sip клиенты, к которым можно подключиться. Он снова и снова посылает запрос, а сервер снова и снова отвечает, что такого sip клиента нет, пока в конце концов не придет ответ о том, что такой sip клиент существует. В итоге, злоумышленник получает список вида:

[1000][1001][1002] - это sip клиенты

3. Затем, злоумышленник запускает "брутофорс" - программа подбора пароля к этим sip клиентам

4. Найдя пароль, злоумышленник запускает у себя на компьютере софтфон и регистрирует его по полученным данным - внешнему ip адресу (который он нашел сканируя открытые 5060 порты, логин (который такой же, как и номер найденного им sip-клиента) и пароль, который он подобрал в результате брутофорса.

5. Злоумышленнику более ничего не мешает совершать международные звонки.

Причин такому легкому взлому злоумышленниками несколько:

1. Asterisk по определенным причинам имеет выделенный IP адрес и смотрит в интернет (например этот сервер Asterisk так же является и сервером раздающим интернет, так, что "провод с интернетом и выделенным белым IP" вставлен прямо в этот сервер). При этом, на сервере с Asterisk не настроен фаерволл, и этот Asterisk повернут в сеть с распростертыми объятиями.

2. На ssh и sip назначены дефолтные порты

3. Используются простые пароли для sip клиентов

4. Не включена функция защиты от перебора существующих sip клиентов

5. Не реализованная функция защиты на уровне Dial плана

6. Не реализована функция доступа только из локальной сети

7. По ssh разрешен доступ root пользователя

8. В linux не отключены не нужные службы с дырами

9. Используется система с PBX-интерфейсом, например Elastix, которая имеет дополнительные уязвимости.

10. На уровне SIP провайдера разрешены международные звонки (когда они не нужны)

11. На уровне SIP провайдера не реализована функция лимитирования (звони сколько хочешь, но в конце месяца получишь счет)

Тот Asterisk, который мы сейчас сделали - он как котенок среди акул. У него нет никакой защиты и его очень легко могут взломать. Но мы исправим эту ситуацию.

Итак, наша ситуация - наш сервер Asterisk смотрит в интернет и имеет выделенный (белый) ip адрес. Если мы сейчас запустим софтфон X-lite где-нибудь в Африке, в качестве ip адреса укажем наш выделенный ip адрес Asterisk и введем логин и пароль - мы зарегистрируем софтфон точно так же, как если бы он находился внутри нашей локальной сети. Это не допустимо!

ИСПРАВЛЯЕМ:

7.1 Меняем SIP порт

SIP порт, это порт по которому софтфон (или обычные телефоны) подключаются к Asterisk.

Для этого:

```
nano /etc/asterisk/sip.conf
```

и сразу после надписи [general] пишем:

```
bindport=3348;
```

Сохраняем файл, делаем core reload и проверяем:

Запускаем софтфон X-lite, заходим в настройки и в разделе "Domain" к ip адресу через двоеточие дописываем :3348. У меня это выглядит так: 192.168.0.18:3348

После этого софтфон должен зарегистрироваться.

Все. Теперь тот, кто не знает порт, не сможет зарегистрировать телефон на сервере Asterisk, а соответственно - не сможет совершать звонки.

7.2 Запрещаем чужакам SIP подключение

Сделаем так, чтобы софтфон (или обычный телефон) мог подключаться к Asterisk ТОЛЬКО ЕСЛИ ОН НАХОДИТСЯ ВНУТРИ НАШЕЙ ЛОКАЛЬНОЙ СЕТИ.

Для этого:

```
nano /etc/asterisk/sip.conf
```

и для КАЖДОГО sip-клиента добавляем строчки:

```
deny=0.0.0.0/0.0.0.0
permit=192.168.0.1/24
allowguest=no
call-limit=2;
```

Покажу на примере [1001]. У меня это выглядит так:

```
[1001]
deny=0.0.0.0/0.0.0.0
permit=192.168.0.1/24
type=friend
host=dynamic
insecure=invite
username=1001
secret=1234
context=outcoling
disallow=all
allow=alaw
mailbox=1001@default
language=ru
allowguest=no
```

где

deny=0.0.0.0/0.0.0.0 - запрещаем подключение к sip клиенту с любых адресов

permit=192.168.0.1/24 - разрешаем подключение к sip-клиенту ТОЛЬКО ЕСЛИ запрос подключения инициализирован из нашей локальной сети (начиная 192.168.0.1 и заканчивая 192.168.0.255)

allowguest=no - запрещаем неавторизованным гостям подключаться к нашему sipy и требуем аутентификацию

call-limit=2 - означает, что одновременно могут звонить используя этот sip только 2 человека (два потока)

Далее сохраняем и делаем core reload

Примечание: строка 192.168.0.1/24 вовсе НЕ означает следующее: 192.168.0.1-192.168.0.24. Она означает следующее: 192.168.0.1-192.168.0.255

Понятно, что если у Вас сеть в другом диапазоне, Вы указываете другой диапазон, например 192.168.1.1/24

Все. Теперь, если кто-нибудь из Африки введет в софтфон наш внешний ip-шник - он не сможет зарегистрировать свой софтфон даже зная логин, пароль и порт к sip-клиенту, потому, что разрешено подключение ТОЛЬКО ЕСЛИ запрос происходит изнутри нашей локальной сети

7.3 Защищаем сервер от перебора по номерам

Находим в файле sip.conf с помощью команды Ctrl+W следующую строку:

```
;alwaysauthreject=yes;
```

и раскомментируем её. Она означает, что мы защищаем сервер от перебора по номера. Подробное объяснение работы этой функции представлено в видеокурсе в разделе обеспечения безопасности.

7.4 Устанавливаем более сильные пароли для sip-клиентов.

Недопустимо, чтобы пароли были вида 1234 или 6789 или просто числовые. Необходимо создать надежные, 16-символьные со служебными знаками пароли.

Хороший генератор паролей: http://www.randstuff.ru/password/

Рекомендуют делать пароли из 16 символов со служебными знаками и цифрами. Такой пароль практически невозможно подобрать брутофорсом!

В моем случае для sip-клиентов я устанавливаю разные пароли. Например, для [1001] в разделе secret я написал

```
secret=9%H4u(Kr&&7q5lg9
```

(не забываем поменять пароль в наших софтфонах или телефонах, иначе они не подключаться)

7.5 Запрещаем международные вызовы на уровне Dial плана

Можно сделать так, чтобы если кто-то пытается совершить международный звонок (например злоумышленник), то он получал бы во первых сообщение о том, что ему это запрещено, во вторых администратору отсылалось бы на почту сообщение о том, что кто-то пытается сделать межгород, ну а в третьих просто бы ему это не позволили бы сделать.

Для этого перейдем в extensions.conf

```
nano /etc/asterisk/extensions.conf
```

В конец контекста [outcoling] добавим следующие строки:

```
exten => _7810X.,1,System(echo «To» ${EXTEN} «Ext» ${CALLERID(num)} | mail -s «8-10 ALARM» test@gmail.com);
exten => _7810X.,n,Hangup()
```

Известно, что для того, чтобы позвонить в другую страну, необходимо набрать код 810. Данные строчки как раз это и регулируют. То-есть, если мы попытаемся позвонить в другую страну, набрав номер вида 7810XXXX... то сразу попадем на написанное выше правило, которое не позволяет нам этого сделать.

а) exten => _7810X.,1,System(echo «To» ${EXTEN} «Ext» ${CALLERID(num)} | mail -s «8-10 ALARM» maycal@podolsk-f1.ru) - данная строка отошлет e-mail администратора test@gmail.com сообщение о том, что была совершена попытка международного вызова

б) exten => _7810X.,n,Hangup() - повесит трубку.

Это означает, что злоумышленник, даже если и вдруг каким-то чудом получит доступ к sip-клиенту, он сможет позвонить куда угодно, но только не по межнароду)

Как видно, у меня написано _7810X. У SIP провайдера zadarma, чтобы совершить звонок, нужно набирать именно семерку вначале. Если у Вас другой провайдер, и Вы набираете девятку вначале, то, соответственно выглядеть будет так: _9810X.

Еще одно примечание: для того, что e-mail письмо могло быть отправлено, необходимо настроить почтовый клиент Postfix. О том, как это делается подробно описано в начале раздела "Усовершенствуем голосовую почту. Голосовая почта на каждый телефон с отправкой уведомления по e-mail."

Кроме всего прочего, если кто-то пытается набрать межнарод, то ему можно еще и проиграть сообщение о том, что данный вызов запрещен:

exten => _7810X.,n,playback(zapresheno) (предварительно нужно записать файл zapresheno)

Играясь с Asterisk, можно определенным отделам в организации разрешать звонить по международке, а другим - запрещать. Сделать это можно с помощью контекстов. Сейчас у нас для всех sip-клиентов один контекст outcoling. Если для разных sip-клиентов (людях в отделах фирмы) сделать разные контексты - в одном контексте будет запрет на международные вызовы, в других контекстах (отделах) - не будет.

Все! Теперь злоумышленник не сможет установить связь по международной линии.

7.6 Настройка встроенного фаерволла iptables.

Помните, когда мы только устанавливали Asterisk, мы отключили фаерволл iptables? Самое время его включить!

Выполняем команды:

```
service iptables start
```

После того, как мы запустили iptables, мы обнаруживаем, что у нас вообще ничего не работает - софтфоны не регистрируются, к веб серверу нет доступа, samba не работает.

А все потому, что по дефолту фаерволл iptables блокирует практически ВСЕ!

Перейдем к редактированию конфигурационного файла iptables:

```
nano /etc/sysconfig/iptables
```

По дефолту мы там увидим следующее:

-A INPUT –m state --state ESTABLISHED,RELATED –j ACCEPT (разрешаем пакеты для уже установленных соединений)

-A INPUT –p icmp –j ACCEPT (разрешаем пакеты протокола icmp)

-A INPUT –i lo –j ACCEPT (разрешаем трафик с интерфейса lo)

-A INPUT –m state --state NEW –m tcp –p tcp –dport 22 –j ACCEPT (разрешаем новое соединение ssh)

-A INPUT –j REJECT --reject-with icmp-host-prohibited (запрещаем все остальные входящие соединения)

-A FORWARD –j REJECT — reject-with icmp-host-prohibited (запрещаем все транзитные соединения)

Принцип работы этого фаерволла заключается в написании цепочек правил с параметрами (-s source, -p protocol, -i interface и т.д). Это означает, что трафик который попадает к нам в сервер начинает бежать по этим цепочкам правил и проверяться. Если подпадает под правило, которое разрешает данный конкретный трафик (на конкретный порт по конкретному протоколу) – данный трафик проходит дальше к нам в сервер. Если трафик пробежал все цепочки и не попал ни под одно правило – трафик блокируется.

Поэтому, вот то, что там по дефолту (все вот эти -A INPUT....) ни в коем случае не удалять оттуда другие записи! Только вот эти -A INPUT, -A FORWARD….

заменяем на:

```
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j DROP
-A INPUT -i lo -j ACCEPT
-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m multiport --dports 137,138,139,445 -j ACCEPT
-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -s 192.168.0.0/24 -p udp -m state --state NEW -m udp --dport 3348 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 10000:20000 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
```

Теперь объясню что значит каждая строка

`-A INPUT –m state --state ESTABLISHED,RELATED –j ACCEPT` (разрешаем пакеты для уже установленных соединений. Это означает, например, если веб браузер уже установил соединение с каким-нибудь сайтом, то он уже спокойно может с ним обмениваться пакетами)

`-A INPUT –p icmp –j DROP` (запрещаем пакеты протокола icmp. Грубо говоря - разрешаем или запрещаем ПИНГ нашего сервера. В данном стоит параметр DROP - поэтому пинговать наш Linux сервер с Asterisk себя не даст, что хорошо для безопасности. Если бы стоял параметр ACCEPT - пинг бы пошел)

`-A INPUT –i lo –j ACCEPT` (разрешаем трафик с интерфейса lo. Интерфейс lo это внутренний интерфейс Linux. В Windows он тоже есть. Он нас не интересует. Более подробнее о нем есть ссылка на статью в конце этой главы)

`-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m multiport --dports 137,138,139,445 -j ACCEPT` - разрешаем из НАШЕЙ ЛОКАЛЬНОЙ СЕТИ (92.168.0.0/24) проходить пакетам из вне во внутрь Linux на порты 137,138,139,445. Эти порты используются сервером samba (чтобы получить доступ к расшаренной папке по пути \\ip_сервера\папка

`-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT` - разрешаем из НАШЕЙ ЛОКАЛЬНОЙ СЕТИ (92.168.0.0/24) проходить пакетам из вне во внутрь Linux на порт 22. Этот порт используется SSH (мы подключаемся через этот порт используя putty и управляем сервером).

-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT - здесь тоже самое, мы разрешаем получить доступ к нашему веб-серверу на котором собирается

статистика из нашей локальной сети

-A INPUT -s 192.168.0.0/24 -p udp -m state --state NEW -m udp --dport 3348 -j ACCEPT - здесь мы опять же из нашей локальной сети разрешаем подключаться по порту 3348. Этот порт мы использовали для подключения sip-клиентов (помните, в sip.conf мы писали bindport=3348;). Только без фаерволла он сразу заработал, а включив фаерволл, его еще надо прописать и в фаерволле. Обратим внимание, что это не tcp, а UDP порт (сигнальный порт)

-A INPUT -p udp -m state --state NEW -m udp --dport 10000:20000 -j ACCEPT - разрешаем прохождение диапазону портов 10000:20000 (это UDP порты использующие для передачи голоса в телефонии. Без этого мы и нам смогут позвонить, но мы друг друга не услышим. Голос не пройдет)

-A INPUT –j REJECT --reject-with icmp-host-prohibited (запрещаем все остальные входящие соединения)

-A FORWARD –j REJECT --reject-with icmp-host-prohibited (запрещаем все остальные исходящие соединения)

Сохраняем файл и пишем команду:

```
service iptables restart
```

а затем

```
service iptables save
chkconfig iptables on
```

Этим мы включим фаерволл в автозагрузку и окончательно сохраним его параметры.

Теперь проверяем - все должно заработать.

Я теперь поясню. Вот например, в стандартной записи у нас была строчка:

```
-A INPUT –m state --state NEW –m tcp –p tcp –dport 22 –j ACCEPT
```

Эта строка разрешала подключаться к нашему серверу Linux через протокол ssh на 22 порт.

НО

Мы могли подключаться ОТКУДА УГОДНО. Если бы мы находясь в Африке ввели в putty наш внешний ip адрес, то мы смогли бы подключиться к нашему серверу.

А заменив эту строку на

```
-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
```

Мы разрешили подключаться к серверу через SSH ТОЛЬКО изнутри локальной сети! (добавили параметр -s 192.168.0.0/24)

Тоже самое и с остальными правилами. Мы разрешили подключаться к веб серверу ТОЛЬКО ИЗНУТРИ ЛОКАЛЬНОЙ СЕТИ

Мы разрешили подключиться к samba серверу ТОЛЬКО ИЗНУТРИ ЛОКАЛЬНОЙ СЕТИ

Мы разрешили подключаться к sip-клиентам ТОЛЬКО ИЗНУТРИ ЛОКАЛЬНОЙ СЕТИ (дополнительная подстраховка к такому же правилу как и в .sip.conf)

7.7 Изменяем порт SSH, запрещаем пользователю логиниться как root через ssh, добавляем нового пользователя

а) Добавляем нового пользователя

`useradd username` (добавляем нового пользователя с логином username)

`passwd username` (устанавливаем пароль для username)

далее нам предложат ввести новый пароль для пользователя username, поэтому, придумываем новый пароль не менее 6 знаков состоящим из букв, цифр и специальных символов, вводим его и нажимаем enter

b) Редактируем конфигурационный файл SSH

```
nano /etc/ssh/sshd_config
```

И в любое место этого файла пишем:

AllowUsers username (допускаем только что созданного пользователя username к доступу по SSH)

PermitRootLogin no (запрещаем пользователю root логиниться по ssh)

Так же находим строчку

```
#Port 22
```

Раскомментируем её и поменяем порт на любой другой (главное, чтобы новый порт не был занят каким-либо другим приложением)

У меня это стало так:

```
Port 1265
```

Далее сохраняем файл и выполняем команду для перезагрузки службы ssh:

```
service sshd restart
```

d) Разрешаем доступ по ssh через новый порт в фаерволле

Мы поменяли порт доступа к самой службе ssh, но еще не разрешили фаерволлу пропускать нас на этот порт. Для этого:

```
nano /etc/sysconfig/iptables
```

находим там строчку:

```
-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
```

и меняем в ней порт 22 на тот, что мы установили в файле sshd_config, в моем случае на 1265. Строка станет выглядеть так:

```
-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m tcp --dport 1265 -j ACCEPT
```

Сохраняем файл и перезагружаем службу фаерволла:

```
service iptables restart
```

d. Устанавливаем sudo права на пользователя username

`usermod -aG 'wheel' username` (добавляем пользователя username в группу пользователей wheel)

далее

```
nano /etc/sudoers
```

находим там строку:

```
## Allows people in group wheel to run all commands
#%wheel  ALL=(ALL)       ALL
```

и убираем комментарий перед %wheel. В результате выглядеть будет так:

```
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL
```

Этим самым действием мы разрешили пользователям из группы wheel (а там только один пользователь username, вы его внесли туда недавно командой usermod -aG 'wheel' username) получать права суперпользователя

Сохраняем файл

Все. Теперь закрываем все подключения ssh. Для того, чтобы снова подключиться через putty к ssh, нам нужно указать новый порт, который мы установили. В моем случае это 1265

Теперь работает так:

a) Мы подключились к нашему CentOS Linux, на котором стоит Asterisk через Putty по протоколу ssh, НО, если мы попытаемся залогиниться в обычном режиме, а именно укажем логин root и наш пароль от root пользователя, то получим ошибку доступа. А все потому, что мы строкой PermitRootLogin no запретили пользователю root подключаться по ssh. Так залогиниться теперь можно только если подключиться напрямую через консоль (ну к самому серверу подойти или открыть окно гипервизора)

b) Но мы создали нового пользователя username и теперь, для того, чтобы получить доступ по ssh когда он запрашивает логин вводим: username и пароль, который мы задали при добавлении нового юзера

c) По умолчанию, если мы залогинившись от пользователя username попытаемся отредактировать какой-то файл, или запустить какую-нибудь службу, то мы получим от ворот поворот - Acces Denied

d) Для того, чтобы от username можно было бы совершать какие-либо действия, вводим команду:

```
sudo -i
```

и пароль от пользователя username

Если все в порядке, то мы станем рутами (получим права суперпользователя) и сможем делать все, что мы захотим.

Итак, что же мы сделали в этом разделе (резюме):

a. Поменяли порт SSH

b. Добавили нового пользователя

c. Запретили root пользователю подключаться через ssh

d. Новому пользователю дали права суперпользователя

Что нам это даст?

a. Мы поменяли порт ssh. Теперь, если злоумышленник будет сканировать и искать открытые 22 порты, то нас он не найдет, поскольку мы поменяли порт

b. Если злоумышленнику все-таки удастся найти наш порт, обойти наш фаерволл iptables (он у нас пускает по ssh только изнутри локальной сети) то перед ним встанет вторая задача: необходимо подобрать пароль к root. Он начнет перебор паролей, но никогда не подберет, поскольку пользователю root просто на просто запрещен доступ по ssh

Очень мала вероятность того, что он догадается что есть такой пользователь username, а если даже и догадается, то пароль ему к нему подбирать очень и очень долго, ведь он состоит из 6 символов, притом в нем присутствуют служебные символы

P.S только не делайте имя пользователя реально username) Это слишком просто, придумайте свой. У меня например Maycal :)

Осталось поменять пароль на root командой:

```
passwd root
```

после ввода команды введем новый, сложный пароль символов 10-15

7.8 Выключаем Apache из автозагрузки и меняем его порт

Для этого:

```
chkconfig httpd off
```

Теперь при автозагрузке веб сервер запускаться не будет (а логи все равно будет писать, поскольку работает MySQL)

Если нам вдруг надо посмотреть логи (не каждый день же мы их смотрим) просто запустим этот сервис

```
service httpd start
```

посмотрев логи, остановим его:

```
service httpd stop
```

И напоминаю, что подключиться мы к нему можем только из локальной сети!

Кроме того, не лишним будет поменять стандартный порт 80 - в поисках него злоумышленники часто запускают свои скрипты. Не сделаем из нашей системы статистики звонков слабое место, поменяем порт!

Для этого:

```
nano /etc/httpd/conf/httpd.conf
```

а) Находим там строку

```
Listen 80
```

б) и меняем на любой другой, например 7623, после чего сохраняем файл.

в) Заходим в iptables:

```
nano /etc/sysconfig/iptables
```

находим там строку:

```
-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
```

и значение 80 заменяем на 7623

После чего сохраняем файл и перезагружаем iptaples и httpd службы:

```
service httpd restart
service iptables restart
```

Теперь, чтобы попасть на веб интерфейс и посмотреть отчеты, к ip адресу в браузере необходимо дописать порт. Выглядеть это будет так:

ip_адрес_сервера:7623

Все. Теперь веб сервер доступен только из нашей локальной сети, он имеет нестандартный порт и мы его запускаем только тогда, когда он нужен. Если у вас на веб сервере стоит более серьезная конфигурация, к которой нужен частый доступ, необходимо предпринять дополнительные шаги по обеспечению безопасности web сервера

7.9 Отключаем ненужные модули и протоколы Asterisk

Мы используем только SIP, но в asterisk есть еще и другие модули, которые нам не нужны, но через которые нас могут взломать. Поэтому отключаем их:

```
nano /etc/asterisk/modules.conf
```

и в этот файл прописываем следующее:

```
noload => chan_jingle.so
noload => chan_skinny.so
noload => chan_iax2.so
noload => chan_console.so
noload => chan_mgcp.so
noload => chan_gtalk.so
```

Сохраняем файл. Теперь все ненужные службы отключены и никто нас не взломает через дефолтный порт IAX, например.

7.10 Изменим порт управления Астериском (AMI)

Интерфейс управления сервером Asterisk (далее Manager API) позволяет клиентским программам соединяться с серверным приложением Asterisk, отправлять ему команды и/или считывать события, происходящие в АТС, используя TCP/IP протокол. Те, кто занимается интеграцией различных задач, могут найти много полезного для себя, например, отслеживая поведение телефонных абонентов и управляя ими на основании каких-либо правил.

Подключение к астериску через AMI выглядит следующем образом:

Для регистрации в manager API, подключившийся клиент должен пройти авторизацию, Вы должны отправить “Action” запрос с типом запроса: "Login" и указав имя пользователя и пароль в качестве параметров. Пример:

Action: login

Username: admin

Secret: god

Нам этом не нужно. В нашем конфигурационном файле доступа к AMI не заведен никакой пользователь admin (там вообще нет никаких пользователей по дефолту), мы никогда не использовали этот интерфейс, но от греха подальше поменяем дефолтный порт:

```
nano /etc/asterisk/manager.conf
```

Далее найдем там уже знакомую нам по предыдущим пунктам надпись

```
port = 5038
```

и поменяем цифры 5038 на любые другие не занятые. Например 8374.

Кроме того, после слова port допишем следующие строки:

deny=0.0.0.0/0.0.0.0 (запрещает доступ к AMI с любых ip адресов)

permit=192.168.0.1/24 (разрешает доступ к AMI с ip адресов 192.168.0.1-192.168.0.255)

Это позволит нам заблокировать доступ к AMI из любого места. Разрешено только из нашей локальной сети

Сохраним файл

7.11 Настраиваем систему fail2ban

В астериске, по дефолту, нет такой системы, которая бы проверяла количество неверных попыток ввести пароль и блокировала бы человека по ip адресу.

Как известно, если мы введем неправильный пароль по попытке подключиться по SSH, или неправильный пароль при попытке зарегистрировать SIP телефон - у нас в логах это отражается. Там будет написано "тот-то, с такого-то ip адреса не правильно ввел пароль". Причем у SSH свой лог, а у Asterisk свой.

Так вот, программа fail2ban собирает всю информацию с этих лог-файлов (с логов SSH, Asterisk, веб-сервера, с любого) и блокирует по ip адресу злоумышленника, который несколько раз неправильно ввел пароль.

Объясню еще раз

Если, например, мы подключились к SSH и 3 раза ввели неправильный пароль, то соответственно в логах SSH это отобразиться.

Программа fail2ban это увидит и через iptables заблокирует ip адрес того человека, который 3 раза ввел неправильно пароль. Почему 3 раза могут неправильно ввести пароль? А потому, что существует специальные программы, брутофорсы, которые перебирают и подбирают пароль. Если никак и ничем это не ограничить, рано или поздно (за день, неделю, месяц, год) такого непрерывного перебора злоумышленник сможет подобрать пароль например к SSH. А если ip злоумышленника будет блокироваться на некоторое время, то такой фокус уже не пройдет.

Возможно, будет более понятно при практической реализации:

а) Устанавливаем fail2ban

В стандартном репозитории программы fail2ban нет, поэтому, нам необходимо подключить дополнительный репозиторий.

Репозиторий, это место где хранятся программы для Linux. Если мы подключим какой-либо репозиторий и дадим команду

yum install fail2ban

то Linux будет искать эту программу на подключенных к ней репозиториях (база данных). Когда Linux увидит, что такая программа существует в какой-либо репозитории, Linux найдет ссылку на эту программу опять же в этом репозитории, и начнет её загрузку. Если же мы хотим установить какую-либо программу не из репозитория, а скачать её откуда-нибудь из интернета, мы используем команду wget.

Итак, подключаем репозиторий:

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

После этого устанавливаем саму программу:

```
yum install fail2ban
```

b) Запустим данную программу и добавим её в автозагрузку:

```
service fail2ban start
chkconfig fail2ban on
```

c) Переходим к редактированию главного конфигурационного файла этой программы

```
nano /etc/fail2ban/jail.conf
```

Рассмотрим структуру этого файла. Как видно, все они разделены на секции. Например [default] и [ssh-iptables]

Параметры из секции [default] применяются ко всем остальным секциям, если не будут переопределены.

Секция [ssh-iptables] отвечает за защиту SSH от повторяющихся неудачных попыток авторизации на SSH–сервере, проще говоря, «brute–force (брутофорс, в переводе - перебор паролей)

Теперь более подробно по параметрам, которые там есть:

ignoreip — IP–адреса, которые не должны быть заблокированы. Можно задать список IP-адресов разделённых пробелами, маску подсети, или имя DNS–сервера.

bantime — время бана в секундах, по истечении которого IP–адрес удаляется из списка заблокированных.

maxretry — количество неверных попыток ввода пароля, после которых применяется правило.

enabled — значение true указывает что данный jail активен, false выключает действие изолятора.

port — указывает на каком порту или портах запущен целевой сервис.

filter — имя .conf файла с шаблоном (который кладется по пути /etc/fail2ban/filter.d/) с регулярными выражениями, по которым идёт поиск «подозрительных совпадений» в журналах сервиса. Например, фильтру sshd соответствует файл /etc/fail2ban/filter.d/sshd.conf.

logpath — путь к файлу журнала, который программа Fail2ban будет обрабатывать с помощью заданного ранее фильтра. Вся история удачных и неудачных входов в систему, в том числе и по SSH, по умолчанию записывается в log–файл /var/log/secure.

findtime — определяет длительность интервала в секундах, за которое событие должно повториться определённое количество раз, после чего санкции вступят в силу. Если специально не определить этот параметр, то будет установлено значение по умолчанию равное 600 (10 минут). Проблема в том, что ботнеты, участвующие в «медленном брутфорсе», умеют обманывать стандартное значение. Иначе говоря, при maxretry равным 6, атакующий может проверить 5 паролей, затем выждать 10 минут, проверить ещё 5 паролей, повторять это снова и снова, и его IP забанен не будет. В целом, это не угроза, но всё же лучше банить таких ботов.

Итак, немного разобрались.

Найдем в этом файле строку `ignoreip = 127.0.0.1/8` и заменим её на `ignoreip = 1.2.3.4`. Этим самым мы указали совершенно левый ip адрес и поэтому, ignoreip работать не будет и будет блокироваться любой ip адрес

Найдем в этом файле строку `bantime  = 600` и заменим её значение на любое, какое заходим. Например 90.

90 это значение в секундах. То-есть 90 секунд будет блокирован ip адрес если 3 раза введет неправильно пароль.

Найдем в этом файле строку findtime = 600 и поменяем её значение на 3600.

Найдем в этом файле строку maxretry = 6 и установим ей значение 3.

Все вместе это работает так (это писать никуда не надо):

если в течении 1 часа:

findtime = 3600

произведено 6 неудачных попыток логина:

maxretry = 6

то банить IP на 24 часа:

bantime = 86400

Это мы установили общие переменные

Найдем и заменим секцию [ssh-iptables] на следующий код:

```
[ssh-iptables]
enabled  = true
filter   = sshd
action   = iptables[name=SSH, port=1265, protocol=tcp]
        sendmail-whois[name=SSH, dest=test@gmail.com, sender=Fail2Ban]
logpath  = /var/log/secure
maxretry = 3
```

Рассмотрим эту запись более подробно:

```
enabled  = true
```

(true означает, что данное правило включено)

```
filter   = sshd
```

(когда fail2ban будет просматривать лог файл ssh (/var/log/secure) в этом логе он будет искать совпадения, прописанные в файле-шаблоне по пути /etc/fail2ban/filter.d/sshd.conf)

```
action   = iptables[name=SSH, port=1265, protocol=tcp]
```

(управляет записью в iptables (ищет запись в iptables с портом 1265 и блокирует её)

```
 sendmail-whois[name=SSH, dest=test@gmail.com, sender=Fail2Ban]
```

(отправляет сообщение на test@gmail.com о том, что кто-то 3 раза ввел пароль не правильно. Здесь вместо test@gmail.com укажите свой почтовый ящик. Помните, что для работы этой функции у Вас должен быть настроен сервер postfix)

```
logpath  = /var/log/secure
```

(это путь к файлу логов, в который SSH пишет свои события maxretry = 3 ;количество неверных попыток ввода пароля, после которого эта система блокирует ip адрес, с которого ввели 3 раза этот пароль)

Внимание, если у Вас другой порт SSH - укажите свой порт, иначе работать не будет!

d) Сохраняем файл и перезапускаем fail2ban:

```
service fail2ban restart
```

Теперь, закрываем putty и открываем его снова. 3 раза неправильно вводим пароль и.. если мы сделали все правильно, то понимаем, что мы больше не можем подключиться к ssh через putty (наш ip заблокирован). При этом, на наш e-mail придет сообщение о том, что такой-то ip адрес был заблокирован.

Через 90 секунд нас разблокируют.

Для рабочей конфигурации рекомендую устанавливать значение bantime значительно больше, чем 90 секунд. Так же можете свой локальный ip адрес добавить в качестве параметра к ignoreip (вместо 1.2.3.4 прописать ip адрес компьютера, с которого подключаетесь через putty к Linux), тогда Вы

случайно не сможете заблокировать себя. Напоминаю, что ignoreip — IP–адреса, которые не должны быть заблокированы.

е) Сейчас у нас блокируется только SSH. Давайте сделаем так, чтобы если человек 3 раза пытается зарегистрировать софтфон с неправильным паролем - его ip адрес так же блокировался. Для этого:

1) Откроем файл

```
nano /etc/fail2ban/jail.conf
```

2) Над секцией [ssh-iptables] вставим следующий код:

```
[asterisk-iptables]
enabled  = true
filter   = asterisk
action   = iptables-allports[name=ASTERISK, port=3348, protocol=udp]
           sendmail-whois[name=ASTERISK, dest=test@gmail.com, sender=Fail2Ban]
logpath  = /var/log/asterisk/messages
maxretry = 3
bantime = 90
```

3) Откроем файл asterisk.conf

```
nano /etc/fail2ban/filter.d/asterisk.conf
```

Все оттуда удалим и вставим следующее:

```
# Fail2Ban configuration file
#
#
# $Revision: 251 $
#

[INCLUDES]

# Read common prefixes. If any customizations available -- read them from
# common.local
before = common.conf



[Definition]

#_daemon = asterisk

# Option:  failregex
# Notes.:  regex to match the password failures messages in the logfile. The
#          host must be matched by a group named "host". The tag "<HOST>" can
#          be used for standard IP/hostname matching and is only an alias for
#          (?:::f{4,6}:)?(?P<host>\S+)
# Values:  TEXT
#
# Asterisk 1.8 uses Host:Port format which is reflected here

failregex = NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - Wrong password
            NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - No matching peer found
            NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - No matching peer found
            NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - Username/auth name mismatch
            NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - Device does not match ACL
```

```
        NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - Peer is not supposed to register
        NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - ACL error (permit/deny)
        NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - Device does not match ACL
        NOTICE.* .*: Registration from '\".*\".*' failed for '<HOST>:.*' - No matching peer found
        NOTICE.* .*: Registration from '\".*\".*' failed for '<HOST>:.*' - Wrong password
        NOTICE.* <HOST> failed to authenticate as '.*'$
        NOTICE.* .*: No registration for peer '.*' \(from <HOST>\)
        NOTICE.* .*: Host <HOST> failed MD5 authentication for '.*' (.*)
        NOTICE.* .*: Failed to authenticate user .*@<HOST>.*
        NOTICE.* .*: <HOST> failed to authenticate as '.*'
        NOTICE.* .*: <HOST> tried  to authenticate with nonexistent user '.*'
        VERBOSE.*SIP/<HOST>-.*Received incoming SIP connection from unknown peer


# Option:  ignoreregex
# Notes.:  regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex =
```

4) Откроем файл logger.conf

```
nano /etc/asterisk/logger.conf
```

и к секции [general] добавим строчку: dateformat=%F %T

к секции [logfiles] добавим строчку security => security

У меня это выглядит так:

```
[general]
dateformat=%F %T

[logfiles]
security => security
```

5) Сделаем core reload для Asterisk и перезагрузим fail2ban:

```
service fail2ban restart
```

Все. Теперь, если мы 3 раза попытаемся зарегистрировать софтфон (или аппаратный телефон), то наш ip полностью заблокируется (не сможем подключить ни телефон, ни подключиться к SSH) в течении 90 секунд, а на свой e-mail получим сообщение о том, что ip адрес заблокирован (если указали e-mail, а не оставили дефолтный test@gmail.com)

7.12 Защита от DOS атак.

DOS атака (Denial-of-service) что в переводе с английского означает "отказ в обслуживании" применяется с целью того, чтобы ваш сервер захлебнулся

При DOS атаке на ваш сервер направляется огромное число мусорных пакетов, центральный процессор вашего сервера пытается их обработать, загружается под 100% и вы уже не сможете совершить какие-либо звонки, ибо сервер обрабатывает только этот мусор.

Так же существует DDOS атака, это та же самая DOS атака, только она мусорные пакеты вам посылают сразу с нескольких вычислительных ресурсов (серверов, компьютеров и т.п)

Защитимся от подобного рода угрозы:

1) Зайдем в iptables:

```
nano /etc/sysconfig/iptables
```

2) после строчки:

```
-A INPUT -i lo -j ACCEPT
```

добавим 2 новых записи:

```
-A INPUT -p tcp -m multiport --dports 1265,7623,3348,137,138,139,445 -m recent --set --name dos-attack
-A INPUT -p tcp -m multiport --dports 1265,7623,3348,137,138,139,445  -m recent --update --seconds 2 --hitcount 20 --name dos-attack -j DROP
```

3) Сохраним файл и перезагрузим iptables:

```
service iptables restart
```

Как же оно работает?

Фаерволл iptables просматривает все эти порты: 1265,7623,3348,137,138,139,445. Это наши SIP порт, SSH, Apach и samba

Далее работает функция "recent". Эта функция считает не превысил ли какой-либо пакет указанное количество (20) за указанное время (2 секунды).

Если, один и тот же пакет в течении 2 секунд уже 20 раз приходит с внешки и пытается проникнуть, он с помощью параметра -j DROP отбрасывается

Таким образом, если кто-то начнет на нас DOS атаку - пакеты не пройдут в наш сервер, не загрузят процессор, а просто будут отбрасываться.

7.13 Улучшение защиты от DOS атак

Существуют возможность по улучшению работы данной системы. С ней можно связать fail2ban. Система будет работать следующим образом:

1) В iptables вместо -j DROP пишется -j LOG --log-level INFO --log-prefix "SIP flood detected: "

Таким образом, пакеты от DOS атаки не отбрасываются, а сообщение о них записывается в лог файл iptables, который находится по пути:

/var/log/messages

Сообщение, сгенерированное iptables о том, что была DOS атака выглядит следующим образом:

```
Nov 23 19:13:05 localhost kernel: SIP flood detected: IN=eth0 OUT=
MAC=00:15:5d:00:11:03:fc:75:16:64:79:b6:08:00 SRC=192.168.0.17 DST=192.168.0.18 LEN=52 TOS=0x00
PREC=0x00 TTL=128 ID=21621 DF PROTO=TCP SPT=51263 DPT=7623 WINDOW=8192 RES=0x00 SYN URGP=0
```

2) fail2ban на основании шаблона, просматривает лог /var/log/messages и если он видит такое сообщение в этом логе, просто блокирует ip адрес который посылает эти сообщения и уведомляет нас по e-mail о том, что была совершена DOS атака

Реализуем!

1) Снова откроем конфигурационный файл iptables и внесем изменения:

nano /etc/sysconfig/iptables

Найдем строку

```
-A INPUT -p tcp -m multiport --dports 1265,7623,3348,137,138,139,445  -m recent --update --seconds 2 --
hitcount 20 --name dos-attack -j DROP
```

и заменим её на

```
-A INPUT -p tcp -m multiport --dports 1265,7623,3348,137,138,139,445 -m recent --update --seconds 2 --hitcount 20 --name dos-attack -j LOG --log-level INFO --log-prefix "SIP flood detected: "
```

Таким образом, при обнаружении DOS атаки пакет не отбрасывается, как это было раньше, а записывается в лог.

Так же, после этой строки вставим следующие 2 строки:

```
-A INPUT -p udp  --dport 3348 -m recent --set --name dos-attack

-A INPUT -p udp  --dport 3348 -m recent --update --seconds 2 --hitcount 20 --name dos-attack -j LOG --log-level INFO --log-prefix "SIP flood detected: "
```

Здесь мы контролируем наш UDP порт по которому подключается телефония, и если обнаруживается флуд - записываем об этом сообщение в файл.

Весь конфиг iptables у меня выглядит так:

```
# Generated by iptables-save v1.4.7 on Sun Nov 17 20:29:31 2013

*filter

:INPUT ACCEPT [0:0]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [6781:1108542]

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

-A INPUT -p icmp -j DROP

-A INPUT -i lo -j ACCEPT

-A INPUT -p tcp -m multiport --dports 1265,7623,3348,137,138,139,445 -m recent --set --name dos-attack

-A INPUT -p tcp -m multiport --dports 1265,7623,3348,137,138,139,445 -m recent --update --seconds 2 --hitcount 20 --name dos-attack -j LOG --log-level INFO --log-prefix "SIP flood detected: "

-A INPUT -p udp  --dport 3348 -m recent --set --name dos-attack

-A INPUT -p udp  --dport 3348 -m recent --update --seconds 2 --hitcount 20 --name dos-attack -j LOG --log-level INFO --log-prefix "SIP flood detected: "

-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m multiport --dports 137,138,139,445 -j ACCEPT

-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m tcp --dport 1265 -j ACCEPT

-A INPUT -s 192.168.0.0/24 -p tcp -m state --state NEW -m tcp --dport 7623 -j ACCEPT

-A INPUT -s 192.168.0.0/24 -p udp -m state --state NEW -m udp --dport 3348 -j ACCEPT
```

```
-A INPUT -p udp -m state --state NEW -m udp --dport 10000:20000 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Sun Nov 17 20:29:31 2013
```

2) Создадим шаблон, которым будет пользоваться fail2ban для поиска строки "SIP flood detected" в лог-файле iptables:

```
nano /etc/fail2ban/filter.d/dos-attack.conf
```

В этот файл вставляем следующее содержимое:

```
# Fail2Ban configuration file
#
#
# $Revision: 251 $
#

[INCLUDES]

# Read common prefixes. If any customizations available -- read them from
# common.local
before = common.conf



[Definition]

#_daemon = asterisk

# Option:  failregex
# Notes.:  regex to match the password failures messages in the logfile. The
```

```
#          host must be matched by a group named "host". The tag "<HOST>" can
#          be used for standard IP/hostname matching and is only an alias for
#          (?:::f{4,6}:)?(?P<host>\S+)
# Values:  TEXT
#
# Asterisk 1.8 uses Host:Port format which is reflected here


failregex = SIP flood detected: IN=.* OUT= MAC=.* SRC=<HOST> DST=.*
ignoreregex =
```

3) Сохраняем файл и идем в тюрьму:

```
nano /etc/fail2ban/jail.conf
```

(jail в переводе с английского означает тюрьма, если кто не знал :))

Там над секцией [asterisk-iptables] вставляем следующее:

```
[dos-attack]
enabled  = true
filter   = dos-attack
action   = iptables-allports[name=dos-attack, protocol=all]
           sendmail-whois[name=DOS-ATTACKE, dest=test@gmail.com, sender=Fail2Ban]
logpath  = /var/log/messages
maxretry = 3
bantime = 90
```

4) Сохраняем файл, перезагружаем fail2ban и проверяем. А как проверить? Необходимо скачать программу, которая начала бы флудить пакетами данных.

Одна из таких - LOIC. Использование этой программы я показываю в видео.

7.14 Защита от сканирования портов

Злоумышленник всегда сканирует порты, прежде чем начать атаку. Его задача - выяснить какие порты открыты. Для защиты от такого рода сканирования нам поможет дополнительный пакет для iptables - xtables-addons

Этот пакет будет работать примерно следующим образом:

а) Проверяет стук в нерабочие порты

б) Повторно проверяет стук в нерабочие порты

в) Начинает отбрасывать любые пакеты от ip адреса, который много стучится в нерабочие порты

Кроме этого, пакет xtables-addons позволяет защититься от многих видов сканирования. О видах сканирования Вы можете почитать здесь:

http://nmap.org/man/ru/man-port-scanning-techniques.html - различные приемы сканирования портов

Перед установкой пакета необходимо убедиться, что ваша система «видит» исходники ядра. Если система их не видит, то после выполнения пункта ж) make && make install нижеприведенной последовательности действий у вас возникнет ошибка и пакет xtables-addons не будет установлен. О том, как решить эту проблему показано в видеокурсе.

1) Установка пакета

а) yum install gcc gcc-c++ make automake unzip zip xz kernel-devel-`uname -r` iptables-devel

б) yum install perl-Text-CSV_XS

в) wget http://downloads.sourceforge.net/project/xtables-addons/Xtables-addons/1.37/xtables-addons-1.37.tar.xz

г) tar xvf xtables-addons-1.37.tar.xz

д) cd xtables-addons-1.37/

е) ./configure

ж) make && make install

2) Перезагружаем CentOS:

```
reboot
```

3) Настройка iptables

```
nano /etc/sysconfig/iptables
```

В открывшемся файле после строки

```
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

добавляем строку:

```
-A INPUT -m psd -j DROP
```

эта строка включает модуль psd, который входит в состав пакета xtables-addons, который мы только что установили

4) Перезагружаем iptables:

```
service iptables restart
```

5) Проверяем открытые порты с помощью Linux программы nmap с какого-нибудь другого удаленного места (об этом в видео)

7.15 Сертификация SSH

Существует возможность сделать так, чтобы подключиться по ssh (через putty) можно было бы только если на компьютере, с которого подключаются к серверу Linux установлен сертификат.

Ранее мы делали так, чтобы root пользователь не мог подключаться. Здесь же мы снова разрешаем подключаться root пользователю, но только через сертификаты, а не через пароль. Это удобно, если, например с удаленного хоста нужно часто подключаться к Asterisk - не нужно вводить пароли, да и никто у кого нет сертификата не сможет подключиться к вашему Linux с установленным на нем Asterisk.

Общая процедура следующая:

1) Генерируется ключи

2) Сгенерированный ключ заносится в файл authorized_keys

3) Сгенерированный ключ вытаскивается из Linux в Windows

4) Вытащенный сгенерированный ключ с помощью программы puttygen преобразовывается

5) Полученный преобразованный файл подключается к putty

6) Настраивается сама служба ssh в Linux

7) Проверка работоспособности

Приступим!

1) В консоли Linux напишем:

```
ssh-keygen
```

Далее предлагается указать место для сохранения ключа. По умолчанию это папка .ssh в вашей домашней директории. Для того, чтобы согласиться с настройками по умолчанию, просто нажмите «Enter».

Далее нас просят ввести идентификационную фразу. Эта фраза доступа к секретному ключу который сейчас генерируется. Для упрощения своей жизни на данном этапе мы не будем создавать такую фразу. Мы создадим её позже (об этом в видео). Чтобы не задавать идентификационную фразу ничего не вводим и просто нажимаем «Enter».

2) Все. Теперь переходим в директорию .ssh

Внимание! Именно .ssh а не просто ssh. Это две разные папки!

Переходим в эту папку:

```
cd .ssh (папка с точкой!)
```

В этой папке будут 2 файла:

id_rsa (приватный ключ) и id_rsa.pab (публичный ключ) - один с расширением а другое без оного

Находясь в этой папке пишем команду

```
cat id_rsa.pub >> authorized_keys
```

Эта команда создаст файл authorized_keys, возьмет содержимое файла id_rsa.pub и положит его в authorized_keys.

3) Теперь нам необходимо вытащить файл id_rsa из Linux в наш Windows из папки .ssh. Сделать это можно выполнив следующий алгоритм действий:

а) Стартануть сервер samba, расшарить папку .ssh, перезапустить этот сервер). Только не забудьте потом её закрыть.

б) Изменить права доступа с помощью команд: `chmod 7777 /root; chmod 7777 /root/.ssh;`

в) Зайти в папку .ssh (командой cd .ssh) и выполнить команду: `chmod 7777 id_rsa`

б) Зайти из windows через "выполнить" и введя \\ip_адрес_сервера в расшаренную папку и достать оттуда файл id_rsa

Кратко опишу команды:

Для доступа к файлу конфигурации samba:

```
nano /etc/samba/smb.conf
```

Находим там строку

```
path = "что-то там"
```

и заменяем её на

```
path = /root/.ssh
```

/etc/init.d/smb start - запуск сервера samba

/etc/init.d/smb restart - перезагрузка сервера samba

4) Скачиваем программу puttygen http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html запускаем её и в появившемся окне нажимаем кнопку load. Указываем путь к файлу id_rsa, который мы вытащили из Linux в Windows. Далее нажиманием на кнопку "Generate", указываем место сохранения файла и в результате получаем новый файл с расширением .ppk

5) Запускаем putty. Вводим наш ip адрес, порт для подключения к Linux, далее переходим на вкладку (справа) SSH->Auth. Там в поле "Privat key file for authentication" указываем путь к нашему .ppk файлу

Далее переходим на вкладку Connection->Data и там в поле auto-login user name пишем "root"

На вкладке sessions жмем "save". Наглядно это видно в видео.

6) Далее заходим в конфигурационный файл ssh:

```
nano /etc/ssh/sshd_config
```

и комментируем строку:

```
AllowUsers Maycal
```

После коммента будет выглядеть так:

```
#AllowUsers Maycal
```

Далее

```
PermitRootLogin without-password
```

(установили значение «without-password»)

Далее находим следующие строчки и приводим их в соответствие с этим:

```
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
RhostsRSAAuthentication no
HostbasedAuthentication no
PermitEmptyPasswords no
UseLogin no
PasswordAuthentication no
```

то есть где надо комментируем, где надо раскомментируем, где надо меняем с yes на no и наоборот

После этого сохраняем файл и выполняем перезагрузку ssh:

```
service sshd restart
```

7) Обязательно! Поменять обратно права доступа с помощью команд: chmod 700 /root; chmod 700 /root/.ssh; затем сайти в папку .ssh (командой cd .ssh) и выполнить команду: chmod 600 authorized_keys Без этого наш ключ не примится! Затем закрыть расшаренную папку (просто поменяв path = /root/.ssh на path = / в конфигурационном файле /etc/ssh/sshd_config. Файлы id_rsa и id_rsa.pab можно удалить из Linux через midnight commander

8) Все. Закрываем старое ssh соединение и снова пытаемся приконнектиться через putty. Если все ОК, то putty откроется и нам уже не нужно вводить никакой пароль.

Еще раз о том, как оно все работает более подробно:

С помощью программы ssh-keygen, генерируется два ключа (пара ключей). Это публичный ключ (public key) и приватный ключ (private key)

Секретный ключ вы копируете себе на компьютер (с которого подключаетесь через putty к серверу)

Публичный ключ остается на вашем Linux сервере с Asterisk и кладется в специальный файл, который знает только SSH сервер. По дефолту это файл authorized_keys. Назначается этот файл (путь к этому файлу и сам файл) в конфигурационном файле SSH сервера.

Далее, когда вы подключаетесь через Putty (по протоколу SSH) к вашему Linux серверу с Asterisk, то Linux высылает вам информацию, которая хранится в файле authorized_keys. То есть высылает вам открытый ключ. А ваша сторона, то есть ваш Putty с помощью вашего ключа (в видеокурсе это файл certificate.ppk) расшифровывает ту информацию, которую выслал Linux сервер из файла authorized_keys. Файл certificate.ppk это приватный ключ. Так вот, если информация высланная Linux сервером была расшифрована на вашей стороне файлом (ключем) certificate.ppk, то значит вы тот, за кого себя выдаете и вы получаете доступ в систему. Если же у вас нет ключа certificate.ppk либо это совершенно левый ключ, то информация посланная Linux сервером взятая из файла authorized_keys либо будет не расшифрована, либо расшифрована не правильно и Linux вас не пустит к себе через протокол SSH.

7.16 Отключение samba

а) Зайдем в его конфигурационный файл:

```
nano /etc/samba/smb.conf
```

и найдем там строку:

```
path = /какая-то расшаренная папка
```

закомментируем:

```
;path = /какая-то расшаренная папка
```

таким образом у нас больше нет расшаренных папок на сервере Asterisk

б) Выключим samba из автозагрузки:

```
chkconfig smb off
```

7.17 Дополнительная защита

а) Звоним провайдеру и просим, чтобы он запретил международные звонки (если они нам не нужны)

б) Звоним провайдеру и просим, чтобы он ограничил максимальную стоимость звонков в месяц (установил лимит) ну например в 10000 р. Тогда никто не сможет наговорить на миллионы рублей.

в) Если вам нужен частый УДАЛЕННЫЙ доступ через putty к Asterisk, то лучше будет устанавливать защищенное VPN соединение.

г) Не забываем сделать нормальные пароли к веб-интерфейсам аппаратных телефонов (если Вы их используете). В некоторых моделях можно поменять http порт. Смена http порта с 80 на любой другой значительно усложнит задачу злоумышленникам попасть на веб-интерфейс конкретного телефона.

7.18 Итоги обеспечения безопасности

Таким образом мы значительно повысили безопасность нашего Asterisk сервера. Никто не может приконнектить свой сипфон или же просто подключиться к нашему астериску как-либо (через Apache, putty, samba и т.п) если он находится вне нашей локальной сети и тем более не знает порта. Никто не знает наши порты - мы их все поменяли. Никто не сможет нас просканировать - мы поставили защиту от сканирования. Никто не сможет совершить DOS или DDOS атаку - его ip сразу заблокируют, а админ сразу узнает об этом. Никто не сможет подключиться к putty не имея сертификата, никто не сможет совершить международный звонок без ведома админа (ему приходит e-mail), у нас установлены сложнейшие пароли к sip-клиентам, никто не сможет подобрать пароли к sip-клиентам поскольку Asterisk больше не сообщает о том, правильный пароль или нет, есть ли такой sip клиент (номер) или нет. Никто не сможет назвонить на сотни тысяч рублей, поскольку мы поставили ограничение (лимит) на уровне провайдера.

Никто не сможет гарантировать Вам сто процентную защиту от взлома, однако, чтобы взломать нашу систему потребуется очень высокая квалификация взломщиков и большие деньги.

Статьи:

http://habrahabr.ru/post/188440/ - главная статья по обеспечению безопасности Asterisk

http://www.it-kub.ru/home/35-articles/54-bezopasnostasterisk.html - изменяем порт ssh, создаем нового пользователя ssh, запрещаем root'у подключаться через ssh, а так же устанавливаем дополнительный компонент управления iptables с возможностью ведения логов

http://www.pbxware.ru/blog/2011/06/7/sem-shagov-po-uluchsheniyu-bezopasnosti-asteriska/ - Семь шагов по улучшению безопасности Астериска

http://nagg.ru/2011/03/vzlom-i-zashhita-vashego-servera-asterisk/ - методика взлома asterisk с помощью linux программы sipvicious, статья 1

http://www.pvsm.ru/vzlom/8418 - методика взлома asterisk с помощью linux программы sipvicious, статья 2

http://vimeo.com/2524735 - программа для взлома SIP с GUI интерфейсом

http://linuxru.org/linux/56 - команды iptables для дополнительной защиты от сканирования портов

http://habrahabr.ru/company/myasterisk/blog/130325/ - настройка пропускной способности SIP (защита от большого количества трафика, т.н DDOS атаки)

http://ru.wikipedia.org/wiki/Loopback - о lo интерфейсе в Linux

http://www.randstuff.ru/password/ - хороший генератор паролей

http://www.fail2ban.org/wiki/index.php/Asterisk - настройка Asterisk + fail2ban 1

http://habrahabr.ru/post/194356/ настройка Asterisk + fail2ban 2

http://putty.org.ru/articles/fail2ban-ssh.html - настройка Asterisk + fail2ban 3

https://www.digitalocean.com/community/articles/how-to-protect-ssh-with-fail2ban-on-centos-6 - подключение репозитория для загрузки fail2ban

http://www.fail2ban.org/wiki/index.php/MANUAL_0_8 - как составить шаблон для fail2ban (в конце файла)

http://www.lissyara.su/articles/freebsd/security/fail2ban/ - команда fail2ban-regex (проверить работу шаблона по поиску совпадений в лог-файле)

http://nmap.org/man/ru/man-port-scanning-techniques.html - Различные приемы сканирования портов

http://nmap.org/man/ru/man-port-scanning-basics.html - Основы сканирования портов

http://www.howtoforge.com/xtables-addons-on-centos-6-and-iptables-geoip-filtering - установка xtables-addons

http://vds-admin.ru/ssh/ssh-autentifikatsiya-po-klyucham-ispolzovanie-programm-ssh-keygen-i-ssh-agent - сертификация ssh

http://habrahabr.ru/post/188440/ - сертификация ssh 2

8. Реализация дополнительных функций Asterisk

8.1 Конференц-связь Asterisk

Конференц-связь, это функция, при которой несколько человек могут разговаривать друг с другом одновременно

В asterisk это выглядит следующим образом:

1. Сотрудник организации, который хочет подключиться к конференции звонит на определенный номер
2. Далее он вводит пароль к конференции
3. Все. Сотрудник является участником конференции

Тоже самое делают и другие участники конференции, таким образом формируется комната конференц-связи.

Притом есть такая функция: конференция может начаться сразу, когда количество участников составит от двух человек, либо у сотрудников, подключившихся к конференции будет играть музыка, пока не придет лидер конференции.

Реализуем функцию конференц-связи!

Для реализации этой функции нам понадобятся только 2 файла:

confbridge.conf - основной конфигурационный файл конференц-связи

extensions.conf - основной конфигурационный файл самого asterisk

1. Настройка confbridge.conf

Зайдем в этот конфигурационный файл:

```
nano /etc/asterisk/confbridge.conf
```

и после раздела [general] напишем следующее:

```
[darkmaycalbridge]
type=bridge
max_members=20
mixing_interval=10
internal_sample_rate=auto
record_conference=yes

[generaluser]
type=user
music_on_hold_when_empty=yes
music_on_hold_class=default
announce_user_count_all=yes
announce_join_leave=yes
dsp_drop_silence=yes
denoise=yes
pin=456



[menu]
type=menu
*=playback_and_continue(conf-usermenu)
*1=toggle_mute
1=toggle_mute
*2=leave_conference
2=leave_conference
*4=decrease_listening_volume
4=decrease_listening_volume
*5=reset_listening_volume
5=reset_listening_volume
*6=increase_listening_volume
6=increase_listening_volume
```

```
*7=decrease_talking_volume
7=decrease_talking_volume
*8=reset_talking_volume
8=reset_talking_volume
*9=increase_talking_volume
9=increase_talking_volume
*0=no_op
0=no_op
```

Сохраним файл.

Как видно, у нас есть 3 секции:

[darkmaycalbridge], [generaluser] и [menu]. Расскажу про каждую из них и про параметры, которые содержат эти секции.

а) *[darkmaycalbridge]* это профиль конференции. Профиль конференции — это некие параметры, которые будут применяться к самой конференции. Например количество максимальных участников конференции, будет ли записываться конференция и прочее.

В нашем случае профиль конференции имеет следующие параметры:

type=bridge - указывает, что это именно профиль конференции

max_members=20 - указывает, что максимальное количество участников конференции - 20.

mixing_interval=10 - это технический параметр. Установка, в миллисекундах, определяющая внутреннюю нарезку звуковых потоков на семплы для их смешивания.

internal_sample_rate=auto - параметр устанавливает внутреннюю частоту дискретизации для смешивания звуковых потоков в конференции.

record_conference=yes - параметр включающий запись конференции.

б) *[generaluser]* это профиль пользователя. Здесь устанавливается пароль для входа в конференцию, а также указывается что ему делать можно, а что ему делать нельзя.

type=user - указывает, что это именно профиль пользователя

music_on_hold_when_empty=yes - будет ли проигрываться музыка, пока участники конференции ожидают её начала

music_on_hold_class=default - класс, из которого будет проигрываться музыка пока участники конференции ожидают её начала. В нашем случае будет проигрываться такая же музыка, которая проигрывается вместо гудка.

announce_user_count_all=yes - устанавливает, будет ли проигрываться анонс числа участников конференции всем ее участникам, когда новый пользователь входит в нее. Если аргументом является число, то анонс будет производиться если число участников конференции превышает указанное значение.

announce_join_leave=yes - если включено, пользователю будет предложено представиться перед тем, как он войдет в конференцию. После того, как имя будет записано, оно будет использоваться в сообщениях о входе и выходе пользователя в конференцию.

dsp_drop_silence=yes - указывает серверу Asterisk детектировать тишину и не позволяет звуковым данным, которые определяются как тишина, попадать в конференцию. Включение этой опции может резко повысить производительность и помогает в удалении фоновых шумов из конференции. Эта опция рекомендуется для крупных конференций, в связи с повышением производительности.

denoise=yes - применять или нет фильтр шумоподавления для аудиопотока пользователя конференции перед его смешиванием

pin=456 - устанавливает пароль для пользователя, чтобы он мог войти в конференцию

в) *[menu]* - меню. Меню позволяет участникам конференции включать или выключать свой микрофон, изменять громкость и прочее

**=playback_and_continue(conf-usermenu)(*если участник конференции нажмет *, то ему проговорят какая кнопка меню за что отвечает)

1=toggle_mute (выключить микрофон)

2=leave_conference (покинуть конференцию)

4=decrease_listening_volume (уменьшить громкость в наушниках)

5=reset_listening_volume (вернуть громкость в наушниках к значению по умолчанию)

6=increase_listening_volume (увеличить громкость в наушниках)

7=decrease_talking_volume (уменьшить чувствительность микрофона)

8=reset_talking_volume (вернуть чувствительность микрофона в исходное состояние)

9=increase_talking_volume (увеличить чувствительность микрофона)

0=no_op (не используется)

Рекомендую использовать именно такие параметры, как у меня, поскольку они позволяют добиться лучшего звучания вовремя конференц-связи.

2. Настройка extensions.conf

Теперь нам необходимо создать номер, на который мы будем звонить чтобы попасть в конференцию.

Перейдем в файл extensions.conf

```
nano /etc/asterisk/extensions.conf
```

и в конец контекста [outcoling] добавим строчку:

```
exten => 100,1,ConfBridge(1234,darkmaycalbridge,generaluser,menu)
```

Рассмотрим эту строку:

100 - номер, который мы будем набирать на телефоне, чтобы присоединиться к конференции

ConfBridge - вызов функции ConfBridge

1234 - номер конференции. Кроме как здесь, мы его нигде не указывали. То-есть если кто-то набирает номер 100, он попадает в комнату с номером 1234

darkmaycalbridge - указываем какой будет использоваться профиль КОНФЕРЕНЦИИ (он будет искать секцию [darkmaycalbridge] в файле confbridge.conf)

generaluser - указываем какой будет использоваться профиль ПОЛЬЗОВАТЕЛЯ (он будет искать секцию [generaluser] в файле confbridge.conf)

menu - указываем какое будет использоваться меню (он будет искать секцию [menu] в файле confbridge.conf)

Таким образом, все это работает так:

1. Кто-то с телефона набирает номер 100. По уже известным причинам мы попадаем в контекст outcolling (так прописано в sip.conf)

2. В контексте outcoling происходит выполнение экстеншена

```
"exten => 100,1,ConfBridge(1234,darkmaycalbridge,generaluser,menu)"
```

3. Таким образом вызывается функция ConfBridge, создается комната для конференции с номером 1234 и к этой комнате прилепляются все параметры, указанные в секциях darkmaycalbridge, generaluser и menu которые ранее мы задали в файле

confbridge.conf

Для того, чтобы стало еще понятнее:

Если мы сделали бы так:

`exten => 100,1,ConfBridge(1234,darkmaycalbridge,generaluser)` то у нас не было бы меню

а если бы мы сделали бы так:

```
exten => 100,1,ConfBridge(1234)
```

То в этом случае создалась бы комната 1234, но, все параметры которые описаны в darkmaycalbridge, generaluser и menu

(пароли, настройки качества звучания, максимальное количество участником и т.п) НЕ применились бы. Вместо этого сработали бы дефолтные секции, которые есть в confbridge.conf (там есть [default_user], [default_bridge]) но нам это не нужно.

Все! Проверяем работу:

1) Берем телефон и звоним с него на номер 100

2) Нам сообщают о том, чтобы мы ввели пароль. Пароль мы задали - 456

3) Далее нас просят сказать имя и нажать #. Говорим имя и нажимаем #

4) Нам говорят, что мы являемся первым участником конференции и начинает играть музыка

5) Можем нажать на * и прослушать инструкции о том, как управлять меню

6) Берем второй телефон и делаем тоже самое

7) После того, как второй телефон вошел в конференцию (присоединился второй участник), звучание музыки прекращается и конференция начинается.

8) Далее может подключиться второй, третий, четвертый и т.п участники.

3. Добавляем функцию, при которой конференция не начнется, пока не придет её лидер.

Выглядеть это будет так:

1. Все подключаются к конференции, но все равно слышат музыку

2. Музыка прекратится (конференция начнется) только тогда, когда зайдет администратор (ведущий конференции).

Для этого

а) nano /etc/asterisk/confbridge.conf

В этом файле к секции [generaluser] в любое место добавляем:

```
wait_marked=yes
```

Этот параметр не разрешает начаться конференции, пока не придет её лидер.

б) Так же в файл confbridge.conf добавляем нового пользователя:

```
[adminuser]
type=user
music_on_hold_when_empty=yes
music_on_hold_class=default
announce_user_count_all=yes
announce_join_leave=yes
dsp_drop_silence=yes
```

```
denoise=yes
marked=yes
admin=yes
pin=123
```

Как видим, здесь у него уже другой пароль (пароль администратора) и 2 новых параметра:

```
marked=yes
```

Этот параметр говорит asterisk'у о том, что этот пользователь - ведущий которого все и ждут.

```
admin=yes
```

Этот параметр говорит asterisk'у о том, что этот пользователь админ, который может управлять конференцией (закрывать и открывать её с помощью меню, кикать пользователей)

в) Добавляем новое меню для администратора (с более расширенными функциями возможности закрытия конференции и кика других участников)

```
[admin_menu]
type=menu
*=playback_and_continue(conf-adminmenu)
*1=toggle_mute
1=toggle_mute
*2=admin_toggle_conference_lock ; only applied to admin users
2=admin_toggle_conference_lock  ; only applied to admin users
*3=admin_kick_last      ; only applied to admin users
3=admin_kick_last       ; only applied to admin users
*4=decrease_listening_volume
4=decrease_listening_volume
*6=increase_listening_volume
6=increase_listening_volume
*7=decrease_talking_volume
7=decrease_talking_volume
```

```
*8=no_op
8=no_op
*9=increase_talking_volume
9=increase_talking_volume
```

г) сохраняем файл и заходим в extensions.conf:

```
nano /etc/asterisk/extensions.conf
```

здесь мы к строке

```
exten => 100,1,ConfBridge(1234,darkmaycalbridge,generaluser,menu)
```

добавляем строку

```
exten => 200,1,ConfBridge(1234,darkmaycalbridge,adminuser,admin_menu)
```

Отличий второй строки от первой только три:

номер 100 заменен на номер 200, параметр generaluser заменен на adminuser и параметр menu заменен на admin_menu. То-есть работает так:

если человек звонит на номер 100, то он подключается к конференции как обычный юзер (с параметром generaluser);

если человек звонит на номер 200, то он подключается к конференции как администратор (с параметром adminuser) и + у него более расширенное меню - admin_menu

г) Сохраняем файл и делаем core reload

д) Проверяем работу:

Нам потребуется 3 телефона

1) С первого телефона звоним на номер 100 (как простой участник конференции)

2) Вводим свой пароль (у нас 456)

3) Говорим своё имя и нажимаем #

4) Нам говорят о том, что конференция не начнется, пока не придет её ведущий

Берем второй телефон и делаем тоже самое

Итак, теперь два участника подключены к конференции, но все равно играет музыка, поскольку ведущий еще не подключился

1) Берем третий телефон и звоним на номер 200 (как ведущий конференции)

2) Вводим свой пароль (у нас 123) для adminuser

3) Говорим своё имя и нажимаем #

4) Конференция начинается (поскольку вы и есть тот админ, которого все ждут)

5) Можем нажать * и прослушать инструкции по админскому меню

P.S

1) К конференции могут подключаться не только пользователи внутри сети. К ней может подключиться любой человек (например с мобильника) знающий пароль. Для этого в файл extensions.conf в контекст [menu] между записями

```
exten => 2,2,Voicemail(1002@default)
```

и

```
exten => s,4,Wait(5)
```

можно написать:

```
exten => 3,1,ConfBridge(1234,darkmaycalbridge,generaluser,menu)
```

В таком случае, если кто-то позвонит с внешки и нажмет кнопку 3, он подключиться к конференции

2) У нас была строка:

```
exten => 100,1,ConfBridge(1234,darkmaycalbridge,generaluser,menu)
```

так вот, если у нас организация большая, и требуется проведение сразу нескольких конференциях в разных комнатах, можно добавить к этой строке вот такую строку:

```
exten => 300,1,ConfBridge(5678,darkmaycalbridge,generaluser,menu)
```

Здесь у нас поменялся номер на который мы звоним и номер комнаты. То-есть теперь, люди позвонившие на номер 100 будут общаться в комнате номер 1234, а люди позвонившие на номер 300 будут общаться в комнате 5678. Это будут две разные конференции, они не будут слышать друг друга и вообще никак не будут друг с другом взаимодействовать.

3) Есть еще множество параметров для профиля пользователей и профиля конференции. О них вы можете узнать здесь:

http://voip.rus.net/tiki-index.php?page=Asterisk+ConfBridge

4. Русифицируем

У нас получилась такая ситуация - половина инструкций для участников конференции говорится на русском, половина – на английском. На необходимо заменить все файлы в папке en на русские

Для этого напишем команду:

```
mc
```

Откроется midnight commander

Перейдем по пути /var/lib/asterisk/sounds

Для этого внизу midnight commander есть командная строка. Напишем туда:

```
cd /var/lib/asterisk/sounds
```

Далее удалим папку en. Для этого нажмем на неё правой кнопкой мыши (папка станет желтого цвета)

и нажмем кнопку f8

Появится красная табличка - подтвердим удаление

Теперь снова создадим папку en:

```
mkdir /var/lib/asterisk/sounds/en
```

(зачем её удалять и снова создавать? - чтобы удалить все её содержимое)

перейдем в созданную папку:

```
cd /var/lib/asterisk/sounds/en
```

скачаем языковой пакет и распакуем его:

```
wget http://downloads.asterisk.org/pub/telephony/sounds/asterisk-core-sounds-ru-alaw-current.tar.gz
tar xfz asterisk-core-sounds-ru-alaw-current.tar.gz
```

5. Записи конференций

Записи конференций хранятся по пути /var/spool/asterisk/confbridge. Доставать их оттуда можно например расшарив эту папку используя сервер samba

Все. На этом конфигурация конференц-связи Asterisk завершена!

Статьи по этому делу:

http://voip.rus.net/tiki-index.php?page=Asterisk+ConfBridge - confbrige

8.2 Парковка вызовов

Парковка вызовов — это вот какая штука:

1) Вам кто-то звонит и Вы поднимаете трубку

2) Однако, Вам нужно поговорить с этим человеком с другого места и с другого телефона. Например, Вас попросили подойти к серверной, а серверная находится в другом корпусе здания, НО - там есть телефон, подключенный к Asterisk

3) Тогда Вы паркуете вызов. Идете в серверную и там уже поднимаете трубку, продолжая разговор с позвонившем Вам человеком.

Технически это происходит следующим образом:

1. Вам позвонили и Вы сняли трубку

2. Далее Вы делаете слепой перевод (нажимаете # и вводите номер 900), т.е просто переводите звонок на номер 900

3. Asterisk сообщает парковочный номер

4. Вы подходите к другому телефону, звоните на номер который вам сказали и продолжаете разговор с собеседником. При этом, пока вы идете в другому телефону, собеседник слушает музыку

Реализуем эту функцию:

а) Отредактируем файл features.conf

```
nano /etc/asterisk/features.conf
```

В самом начале этого файла, в разделе [general] найдем строку:

```
parkext=700
```

и заменим её на

```
parkext=900
```

Далее найдем строку

```
parkpos=701-720
```

и заменим её на

```
parkpos=901-920
```

Далее найдем строку

```
parkingtime=45
```

и заменим её на

```
parkingtime=18000
```

Примечание: если какие-то из строк закомментированы - раскомментируйте

Разберемся что к чему:

parkext=900 - номер, на который мы будем парковать (переводить) собеседника

parkpos=901-920 - парковочные слоты, номера которых вам будет сообщать Asterisk

parkingtime=18000 - время в секундах, через которое наш запаркованный собеседник автоматически вернется на тот телефон, на который он и звонил. 45 секунд слишком мало, чтобы подняться на лифте и дойти до серверной, поэтому ставим значительно больше

б) Отредактируем extensions.conf

```
nano /etc/asterisk/extensions.conf
```

В самый конец контекста [outcoling] напишем:

```
exten => 900,1,Park()
exten => _9XX,1,ParkedCall(${EXTEN})
exten => _XXX,hint,park:${EXTEN}@parkedcalls
```

Эти строчки нужны для того, чтобы парковка вызова работала.

в) Далее, команда core reload НЕ поможет. Для принятия параметров features.conf необходимо выполнить команды:

зайти в asterisk CLI:

```
asterisk -r
```

далее

```
core restart now
```

далее

```
asterisk -r
```

далее

```
features reload
```

г) Проверяем работу парковки вызовов

Для этого лучше всего позвонить с внешки.

1. Берем мобильный телефон и звоним какому-нибудь внутреннему абонентку

2. Отвечаем на звонок

3. Нажимаем # (при этом слышим как нам произносят слово "перевод" и добавляем 900)

4. Asterisk голосом женщины сообщает нам парковочный номер. Он будет 901

5. В мобильнике слышим музыку (поставлен на удержание), а внутренний телефон (на который звонили) отключается

6. Идем в другое место, к другому телефону. С этого телефона звоним на парковочный номер, который нам сообщили. В нашем случае это 901

7. Продолжаем разговор с абонентом.

P.S Если не работает, то попробуйте перезагрузить весь CentOS. Возможно команда core restart now не помогла

Кроме того, для того, чтобы работала парковка, необходимо чтобы работала функция перенаправления звонков.

О функции перенаправления звонков можно почитать ближе к началу этой инструкции в разделе "Перенаправление звонков"

P.S 2

Строки:

```
exten => 900,1,Park()
exten => _9XX,1,ParkedCall(${EXTEN})
exten => _XXX,hint,park:${EXTEN}@parkedcalls
```

можно было бы заменить одной простой функцией:

```
include => parkedcalls
```

но эта функция у меня не сработала

Статьи по этому делу:

http://www.voip-info.org/wiki/view/Asterisk+cmd+ParkedCall

http://ankisa-blog.blogspot.ru/2012/11/asterisk-18-fixed.html

http://voip.rus.net/tiki-index.php?page=Asterisk+call+parking

8.3 Переадресация звонков (FollowMe)

Переадресация звонков работает следующим образом:

1. Вам с внешки кто-то звонит на внутренний номер
2. Если Вы не отвечаете в течении установленного времени, то звонок переводится к Вам на мобильный
3. Если Вы не отвечаете и на мобильный тоже, включается автоответчик

Для реализации это функции:

1. Откроем главный файл отвечающий за эту функцию:

```
nano /etc/asterisk/followme.conf
```

В самый конец файла напишем:

```
[1001]
context => outcoling
number = 00000000000,40
```

Где

[1001] - внутренний номер, с которого будет произведена переадресация.

context => outcoling - контекст (секция в файле extensions.conf) через который будет производиться звонок на мобильный.

00000000000 - номер мобильного телефона, на который будет совершен звонок если абонент 1001 не ответит.

40 - время в секундах, в течении которого Asterisk будет звонить на мобильный телефон. Не делайте это время меньше чем у меня - не успеете взять трубку на мобильном.

2. Теперь зайдем в extensions.conf

```
nano /etc/asterisk/extensions.conf
```

В контексте [menu] найдем все строчки отвечающие за работу контекста, в случае если мы нажали кнопку 1 и заменим их на другие.

То-есть находим там строчки:

```
exten => 1,1,Dial(SIP/1001,10,m&t)
exten => 1,2,Voicemail(1001@default)
```

и меняем их на

```
exten => 1,1,Answer()
exten => 1,2,Dial(SIP/1001,10,m&t)
exten => 1,3,FollowMe(1001)
exten => 1,4,Voicemail(1001@default)
```

Как видно, если кто-то прослушав меню решил связаться именно с тысяча первым абонентом, нажав кнопку 1, то сначала поднимается трубка:

```
exten => 1,1,Answer()
```

затем звонок идет на абонента 1001 в течении 10 секунд. Если там трубку не взяли в течении 10 секунд, звонок идет на функцию FollowMe

```
exten => 1,3,FollowMe(1001)
```

В файле FollowMe он находит указанную ему запись (1001) и там уже видит на какой телефонный номер перенаправлять звонок. В нашем случае это 00000000000 - мобильный

и если в течении 40 минут никто не ответил на мобильный -

```
exten => 1,4,Voicemail(1001@default)
```

включается автоответчик

Весь контекст [menu] выглядит следующим образом:

```
[menu]
exten => s,1,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN})
exten => s,2,MixMonitor(/var/www/html/callrecords/${fname}.wav)
exten => s,3,Background(/var/lib/asterisk/moh/voicemenu/voicemenu)
exten => 1,1,Answer()
exten => 1,2,Dial(SIP/1001,10,m&t)
exten => 1,3,FollowMe(1001)
exten => 1,4,Voicemail(1001@default)
exten => 2,1,Dial(SIP/1002,30,m&t)
exten => 2,2,Voicemail(1002@default)
exten => 3,1,ConfBridge(1234,darkmaycalbridge,adminuser,menu)
exten => s,4,Wait(5)
exten => s,5,Dial(SIP/1001&SIP/1002,30,t&m)
exten => s,6,Goto(autoanswer,s,1)
```

Это работает только если кто-то звонит с внешки. Если мы хотим, чтобы при звонке на 1001 работала переадресация на мобильный даже если кто-то звонит со внутреннего номера, то в контекст outcoling, в самое начало пишем следующее:

```
[outcoling]
exten => _1001,1,Answer()
exten => _1001,n,Dial(SIP/1001,10,t&m)
exten => _1001,n,FollowMe(1001)
```

Теперь если конкретно кто-то позвонит на номер 1001 (с другого внутреннего телефона), сработают 3 этих правила в приоритете перед всеми остальными в этом контексте.

Не забываем сделать core reload после внесенных изменений

8.4 Очередь звонков. Создаем Call-центр.

Пример очереди звонков вы можем наблюдать, если позвоним, например в какой-нибудь Call центр. Там нам говорят, пожалуйста, оставайтесь на линии, скоро вам ответят и вы слушаете музыку в трубке телефона, пока какой-нибудь оператор не соединится с вами.

У меня радостная новость - такая же функция есть и в Asterisk. По моему мнению, она подходит крупным организациям в которых есть Call центры. Например крупный провайдер куда поступает сотни звонков в течении дня и на их обработке сидит несколько человек, либо это крупная торговая компания где идет сотни звонков от клиентов.

Мы эту систему реализуем следующим образом:

1. Кто-то звонит с внешки

2. Этот "кто-то" слышит наше уже давно записанное голосовое меню: "если вы хотите связаться со специалистом Михаилом Юрьевичем, нажмите 1. Если вы хотите связаться со специалистом Александром Владимировичем нажмите 2 или оставайтесь на линии"

3. Так вот, мы выбираем "оставаться на линии". Сейчас, если мы не выбрали ни Михаила Юрьевича ни Александра Владимировича то звонят сразу 2 телефона - SIP1001 и SIP1002. А мы сделаем так, что звонок вместо этого будет поступать в очередь.

Процедура создания такой системы выглядит следующим образом:

1. Настройка файла agents.conf

2. Настройка файла queues.conf

3. Настройка файла extensions.conf

4. Проверка работы системы очереди звонков

Реализуем!

1. Настройка файла agents.conf

agents.conf это файл, в котором настраиваются агенты. Агенты - это люди, операторы которые и будут отвечать на звонки.

Откроем этот файл:

```
nano /etc/asterisk/agents.conf
```

и в самый конец файла напишем:

```
agent => 1001,123,Mikhail
agent => 1002,456,Alexander
autologoff=15
```

Сохраняем и закрываем файл.

2. Настройка файла queues.conf

Открываем файл queues.conf:

```
nano /etc/asterisk/queues.conf
```

и в самый конец файла пишем:

```
[operators]
music = default
strategy = ringall
context = queue-out
autofill = yes
announce-position = limit
wrapuptime=50
announce-frequency = 30
announce-holdtime = yes
joinempty = yes
member => Agent/1001
member => Agent/1002
```

Сохраняем и закрываем файл

3. Настройка файла extensions.conf

В контекст [outcoling] в самый конец пишем:

```
exten => 800,1,AgentLogin()
```

Далее в контексте [menu] находим строчку:

```
exten => s,5,Dial(SIP/1001&SIP/1002,30,t&m)
```

и удаляем либо комментируем её. Комментарий означает, что эта строчка теперь просто надпись и выполняться не будет. Я её закомментирую (точка с запятой в начале строчки):

```
;exten => s,5,Dial(SIP/1001&SIP/1002,30,t&m)
```

сюда же (после закомментированной строчки) добавляем новую строку:

```
exten => s,5,Queue(operators,,,,100)
```

Сохраняем файл и делаем core reload

4. Проверяем работу системы.

Сделали core reload?

Ок

1) Берем первый телефон - 1001.

2) Звоним с этого телефона на номер 800

3) Голос девушки нам сообщает, чтобы мы ввели номер.

4) Вводим номер первого агента (оператора). В нашем случае он такой же, как и сам внутренний номер - 1001

5) Далее нас просят ввести пароль. Для 1001 мы установили пароль 123 - вводим его

6) Нам говорят, что оператор зарегистрирован и начинает играть музыка.

Все! Оператор зарегистрирован и ждет входящих звонков.

7) Берем второй телефон - 1002 и делаем тоже самое. Пароль у нас здесь уже будет 456

Итак, у нас зарегистрировано 2 оператора.

8) Берем мобильный телефон и звоним на наш астериск. Прослушиваем голосовое приветствие, но никого не выбираем, а "остаемся на линии".

9) Теперь подойдем сначала к 1001 телефону, а потом к 1002. Вот в том из них, в котором тишина (не играет музыка) - уже установлено соединение с мобильником.

10) Чтобы оператор мог завершить разговор с клиентом (скинуть его) - нельзя нажимать на красную трубочку. Ибо отключится сам оператор от Asterisk. Достаточно нажать * и клиент отключиться, а у оператора продолжить играть музыка и он будет ждать следующего звонка.

Если, например 1001 и 1002 уже с кем-то разговаривают, то наш звонок с мобильного встанет в очередь и мы услышим голос девушки, говорящий нам о том, что мы первые в очереди, пожалуйста, подождите.

Если вообще никто так и не ответит клиенту висящему на очереди - клиент услышит автоответчик.

Как видно, здесь нет такого что телефон операторов "либо звонит, либо не звонит". В этом случае операторы все время подключены к Asterisk и если на очереди сейчас никого нет, то они слушают музыку. Если кто-то звонит с внешки и один из операторов свободен - то у оператора прекратится музыка, он слышит звуковой сигнал типа "бииб" и тут же начнет разговаривать с клиентом. Понятно, что для операторов call центра будет крайне неудобно работать с обычными телефонами и трубками. Для этого предназначаются микрофон, наушники и софтфон.

Более подробно и наглядно показано в видео.

5. Теперь рассмотрим более подробно строчки, которые мы понаписали в конфигах.

В agents.conf:

а) agent => 1001,123,Mikhail и agent => 1002,123,Alexander - как раз те самые агенты.

То-есть 1001 и 1002 как раз и есть агенты (внутренние номера сотрудников) которые и есть наши менеджеры отвечающие на звонки.

123 и 456 - пароли для агентов

Ну а Mikhail и Alexander это имена. Эти имена - произвольные, пишите какие хотите, хоть абра-кадабру, только на английской раскладке клавиатуры. Ни на что не влияет, это просто памятка для вас.

б) autologoff=15 - это время в секундах. Работает так: если кто-то с внешки стоит в очереди, а оператор который совершенно свободен не берет трубку - этого оператора выкидывают из системы (чуть позже будет более понятно, что значит "выкидывает из системы")

в queues.conf:

[operators] - секция, которую мы вызываем с помощью команды exten => s,5,Queue(operators,,,,100)

music = default - музыка, которая будет проигрываться человеку, стоящему в очереди на ожидание. У нас все такая же LWR, которую мы поставили в качестве музыки вместо гудка давным-давно. Вы на ожидание можете поставить, например, рекламные анонсы, хотя лично меня это напрягает. О том как это делается можно узнать в самом начале файла в разделе "установка музыки вместо гудка", пункт д. Тобишь вместо default у Вас будет другой параметр.

strategy = ringall - это стратегия. Вот какие стратегии бывают:

ringall: вызываются все доступные участники до тех пор, пока кто-то из них не ответит на вызов (по умолчанию).

rrmemory: циклически вызывается каждый из доступных участников.

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.

context = queue-out - Контекст, который будет использован, если звонящий нажал какую-либо цифровую кнопку, пока находится в очереди. Я не использовал этот параметр. Вы можете вывести его например опять в menu. То-есть если во время ожидания клиент нажмет на какую-нибудь кнопку, то опять услушит меню и сможет выбрать конкретного менеджера.

Например, если вы напишите контекст menu, то если во время ожидания клиент наберет внутренний номер сотрудника (в нашем случае это 1 или 2, то звонок пойдет на этого внутреннего сотрудника). Считаю не очень полезной функцией. Кроме того, нужно будет записать голосовое сообщение о том, что "во время ожидания Вы по-прежнему можете набрать внутренний номер сотрудника)

autofill = yes - (автозаполнение) позволяет Asterisk более эффективно распределять звонки между участниками обработки очереди вызовов, особенно если в очереди находятся несколько вызывающих абонентов и несколько агентов обработки вызовов могут принять звонок. Рекомендуется задавать для autofill значение yes.

announce-position = limit - если количество ожидающих в очереди больше 6, то позиция в очереди клиенту сообщаться не будет (чтобы его не пугать)

wrapuptime=50 - время в секундах. Объясню на примере: вы оператор. И вы только что поговорили с каким-либо человеком. Если в очереди еще кто-то есть, он тут же начнет с Вами разговор. Так вот, чтобы такого не было - есть время, а данном случае 50 секунд в течении которых звонок вам поступать не будет и вы немного отдохнете.

announce-frequency = 30 - через этот промежуток времени система будет напоминать человеку стоящему в очереди о том, что все пучком и скоро ему уделят внимание

announce-holdtime = yes - определяет, будет ли ожидающему человеку сообщать время, оставшееся до начала разговора с ним

joinempty = yes - пускать ли в очередь клиента, если все операторы сейчас разговаривают. Конечно же пускать! Для этого и существует очередь.

member => Agent/1001 и *member => Agent/1002* - здесь мы указываем агентов (операторов) которые будут входить в группу operators. В реальной организации агентов будет штук 10-20.

в) Теперь сам extensions.conf

exten => 800,1,AgentLogin() - если мы на телефоне набираем номер 800 то происходит вызов внутренней функции Asterisk'a AgentLogin(), которая позволяет сотруднику зарегистрироваться в качестве оператора и ожидать входящих звонков

exten => s,5,Queue(operators,,,,100) - когда клиенту проигрывается колосовое меню и он не выбирает какого-то конкретного оператора а просто остается на линии, то как раз и вызывается функция Queue которая и помещает клиент в очередь. При этом, Asterisk в файле queues.conf ищет секцию [operators] в которой и прописаны все параметры.

100 - время в секундах, в течении которого человек будет висеть на очереди. Если пройдет 100 секунд но никто так и не поговорит с ним - он автоматически пойдет дальше по диалплану. А по диалплану у нас автоответчик.

Кстати, перед это строчкой можно сделать дополнительное голосовое приветствие (например о том, что сейчас Вы будите помещены в очередь, но вовремя очередь Вы все равно можете вызвать сотрудника по его внутреннему номеру).

Строка будет выглядеть так:

exten => s,5,Background(/var/lib/asterisk/moh/voicemenu/ваш звуковой файл)

exten => s,6,Queue(operators,,,,100)

Существует еще много возможностей у нашего Call-центра. Например, операторов можно распределять в группы и устанавливать их приоритет. Например, определенная группа операторов относится к группе менеджеры, а вторая группа операторов относится к группе "тех. поддержка". Тех. поддержка будет отвечать только в том случае, если все менеджеры заняты.

Кроме того, существует встроенная система записи разговоров Call-центра, но здесь я её не показываю, поскольку разговоры у нас уже записываются автоматически, мы это настраивали в Dial плане

Более подробно по этим ссылкам:

http://asterisk.ru/knowledgebase/Asterisk+call+queues - Очереди вызовов в сервере Asterisk

http://asterisk.ru/knowledgebase/Asterisk+config+agents.conf - Файл конфигурации agents.conf

http://asterisk.ru/knowledgebase/Asterisk+config+queues.conf - Файл конфигурации queues.conf

http://asterisk.ru/knowledgebase/Asterisk+cmd+agentlogin - Команда AgentLogin()

http://voip.rus.net/tiki-index.php?page=Asterisk+cmd+Queue - Команда Queue()

http://r00ssyp.blogspot.ru/2013/01/asterisk.html - Asterisk. Простые очереди queues.

http://fb2.booksgid.com/content/FD/dzhim-meggelen-asterisk-buduschee-telefonii-vtoroe-izdanie/131.html - книга по Asterisk. Описание глобальных параметров queues.conf

8.5 Работа Asterisk в зависимости от дня недели и времени суток

Существует возможность сделать так, чтобы во вне рабочее время телефоны просто так не звонили, а клиенту проигрывалось сообщение о том, что сейчас не рабочее время с возможностью оставить сообщение на голосовую почту.

Общий принцип работы такой:

1) Если не рабочее время - проигрываем сообщение о том, что время не рабочее и просим оставить сообщение

2) Если время рабочее - Asterisk работает в привычном режиме.

Реализуем!

а) nano /etc/asterisk/extensions.conf

В этом файле найдем контекст [incoming] и заменим его содержимое следующим:

```
exten => _X.,1,GotoIfTime(00:00-23:59|sat-sun|*|*?workinghours,s,1)
exten => _X.,2,GotoIfTime(18:00-09:00|mon-fri|*|*?workinghours,s,1)
exten => _X.,3,Goto(menu,s,1)
```

б) После контекста incoming добавим новый контекст workinghours со следующим содержимым:

```
[workinghours]

exten => s,1,Background(/var/lib/asterisk/moh/worktime/workingtime)
exten => s,2,Set(fname=${STRFTIME(${EPOCH},,%Y%m%d%H%M)}-${CALLERID(number)}-${EXTEN})
exten => s,3,Record(/var/www/html/voicemail/${fname}.wav,0,8,X)
exten => s,4,Hangup
```

в) Рассмотрим эту связку контекстов. Строка:

```
exten => _X.,1,GotoIfTime(00:00-23:59|sat-sun|*|*?workinghours,s,1)
```

работает следующим образом: если в указанный промежуток времени (00:00-23:59) с субботы по воскресенье (sat-sun) поступает звонок с внешки, то начинает выполняться контекст workinghours.

А в контексте [workinghours] уже проигрывается сообщение о том, что сейчас не рабочее время и включается функция автоответчика.

Далее, строка `exten => _X.,2,GotoIfTime(18:00-09:00|mon-fri|*|*?workinghours,s,1)` делает тоже самое, но с 18 часов вечера до 9 утра с понедельника по пятницу.

Сообщение проигрывается с помощью строки:

```
exten => s,1,Background(/var/lib/asterisk/moh/worktime/workingtime)
```

о том, как загрузить музыку(голосовое сообщение), как её перекодировать под формат Asterisk написано во втором разделе (установка музыки вместо гудка). Единственное, pscp воспользоваться не получиться, поскольку мы намутили с системами защиты. Лучше ту папку, куда нам нужно залить голосовое приветствие расшарить через сервер samba.

Если же входящий звонок не подпадает ни под одно из условий (рабочее время), то выполняется строка `exten => _X.,3,Goto(menu,s,1)` которая запускает обычное голосовое меню (начинает выполняться контекст menu)

Таким образом, телефоны в нашей организации будет звонить только с понедельника по пятницу с 9 до 18. В любое другое время мы получим сообщение о том, что сейчас не рабочее время. Таким же образом можно сделать, например чтобы ночью звонила другая группа телефонов если в контексте [workinghours] написать соответствующий диалплан.

Статьи по этому делу:

http://voip.rus.net/tiki-index.php?page=Asterisk+cmd+GotoIfTime - функция GotoIfTime

http://shub123.ucoz.ru/sokrashenia.html - Английские сокращения названий месяцев (в середине страницы)

9. Заключение

Мы, с Вами, не зная Linux, с абсолютного нуля подняли мощнейшую систему АТС IP телефонии Asterisk. Наша АТС отвечает всем современным требованиям безопасности и обеспечивает почти максимально возможный функционал. Мы создали полноценный продукт готовый к внедрению в организацию. Вам остается лишь немного оптимизировать Ваш созданный сервер под конкретные задачи конкретной организации.