Оглавление

# [Ссылки 2](#__RefHeading___Toc906_4165224408)

# [Список протестированных систем 2](#__RefHeading___Toc908_4165224408)

# [Описание каталогов и файлов 3](#__RefHeading___Toc910_4165224408)

## [Список каталогов 3](#__RefHeading___Toc912_4165224408)

## [Список файлов 5](#__RefHeading___Toc914_4165224408)

# [Подготовка к запуску и отправке 6](#__RefHeading___Toc1388_2352097021)

## [Создание gpg файла с паролем и ssh ключа 6](#__RefHeading___Toc1390_2352097021)

## [Файлы хостов 7](#__RefHeading___Toc1392_2352097021)

## [Файл конфигурации sssc.conf 8](#__RefHeading___Toc1444_2352097021)

## [Комментарии к параметрам файла sssc.conf 9](#__RefHeading___Toc1452_2352097021)

## [Дополнительные файлы для отправки 13](#__RefHeading___Toc1663_2352097021)

## [Файлы отправляемых скриптов 14](#__RefHeading___Toc1665_2352097021)

## [Описание файла script.conf 18](#__RefHeading___Toc1401_1050836455)

# [Работа со скриптом 22](#__RefHeading___Toc1403_1050836455)

## [Выполняемые проверки при запуске скрипта 22](#__RefHeading___Toc1405_1050836455)

## [Интерактивный режим (обычный запуск) 22](#__RefHeading___Toc1407_1050836455)

## [Неинтерактивный режим (запуск с параметрами) 27](#__RefHeading___Toc1409_1050836455)

## [Проверка наличия обновлений 30](#__RefHeading___Toc1014_1378055058)

**Скрипт удаленного администрирования Linux по SSH: SSSC**

**( Ssh Send Script Command )**

**Версия: 20250130\_01**

# Ссылки

GITFLIC: <https://gitflic.ru/user/medved0001>Вконтакте: <https://vk.com/medved0001>

# Список протестированных систем

Скрипт предназначен для удаленного администрирования Linux-систем по SSH. Он разрабатывается в первую очередь для личного использования на Astra Linux, но тестируется по возможности и на других системах.

Список протестированных систем (версия скрипта **20250130\_01**):

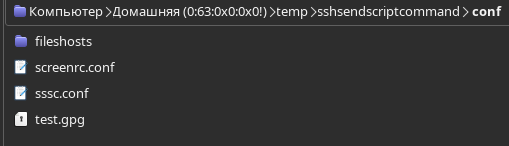
1. Astra Linux 1.8.1.6
2. Альт Рабочая Станция K 10.3
3. RedOS 8

На других системах работа не гарантируется (По мере необходимости и возможности будут доработки). По любым вопросам и проблемам со скриптом обращайтесь в личные сообщения Вконтакте.

# Описание каталогов и файлов

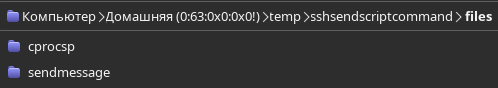
## **Список каталогов**

**Каталог conf**



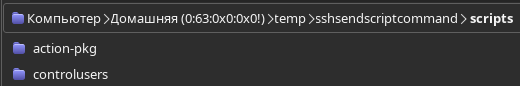
Каталог с файлами конфигураций

**Каталог files**



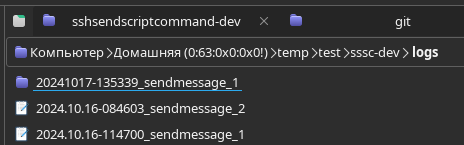
Каталог с дополнительными файлами для отправки (можно указать свой путь в файле конфигураций sssc.conf)

**Каталог scripts**



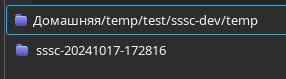
Каталог с отправляемыми скриптами (можно указать свой путь в файле конфигураций sssc.conf)

**Каталог logs**



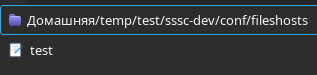
После выполнения скрипта в каталоге **logs** будет сохранен лог выполнения. Для отправки одним потоком лог файл создается в корне каталога **logs**, а при многопоточной отправке создается подкаталог в котором хранятся отдельные логи на каждый поток. Формат имени лог файла/каталога: (Дата-Время)\_(имя первого скрипта в списке)\_(количество отправляемых скриптов)

**Временный каталог temp**



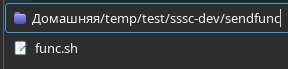
В каталоге **temp** (создается автоматически) хранятся временные файлы, необходимые для отправки. После выполнения созданные файлы автоматически удаляются (Файлы не удалятся в случае закрытия скрипта до завершения отправки после подготовки файлов для передачи).

**Каталог ./conf/fileshosts**



Каталог с файлами хостов

**Каталог sendfunc**



В каталоге **sendfunc** находятся файлы, которые будут переданы на удаленный компьютер, если вы укажите имена нужных файлов в массиве **list\_source\_func** в вашем скрипте (описание будет в примере скрипта).

## **Список** **файлов**

Файлы **remote-temprunscript-cron** и **remote-runprecommand.sh** не требуют ручных изменений для работы.

1. **remote-temprunscript-cron** - подготовленный шаблон cron задачи, который автоматически меняется командами из **remote-runprecommand.sh**, если выбран тип выполнения «**Выполнение в фоновом режиме через задачу cron**».
2. **remote-runprecommand.sh** - Файл содержит функции для запуска скрипта с автовводом пароля sudo и запуска через cron задачу. Сам файл не передается на удаленный компьютер, а выполняется напрямую по ssh в base64 виде.
3. **./conf/sssc.conf** — файл конфигурации скрипта
4. **./conf/screenrc.conf** – файл конфигурации **screen**

# Подготовка к запуску и отправке

## **Создание gpg файла с паролем и ssh ключа**

Создать gpg файл с паролем вы можете сами или через скрипт **create-gpgpassfile.sh**. Для создания ssh ключа используйте команду **ssh-keygen**.

При использовании ключей с паролем можно добавить следующий код в ваш **~/.profile** для однократного введения пароля ключа:

if [ -n "$(which ssh-agent)" ]; then

if [ ! -S ~/.ssh/ssh\_auth\_sock ]; then

if [ "$(pgrep -f ssh-agent -u "$USER" | wc -l)" -gt "0" ]; then

for pid\_value in $(pgrep -f ssh-agent -u "$USER"); do

kill -9 $pid\_value

done

fi

eval `ssh-agent 2>/dev/null`

mkdir -p ~/.ssh

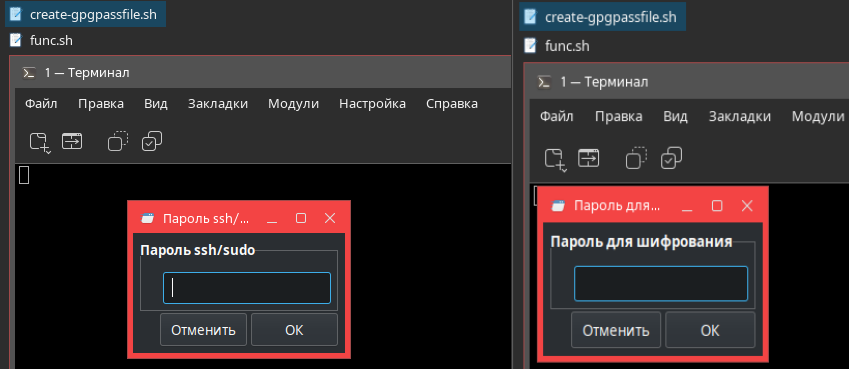
ln -sf "$SSH\_AUTH\_SOCK" ~/.ssh/ssh\_auth\_sock

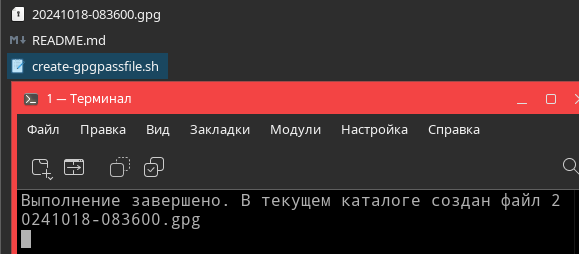
fi

export SSH\_AUTH\_SOCK=~/.ssh/ssh\_auth\_sock

fi

Таким образом расшифрованный ключ будет храниться в памяти до завершения сессии/перезагрузки или пока вы не закроете процесс **ssh-agent**.

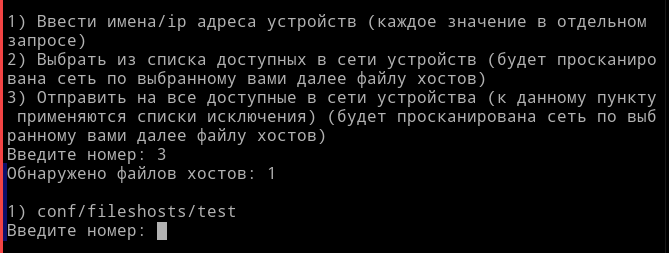




## **Файлы хостов**

Файлы хостов используются для следующих типов формирования списка устройств:

1. Выбор из списка доступных в сети устройств
2. Отправка на все доступные в сети устройства



Создайте файл в каталоге **./conf/fileshosts.** В каждой строке файла должна быть одна запись. Для сканирования доступных устройств в сети используется программа **nmap**, поэтому синтаксис добавления такой же, как у **nmap**. Строки можно комментировать с помощью символа #.

Пример содержимого файла:

localhost

#10.0.50.50-180

10.0.50.218

10.0.51.1/24

## **Файл конфигурации sssc.conf**

Заполните файл конфигурации **sssc.conf**. В файле может быть несколько преднастроенных секций, между которыми можно переключаться. Пример настроек:

usesection=

[tests2]

exec\_no\_display=0

no\_check\_update\_script=0

dirscripts=

dirfiles=

path\_exec\_script\_version=/home/test1/.listscripts

sutype=sudo

typeterminalmultiplexer=tmux

typesendfiles=rsync

skipchangescriptfile=0

loginname=test1

multisend=3

sshConnectTimeout=5

numportssh=1500

sshtypecon=pas

sshkeyfile=

gpgfilepass=conf/test.gpg

gpgpass=12345678

remotedirrunscript=/home/test1/.rstmp

remotedirgroup=astra-admin

listIgnoreInaccurate=test1;test2;10.0.50

listIgnoreAccurate=10.0.51.1;10.0.51.2;test.dc.local

reboot\_max\_try\_wait\_devaice=50

reboot\_time\_wait\_devaice=10

[имя секции]

параметры

…

## **Комментарии к параметрам файла sssc.conf**

**usesection** - В **usesection** задается имя используемой секции настроек. Должен находится в первой строке файла **sssc.conf**. Если переменная не задана, содержит недопустимые символы или указанная секция не найдена, то при запуске скрипта будет выдан список секций для выбора. Условие для проверки значения: **'^[A-Za-zА-Яа-я0-9(),.@\_[:space:]-]+$'**

**exec\_no\_display** -Выполнение без использования **DISPLAY**. Для запросов будет использоваться **dialog** с псевдографикой вместо **zenity** (0 - отключить, 1 - включить). Если значение **DISPLAY** отсутствует, то будет принудительно включено, иначе заданное значение или по умолчанию отключено.

**no\_check\_update\_script** - Для отключения проверки обновлений скрипта укажите значение 1.

**dirscripts** - Относительный или полный путь к каталогу со скриптами (если не указан или каталог не существует, то по умолчанию используется каталог **scripts** в каталоге запуска). Условие для проверки значения: **'^[A-Za-zА-Яа-я0-9(),./@\_[:space:]-]+$'**

**dirfiles** - Относительный или полный путь к каталогу со дополнительными файлами для выбора при отправке (если не указан или каталог не существует, то по умолчанию используется каталог **files** в каталоге запуска). Условие для проверки значения: **'^[A-Za-zА-Яа-я0-9(),.**[**/@\_**](../../../../../@_)**[:space:]-]+$'**

**path\_exec\_script\_version** - Путь к текстовому файлу на удаленном компьютере, куда будет записана версия успешно выполненного скрипта в случае выполнения сравнения версии отправляемого и выполненного ранее скрипта (**необязательная переменная. Файл должен быть доступен для чтения и записи. Рекомендуется не использовать длинный путь**). Если файл не существует, будет создан при наличии доступа. Если не заполнено, то возможность сравнения версии будет отключена. Условие для проверки значения: **'^[A-Za-zА-Яа-я0-9(),./@\_[:space:]-]+$'**

**sutype** - Тип повышения прав. Если **loginname** равно **root**, то можно оставить пустым, т.к. автоматически будет присвоено значение **root**. У остальных пользователей, если не заполнено, будет выбор при запуске скрипта. На текущий момент для повышения прав доступен только **sudo**.

**typeterminalmultiplexer** - Тип многооконного терминала при многопоточной рассылке без параметров (**screen** или **tmux**. Рекомендуется **tmux**).

**typesendfiles** - Тип отправки файлов (**scp** или **rsync**. Рекомендуется **rsync**, но он должен быть установлен на принимающем ПК. **scp** устанавливается с **ssh**).

**skipchangescriptfile** - Пропуск внесения изменений в переменные отправляемого скрипта при отправке, если присутствует **script.conf** (0 - нет, 1 - да). Если не заполнено, будет выбор при запуске скрипта.

**loginname** - Логин пользователя (если не заполнен, запросит при запуске скрипта). Условие для проверки значения: **'^[A-Za-zА-Яа-я0-9@.\_-]+$'**

**multisend** - Количество потоков рассылки (по умолчанию 1, если значение не задано или некорректно). Условие для проверки значения: **'^[0-9]+$'**

**sshConnectTimeout** - Таймаут на подключение к ПК по SSH (по умолчанию 5 секунд, если значение не задано или некорректно). Условие для проверки значения: **'^[0-9]+$'**

**numportssh** - Номер SSH порта (по умолчанию 22, если значение не задано или некорректно). Условие для проверки значения: **'^[0-9]+$'**

**sshtypecon** - Тип SSH подключения (допустимые значения: **pas** - пароль, **key** - ключ. Рекомендуется подключение по ключу).

**sshkeyfile** - Полный путь от корня к файлу закрытого ключа ssh или относительный путь от каталога запуска скрипта. Например: полный путь «**/home/user01/temp/test/usersshkey**» и относительный «**conf/usersshkey**». Если «**sshtypecon**» равно «**pas**», можно оставить пустым. В остальных случаях если не заполнен, содержит недопустимые символы или файл не существует, то запросит выбор файла при запуске скрипта. Условие для проверки значения: **'^[A-Za-zА-Яа-я0-9(),./@\_[:space:]-]+$'**

**gpgfilepass** - GPG файл с паролем подключения SSH и sudo (**пароль может содержать любые символы**). Если логин пользователя «**root**», а тип подключения «**key**», то можно оставить пустым. В остальных случаях необходимо указать полный путь от корня или относительный путь от каталога запуска скрипта. Например: полный путь «**/home/user01/temp/test/test.gpg**» и относительный «**conf/test.gpg**». Если не заполнен, содержит недопустимые символы или файл не существует, то запросит выбор файла при запуске скрипта. Условие для проверки значения: **'^[A-Za-zА-Яа-я0-9(),./@\_[:space:]-]+$'**

**gpgpass** - Пароль от GPG файла (**пароль может содержать любые символы**). Если логин пользователя «**root**», а тип подключения «**key**», то можно оставить пустым. В остальных случаях если не заполнен, запросит при запуске скрипта.

**remotedirrunscript** - Каталог на удаленном компьютере для передачи файлов (**Каталог должен быть доступен для чтения, записи и выполнения. Рекомендуется не использовать длинный путь. Если не существует, будет создан при наличии доступа**). Условие для проверки значения: **'^[A-Za-zА-Яа-я0-9(),./@\_[:space:]-]+$'**

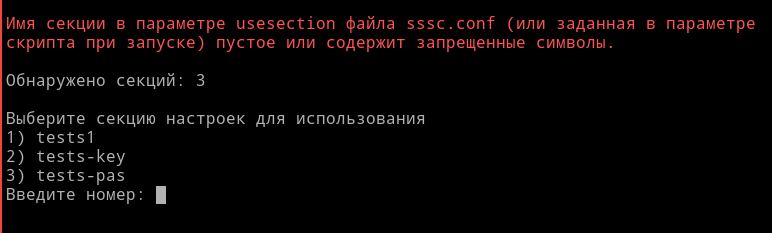
**remotedirgroup** - Имя группы, которая будет выставлена в правах на конечный каталог, если он будет создан при выполнении скрипта. Если группы не существует, то останутся текущие права (**отправляемый скрипт запустится в любом случае**). Смена группы в правах на каталог нужна, если предполагается использовать один каталог "**remotedirrunscript**" для выполнения разными пользователями включенными в одну группу. Условие для проверки значения: **'^[A-Za-zА-Яа-я0-9@.\_-]+$'**

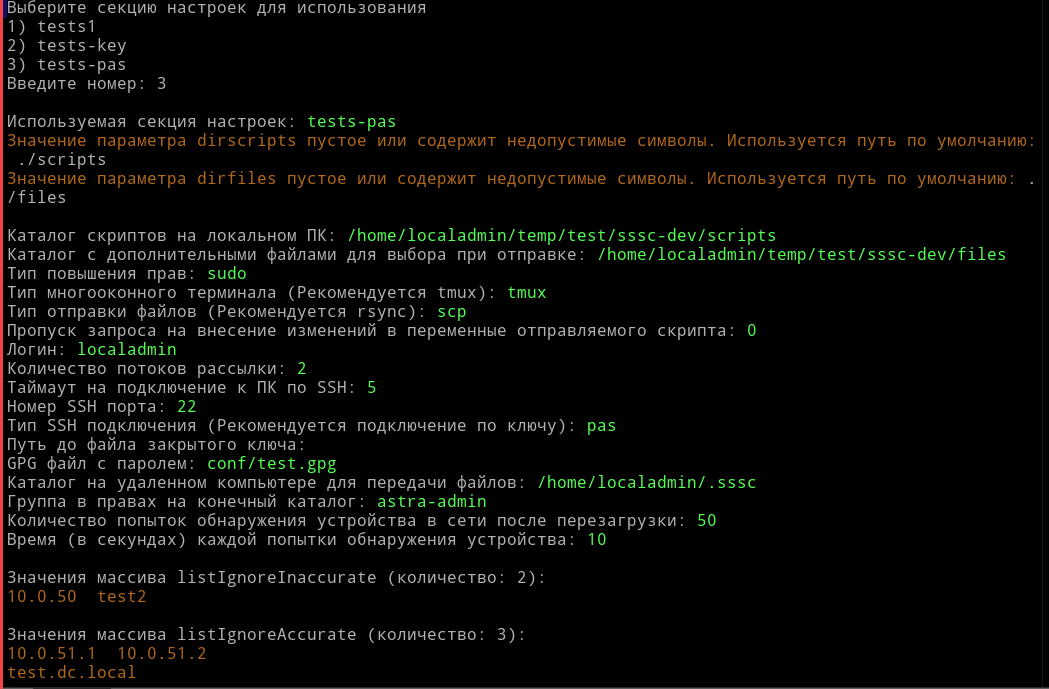
**listIgnoreInaccurate** - Записи для исключения хостов по частичному совпадению (необязательная переменная) (**указать значения в строку разделяя точкой с запятой «;»**). Условие для проверки значения: **'^[A-Za-zА-Яа-я0-9.-]+$'**

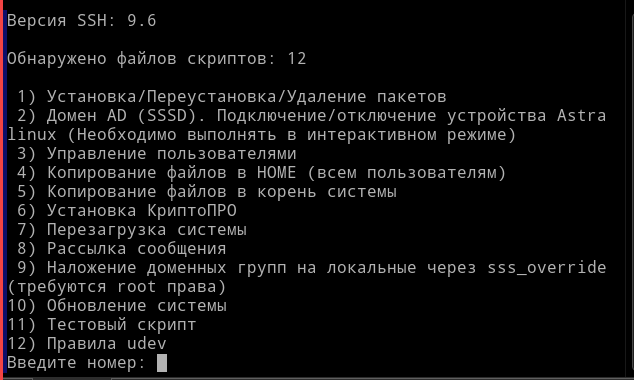
**listIgnoreAccurate** - Записи для исключения хостов по точному совпадению (необязательная переменная) (**указать значения в строку разделяя точкой с запятой «;»**). Условие для проверки значения: **'^[A-Za-zА-Яа-я0-9.-]+$'**

**reboot\_max\_try\_wait\_devaice** - Количество попыток обнаружения устройства в сети после перезагрузки (**по умолчанию 50, если значение не задано или некорректно**). Условие для проверки значения: **'^[0-9]+$'**

**reboot\_time\_wait\_devaice** - Время (в секундах) каждой попытки обнаружения устройства (**по умолчанию 10, если значение не задано или некорректно**). Условие для проверки значения: **'^[0-9]+$'**





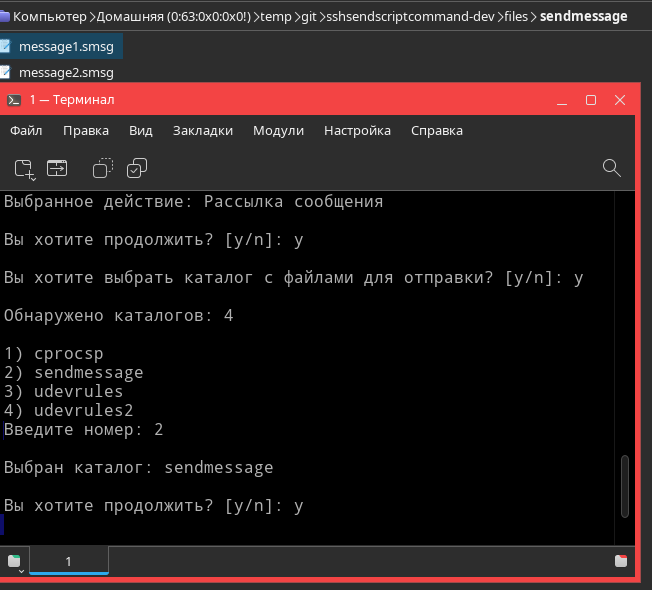


## **Дополнительные файлы для отправки**

Если для выполнения скрипта требуются дополнительные файлы, то необходимо сделать следующее:

1. Создать каталог в каталоге **./files** (или указанном вами в файле конфигурации) и поместить в него необходимые файлы.
2. Выбрать необходимый каталог при отправке.

В своем скрипте вы можете обратится к этому каталогу через переменную **dirfiles** (Переменная **dirfiles** будет автоматически добавлена в скрипт, если выбран каталог с дополнительными файлами для отправки).



## **Файлы отправляемых скриптов**

В каталоге **./scripts** (или указанном вами в файле конфигурации) находятся каталоги с отправляемыми скриптами.

Для добавления своего скрипта в список выбора необходимо создать каталог в **./scripts** и поместить в него скрипт с названием **sendcommand**. Например, **./scripts/testscript/sendcommand.** Подкаталоги **./scripts** сканируются на наличие файла **sendcommand**. Наименование скрипта для списка задается в переменной скрипта **namescript**.

Пример файла скрипта:

#!/bin/bash

namescript="Тестовый скрипт"

#Версия скрипта

let scriptversion=1

#Список файлов, которые необходимо отправить из каталога sendfunc

list\_source\_func=('func test1.sh' 'func test2.sh')

#Подключение переданных файлов list\_source\_func

if [[ "${#list\_source\_func[@]}" -gt "0" ]]; then

for ((num\_list\_func = 0; num\_list\_func < ${#list\_source\_func[@]}; num\_list\_func++)); do

if [[ -f "$(dirname "$(realpath "$0")")/${list\_source\_func[$num\_list\_func]}" ]]; then

source "$(dirname "$(realpath "$0")")/${list\_source\_func[$num\_list\_func]}"

else

echo "Файл (${list\_source\_func[$num\_list\_func]}) не найден"

exit 1

fi

done

fi

#Разместите ниже код вашего скрипта

#Тест каталога переданных файлов

#Переменная dirfiles будет автоматически добавлена в скрипт, если выбран каталог с дополнительными файлами для отправки

echo "Путь к каталогу дополнительных файлов: $dirfiles"

echo ""

#Тест переданных файлов list\_source\_func

ip\_info\_func1

echo ""

uname\_info\_func2

#Тест внесения изменений в переменные

test\_number=

echo "Введенное число: $test\_number"

echo ""

test\_truefalse=

echo "Выбор 0/1: $test\_truefalse"

echo ""

test\_list\_values=

echo "Выбор из списка: $test\_list\_values"

echo ""

test\_text=

echo "Текстовое значение: $test\_text"

echo ""

test\_massive=

echo "Значений в массиве: ${#test\_massive[@]}"

echo "Значения массива:"

for ((v = 0; v < ${#test\_massive[@]}; v++)); do echo "${test\_massive[$v]}"; done

echo ""

test\_errscript="0"

if [[ "$test\_errscript" -eq "1" ]]; then

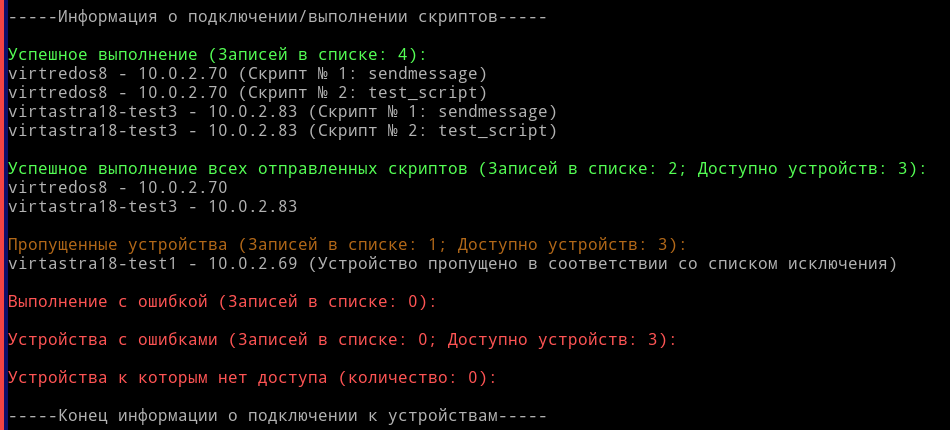
echo -e "Тест завершения скрипта с exit 1\n"

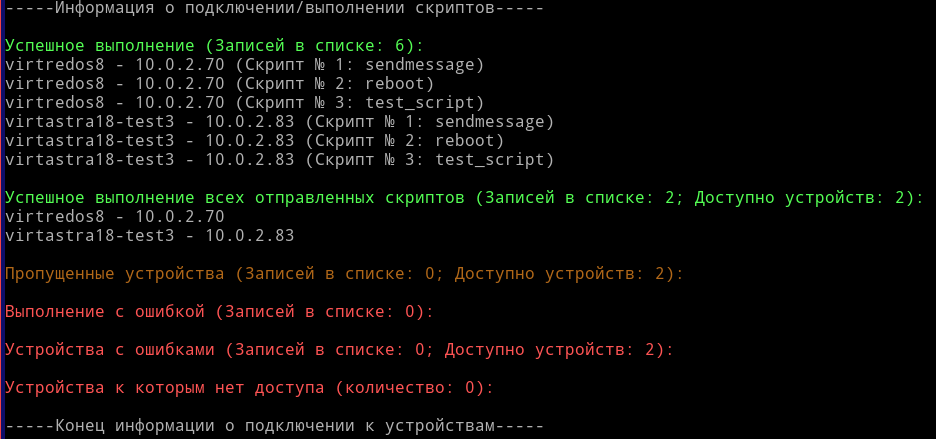
exit 1

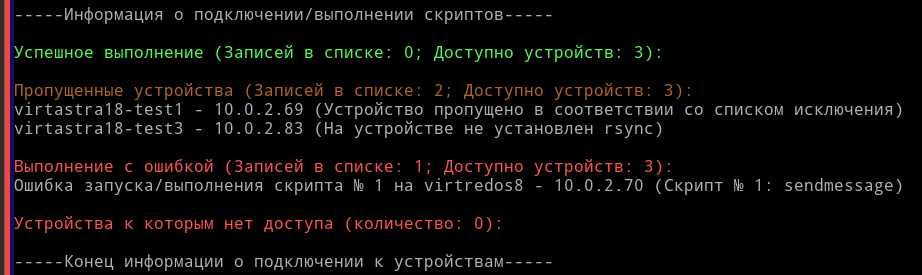
fi

#Конец кода вашего скрипта

Если вы поместите в скрипт **exit 1** или выполняемая вами команда присвоит код ошибки, то информация об этом будет в разделе «**Выполнение с ошибкой**» в конце лога.







## **Описание файла script.conf**

Файл **script.conf** используется для указания переменных, из файла скрипта **sendcommand**, которые вы хотите иметь возможность менять в копии скрипта при отправке (т.е. нет необходимости менять значения в эталонном файле скрипта вручную).

При подготовке файлов к отправке, если файл **script.conf** есть в каталоге с отправляемым скриптом, будет задан вопрос, хотите ли вы внести изменения в переменные копии файла отправляемого скрипта (вопрос можно отключить через параметр **skipchangescriptfile** в **sssc.conf**).

Формат заполнения:

[имя-переменной]

typevalue=тип-переменной

descvalue=Описание

listvalue=значение1;значение2

Поле **listvalue** обязательно, если выбран тип переменной **list**

Допускается использовать следующие типы переменной:

1. number - любое число
2. text - текстовое поле
3. list - выбор из заданного списка значений
4. massive - при выборе данного типа будет запрос значений для массива
5. truefalse - выбор из значений 0/1

Пример:

[test\_number]

typevalue=number

descvalue=Введите любое число

[test\_truefalse]

typevalue=truefalse

descvalue=Выберите одно из значений (0 или 1)

[test\_errscript]

typevalue=truefalse

descvalue=Тест завершения скрипта с кодом exit 1. Выберите одно из значений (0 или 1)

[test\_list\_values]

typevalue=list

descvalue=Выберите значение из списка

listvalue=тест 1;test 2;тест `" %значение '$dirfiles № \*)

[test\_text]

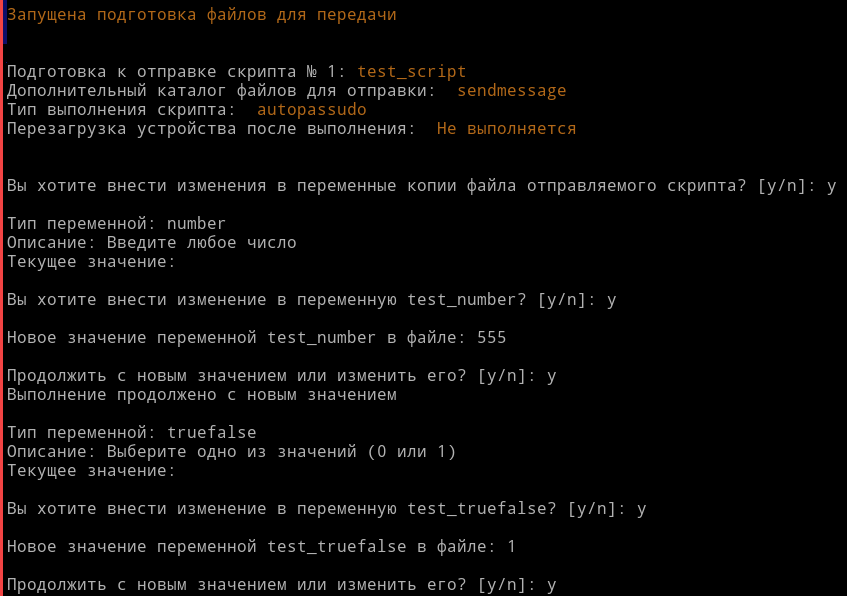
typevalue=text

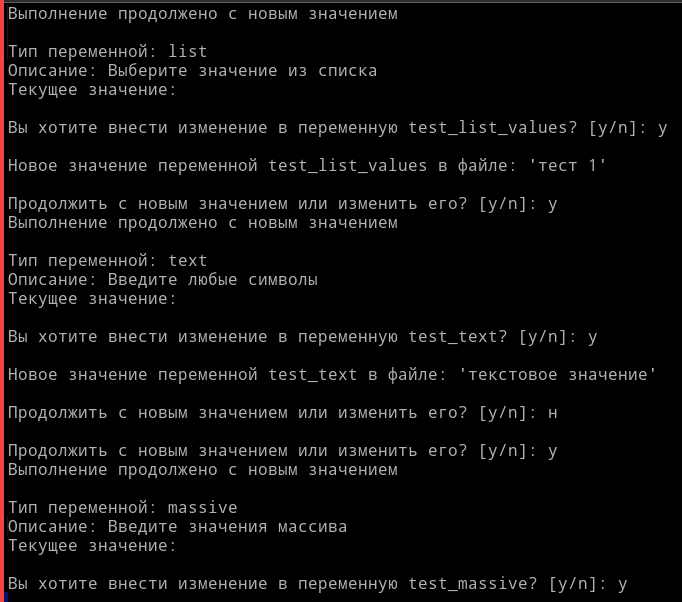
descvalue=Введите любые символы

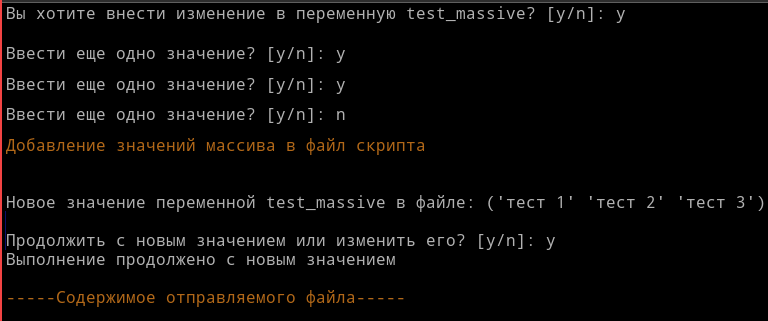
[test\_massive]

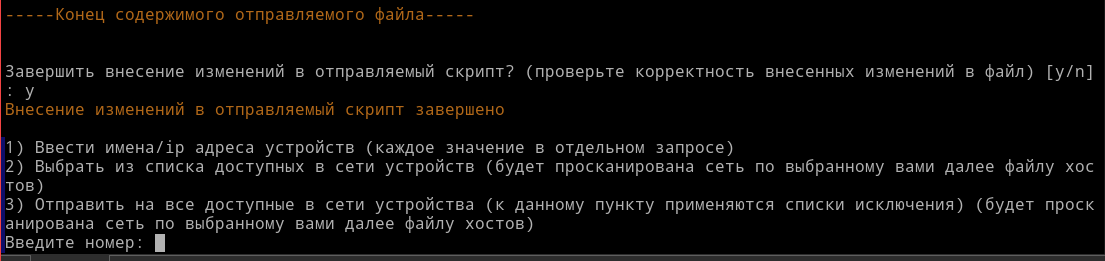
typevalue=massive

descvalue=Введите значения массива









# Работа со скриптом

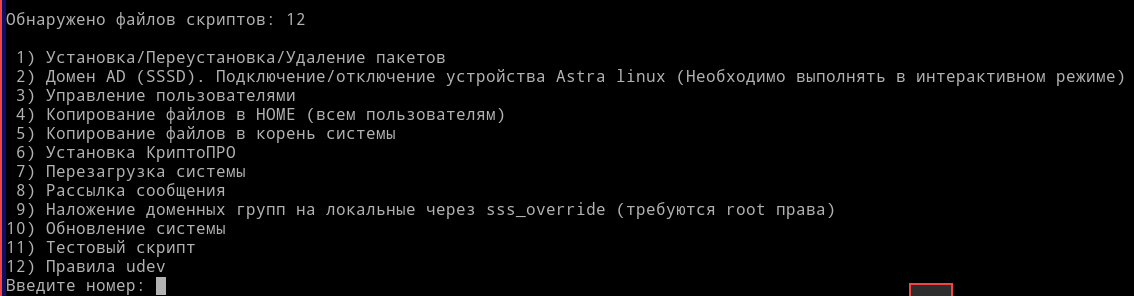
## **Выполняемые проверки при запуске скрипта**

При запуске скрипт проверяет:

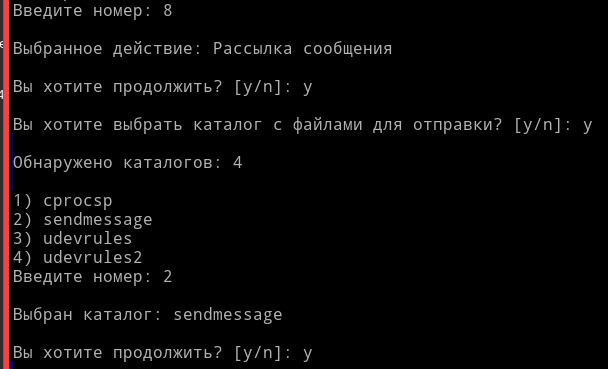
1. Путь к каталогу запускаемого скрипта на соответствие разрешенным символам.
2. Наличие файла **func.sh** и его хеш-сумму.
3. Наличие файлов: **remote-runprecommand.sh**, **sssc.conf**, **screenrc.conf**, **remote-temprunscript-cron.**
4. Хеш-суммы файлов: **remote-runprecommand.sh** и **remote-temprunscript-cron**.
5. Наличие необходимых исполняемых файлов. Если они не найдены, предлагается проверить и установить необходимые пакеты.
   * Проверяемые исполняемые файлы: **nmap**, **sshpass**, **gpg2**, **awk**, **sed**, **grep**, **tmux** или **screen**, **zenity**, **ssh**, **scp** или **rsync**, **ls**, **cut**, **rev**, **cat**.
   * Проверяемые пакеты в случае установки: **nmap**, **sshpass**, **gnupg2** или **gnupg2-gostcrypto**, **gawk**, **sed**, **grep**, **tmux**, **screen**, **zenity**, **openssh-client** или **openssh-clients**, **rsync**
6. Наличие секций настроек и проверка наименования секций в файле **sssc.conf**.
7. Параметр **usesection** из файла **sssc.conf**. Если значение пустое, содержит недопустимые знаки или такая секция не найдена, то будет предложен выбор из найденных секций.

## **Интерактивный режим (обычный запуск)**

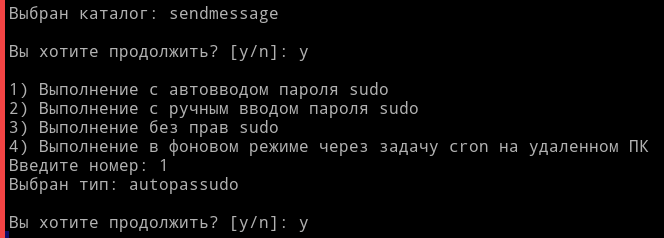
1. Запустите файл **run-sssc.sh** При успешном прохождении всех проверок и считывании настроек отобразится список доступных для отправки скриптов. Если список будет пуст, то скрипт завершит работу.
2. Выберите скрипт для отправки.



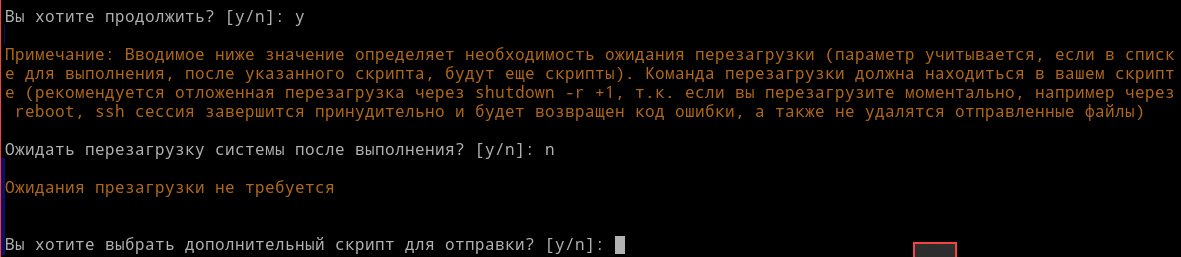
1. При необходимости выберите каталог с дополнительными файлами для отправки.



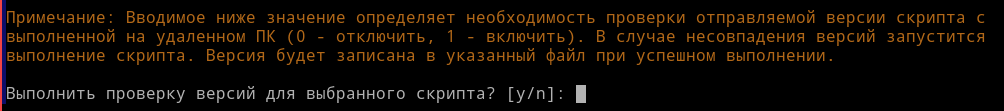
1. Выберите тип выполнения скрипта



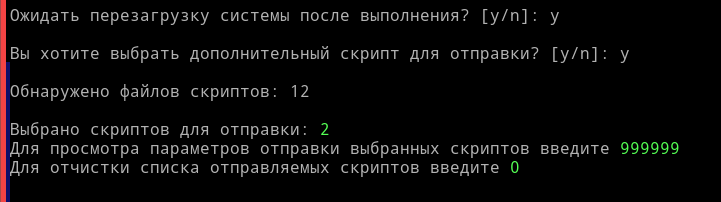
1. Ответить на вопрос об ожидании перезагрузки (параметр учитывается, если после этого скрипта будут еще скрипты для выполнения).

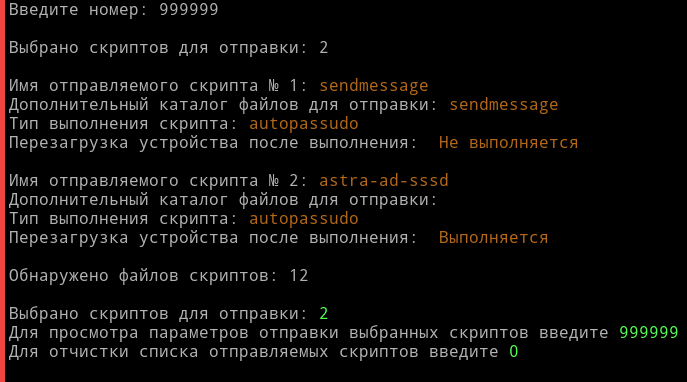


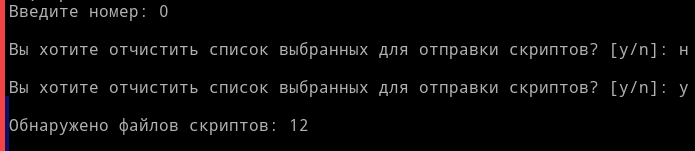
1. Если в файле конфигурации заполнено значение **path\_exec\_script\_version** и в отправляемом скрипте есть переменная **scriptversion** с номером версии, то будет задан вопрос о необходимости сравнения версии отправляемого скрипта с версией выполненной на удаленном устройстве.



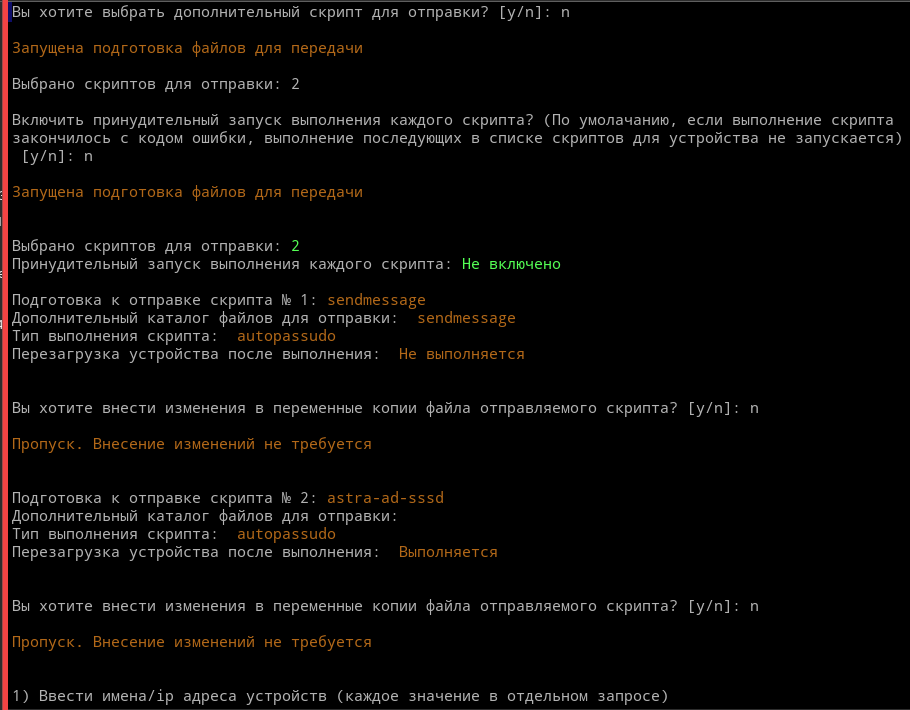
1. Ответить на вопрос о добавлении еще одного скрипта в список отправки. Если выбран вариант добавления дополнительного скрипта в список отправки, то скрипт возвращается к пункту № 2. Повторяем пункты 2-6 нужное количество раз. Можно посмотреть выбранные параметры отправляемых скриптов введя **999999.** Отчистить список и начать заново можно введя **0.**



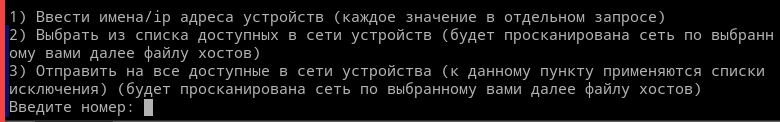




1. Перед подготовкой файлов к отправке, если выбрано более одного скрипта, будет вопрос о необходимости принудительного запуска каждого скрипта, в случае ошибки выполнения предыдущего.
2. При подготовке файлов к передаче, если у отправляемого скрипта присутствует файл **script.conf**, будет задан вопрос об изменении переменных в копии отправляемого скрипта (подробнее в разделе «**Описание файла script.conf»**)



1. Выберите вариант формирования списка устройств для отправки



## **Неинтерактивный режим (запуск с параметрами)**

Скрипт можно запустить из терминала с параметрами. Для вывода доступных параметров запуска нужно запустить скрипт с параметром **-help** (**./run-sssc.sh -help**).

Параметры запуска:

* + **-help**) Вызов справки.
  + **-fes**) Принудительно запустить выполнение каждого скрипта, если отправляется более одного скрипта. (По умолачанию, если выполнение скрипта закончилось с кодом ошибки, выполнение последующих в списке скриптов для устройства не запускается).
  + **-m**) Запуск выполнения через мультиплексор определенный в параметре конфигурации **typeterminalmultiplexer** (применимо к многопоточной отправке) (По умолчанию выполнение каждого потока запускается в фоновом режиме с выводом выполнения в текущее окно).
  + **-hf значение**) Имя используемого файла хостов из каталога **conf/fileshosts/**. Значение с пробелом необходимо заключить в двойные кавычки (Для выполнения команды обязательно должен быть указан один из параметров: **-hf** или **-hn**. При указании обоих параметров приоритет имеет **-hn**, т.е. значение **-hf** будет обнулено).
  + **-hn значение**) Имя или ip адрес устройства. Параметр может повторяться для указания дополнительных значений (Для выполнения команды обязательно должен быть указан один из параметров: **-hf** или **-hn**. При указании обоих параметров приоритет имеет **-hn**, т.е. значение **-hf** будет обнулено).
  + **-sp "Значение"**) - Параметр может повторяться для указания дополнительных значений. Значение имеет следующий вид "**имя каталога скрипта для отправки:имя каталога дополнительных файлов для отправки:тип выполнения скрипта:признак перезагрузки после выполнения скрипта:признак проверки версии скрипта**" (например: "**sendmessage:sendmessage:autopassudo:0:0**"). Описание параметров в порядке использования:

1. **Секция № 1** - Имя каталога отправляемого скрипта в каталоге **./scripts** (или определенный вами каталог в секции настроек файла **sssc.conf**). Обязательный параметр.
2. **Секция № 2** - Имя каталога в **./files** (или определенный вами каталог в секции настроек файла **sssc.conf**) для отправки на удаленный компьютер (Необязательный параметр).
3. **Секция № 3** - Тип выполнения скрипта (Обязательный параметр). Допустимые значения:
   * **autopassudo** - Выполнение с автовводом пароля sudo
   * **nopassudo** - Выполнение с ручным вводом пароля sudo
   * **nosudo** - Выполнение без прав sudo
   * **cronscript** - Выполнение в фоновом режиме через задачу cron на удаленном ПК
4. **Секция № 4** - Признак перезагрузки после выполнения скрипта (Необязательный параметр). Определяет необходимость ожидания перезагрузки (параметр учитывается, если в списке для выполнения, после указанного скрипта, будут еще скрипты). Команда перезагрузки должна находиться в вашем скрипте (рекомендуется отложенная перезагрузка через **shutdown -r +1**, т.к. если вы перезагрузите моментально, например через **reboot**, ssh сессия завершится принудительно и будет возвращен код ошибки, а также не удалятся отправленные файлы). Допустимые значения:
   * 0 - ожидание перезагрузки не выполняется
   * 1 - дождаться перезагрузки системы
5. **Секция № 5** - Признак проверки версии скрипта (0 - отключить, 1 - включить). При подключении к удаленному ПК будет выполнено сравнение отправляемой версии скрипта с выполненной ранее. В случае несовпадения версий запустится выполнение скрипта. Версия будет записана в указанный файл при успешном выполнении. Если в настройках не задан путь к файлу **path\_exec\_script\_version** или в отправляемом скрипте отсутствует переменная с версией **scriptversion**, то значение данной секции будет 0.
   * **-us значение**) Имя секции настроек, которую необходимо использовать (Небязательный параметр. Если необходимо, чтобы запрос с выбором секции не показывался при запуске скрипта, например, при запуске скрипта с параметрами из cron задания или терминала, то необходимо задать имя используемой секции в файле настроек или задать его через этот параметр).

Примеры команд:

**./run-sssc.sh -us "tests-pas" -hf "localhost" -sp "sendmessage:sendmessage:autopassudo:0:0"**

**./run-sssc.sh -us "tests-pas" -hn "10.0.2.50" -hn "virtastra18-test3" -sp "sendmessage:sendmessage:autopassudo:0:0"**

**./run-sssc.sh -us "tests-key" -hn "10.0.2.93" -sp "rename-host::autopassudo:1:0" -sp "astra-ad-sssd::autopassudo:1:1" -sp "management-ca-cert:cert:autopassudo:0:1" -sp "cryptopro-install:cprocsp:autopassudo:0:1"**

## **Проверка наличия обновлений**

По умолчанию при запуске скрипта выполняется проверка обновлений в репозитории. Для отключения проверки обновлений укажите параметр **no\_check\_update\_script** со значением 1 в секции настроек.



