



ASTERCONF
– 2018

Автоматизированное тестирование конфигурации сервера Asterisk и не только.

Евгений Тевелевич
г.Томск



Тестирование в ручную

Когда без автоматизации:

- Настраиваем новую функцию
- Пытаемся повторить действия пользователей
- Изучаем, что то новое



Проблемы ручного тестирования

- Не удобно одновременно использовать много телефонов
- Сложно повторять одинаковую последовательность действий
- Процесс не документируется
- Не возможно проверить все

Меняем человека на робота



На роль робота выбираем:

- **Asterisk**
- **pjsua** (консольный sip-клиент)
- **sipp** утилита для генерации sip трафика.

Инструменты тестирования

Asterisk

- Известный инструмент
- Контроль вызовами в реальном времени
- Логирование вызовов
- Обращение к внешним сервисам
- Не все возможности телефона эмулирует
- Требуется отдельная установка

Инструменты тестирования

Pjsua

- Легко устанавливается
- Простой интерфейс
- Полностью эмулирует работу телефона
- Нет контроля за вызовом во время звонка*
- Не ведет лог звонка**

Инструменты тестирования

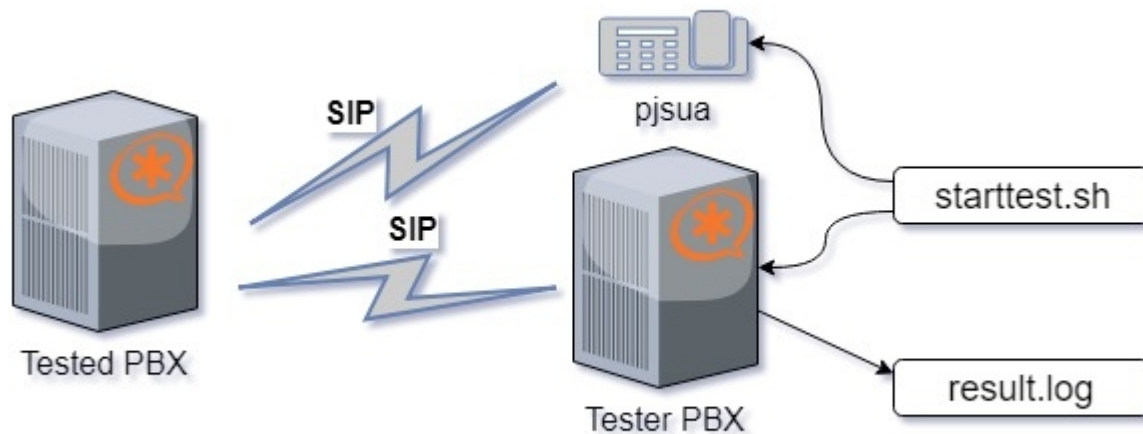
Sipp

- Низкоуровневая функциональность SIP

Использоваться для нагрузочного тестирования

- Сложен в освоении (требует хороших знаний SIP-протокола)

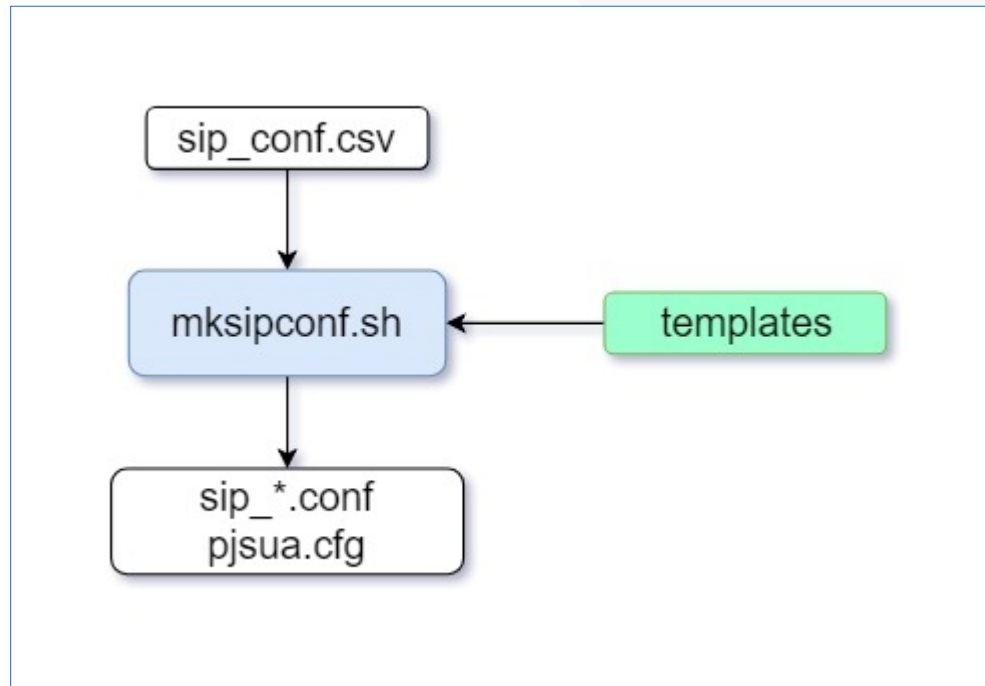
Схема тестирования



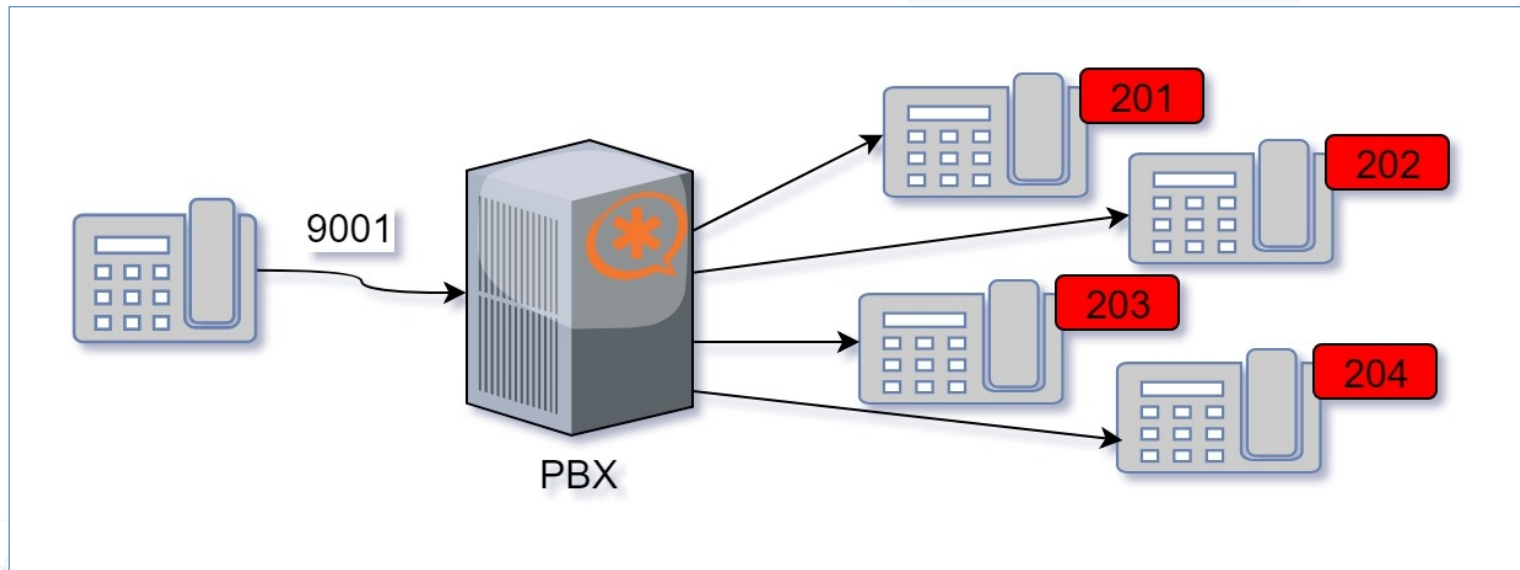
Структура папок тестов

```
pbxtest
├── 01_dial2group
│   ├── cfg
│   │   └── asterisk
│   │       ├── extensions_test.conf
│   │       ├── sip_addreg.conf
│   │       └── sip_reg.conf
│   └── starttest.sh
├── ...
├── 03_attended_transfer_sendpid
│   ├── cfg
│   │   ├── asterisk
│   │   │   ├── extensions_test.conf
│   │   │   ├── sip_addreg.conf
│   │   │   └── sip_reg.conf
│   │   └── pjsua
│   │       ├── 209.cfg
│   │       └── 209.input
│   └── starttest.sh
└── runalltest.sh
```

Генерация конф. для SIP-клиентов



Пример 1: групповой звонок



Пример 1: групповой звонок

```
[in-test]
exten => 9001,1,NoOp()
same => n,Dial(SIP/201&SIP/202&SIP/203&SIP/204,60)
same => n,HangUp()
```

```
NAMETEST=groupdial
TESTVAR=201-202-203-204
TRUNK=SIP/208
```

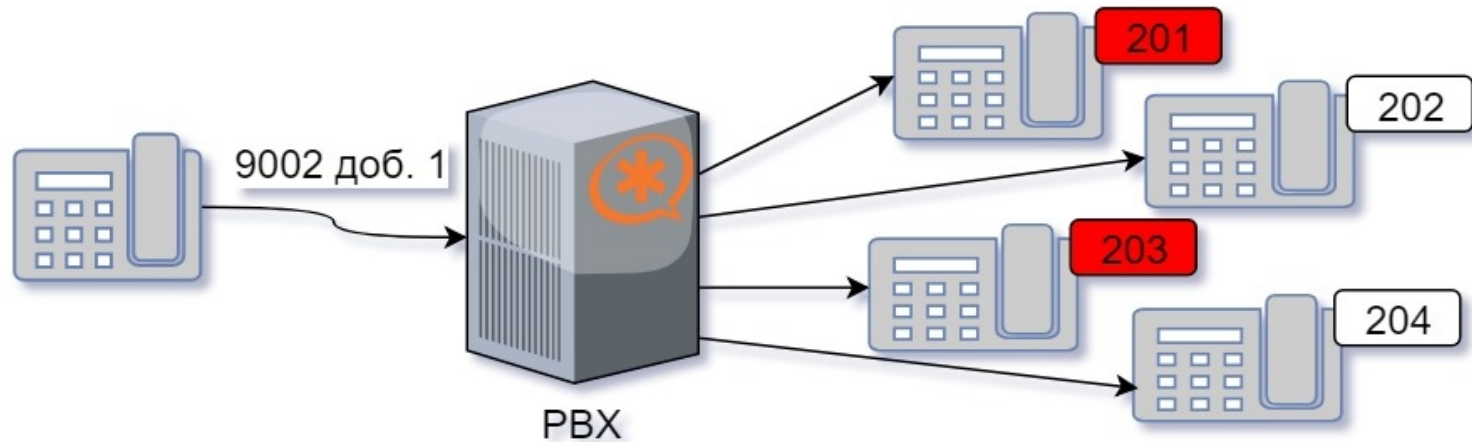
```
[test]
exten => test1,1,NoOp()
same => n,Set(CHANNEL(hangup_handler_push)=hdlrtest,s,1())
same => n,Answer()
same => n,Wait(30)
same => n,HangUp()
```

```
exten => test2,1,NoOp()
same => n,Dial(${TRUNK}/9001,20)
same => n,HangUp()
```

```
exten => _20X,1,NoOp()
same => n,Verbose(1,1 ==${EXTEN}====TESTVAR:${GLOBAL(TESTVAR)}====)
same => n,Set(GLOBAL(TESTVAR)=${LISTFILTER(GLOBAL(TESTVAR),-,${EXTEN})})
same => n,Verbose(1,2 ==${EXTEN}====TESTVAR:${GLOBAL(TESTVAR)}====)
same => n,Wait(0.2)
same => n,Answer()
same => n,Wait(0.5)
same => n,HangUp()
```

```
[hdlrtest]
exten => s,1,NoOp()
same => n,Set(testresult=${IF(${["${GLOBAL(TESTVAR)}"]=""}>PASS:FAIL)})
same => n,Set(msg=${NAMETEST}: ${testresult})
same => n,System(echo ${msg} >${PATHTEST}/result.log)
same => n,Return()
```

Пример 2: IVR



Пример 2: IVR

```
[in-test]
exten => 9002,1,NoOp()
same => n,Goto(ivr-test,s,1)

[ivr-test]
exten => s,1,NoOp()
same => n,Ringing()
same => n,Wait(1)
same => n,BackGround(/tmp/privet)
same => n,WaitExten(4)
same => n,HangUp()

exten => 1,1,NoOp()
same => n,Dial(SIP/201&SIP/203,60)

exten => 2,1,NoOp()
same => n,Dial(SIP/202&SIP/204,60)

exten => _20X,1,NoOp()
same => n,Dial(SIP/${EXTEN},60)
```

```
NAMETEST=ivr-select1
TESTVAR=201-203
TRUNK=SIP/208
```

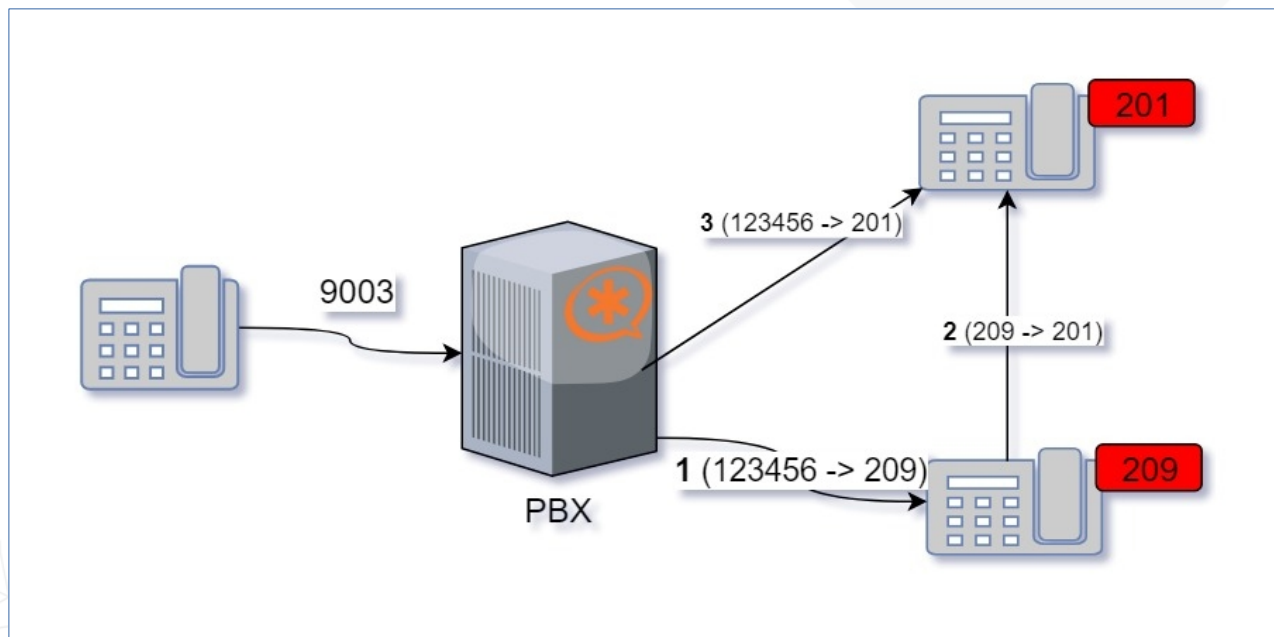
```
[test]
exten => test1,1,NoOp()
same => n,Set(CHANNEL(hangup_handler_push)=hdlrtest,s,1())
same => n,Answer()
same => n,Wait(3)
same => n,SendDTMF(2)
same => n,Wait(30)
same => n,HangUp()

exten => test2,1,NoOp()
same => n,Dial(${TRUNK}/9002,20)
same => n,HangUp()

exten => _20X,1,NoOp()
same => n,Verbose(1,1 ==${EXTEN}====TESTVAR:${GLOBAL(TESTVAR)}====)
same => n,Set(GLOBAL(TESTVAR)=${LISTFILTER(GLOBAL(TESTVAR),-,$${EXTEN})})
same => n,Verbose(1,2 ==${EXTEN}====TESTVAR:${GLOBAL(TESTVAR)}====)
same => n,Wait(0.2)
same => n,Answer()
same => n,Wait(0.5)
same => n,HangUp()

[hdlrtest]
exten => s,1,NoOp()
same => n,Set(testresult=${IF($["${GLOBAL(TESTVAR)}"=""]?PASS:FAIL)})
same => n,Set(msg=${NAMETEST}: ${testresult})
same => n,System(echo ${msg} >${PATHTEST}/result.log)
same => n,Return()
```

Пример 3: attendet transfer (sendpid)



Пример 3: attendet transfer

```
[in-test]
exten => 9003,1,NoOp()
same => n,Set(CALLERID(num)=123456)
same => n,Dial(SIP/209,60)
same => n,HangUp()

exten => _2XX,1,NoOp()
same => n,Dial(SIP/${EXTEN},60)
same => n,HangUp()
```

```
sleep 5000
a
200
sleep 1000
m
sip:201@testedpbx
sleep 2000
X
0
sleep 1000
h
sleep 5000
q
```

```
NAMETEST=attended_transfer_sendpid
TESTVAR=123456
TRUNK=SIP/208
```

```
[test]
exten => test1,1,NoOp()
same => n,Answer()
same => n,Wait(30)
same => n,HangUp()

exten => test2,1,NoOp()
same => n,Wait(2)
same => n,Dial(${TRUNK}/9003,20)
same => n,HangUp()

exten => 201,1,NoOp()
same => n,Set(CHANNEL(hangup_handler_push)=hdlrtest,s,1())
same => n,Wait(0.2)
same => n,Answer()
same => n,Verbose(1, 1==${CALLERID(num)} =====)
same => n,Wait(3)
same => n,Set(TESTVAR2=${CALLERID(num)})
same => n,Verbose(1, 1==${CALLERID(num)} =====)
same => n,HangUp()

[hdlrtest]
exten => s,1,NoOp()
same => n,Set(testresult=${IF(${GLOBAL(TESTVAR)}="${TESTVAR2}")?PASS:FAIL})
same => n,Set(msg=${NAMETEST}: ${testresult})
same => n,System(echo $msg >${PATHTEST}/result.log)
same => n,Return()
```

Результат тестов

```
groupdial: PASS  
ivr-select1: FAIL  
attended_transfer_sendpid: PASS
```

Использование тестов позволяет

- Проверить работоспособность конфигурации
- Реализовать быстро запускаемые повторяемые сценарии для разработки новой функциональности.
- Упростить решение проблем пользователей (пишем скрипт повторяющий действия пользователя, затем отлаживаем проблему)
- Документирование сценариев работы в скриптах для тестирования
- Тестирование перехода на новую версию asterisk
- Расширить возможности мониторинга



ASTERCONF
- 2018

СПАСИБО ЗА ВНИМАНИЕ!

Евгений Тевелевич

+7 (913) 824 5297

tg: @chan_asterisk

