НАУЧНО-ИСПЫТАТЕЛЬНЫЙ ИНСТИТУТ
СИСТЕМ ОБЕСПЕЧЕНИЯ
КОМПЛЕКСНОЙ БЕЗОПАСНОСТИ

СИСТЕМА ЗАЩИТЫ КОРПОРАТИВНОЙ СОТОВОЙ СВЯЗИ «SAFEPHONE»

РУКОВОДСТВО ПО УСТАНОВКЕ И НАСТРОЙКЕ

4.1

Москва

2019

СОДЕРЖАНИЕ

Перечень используемых терминов и сокращений

Таблица 1 – Перечень терминов и сокращений

Сокращение	Полное наименование
AD	Служба каталогов (Active Directory)
ADEP	Программа управления корпоративными приложениями на устройствах Apple (Apple Developer Enterprise Program)
APNs	Служба отправки push-уведомлений на устройства Apple (Apple Push Notification Service)
CSR	Запрос на получение сертификата (Certificate Signing Request)
DNS	Система доменных имён (Domain Name System)
HTTPS	Расширение протокола HTTP для поддержки шифрования в целях повышения безопасности (HyperText Transfer Protocol Secure)
IP	Интернет-протокол (Internet Protocol)
MDM	Система управления мобильными устройствами (Mobile Device Management)
NTP	Протокол сетевого времени (Network Time Protocol)
TCP	Протокол управления передачей (Transmission Control Protocol)
SLES	Операционная система SUSE Linux Enterprise Server
UDP	Протокол пользовательских датаграмм (User Datagram Protocol)
VoIP	Технология передачи голоса поверх протокола IP (Voice Over Internet Protocol)
АРМ	Автоматизированное рабочее место
БД	База данных
ГИС	Географическая информационная система
МСК	Мобильное средство коммуникации (смартфон, планшетный компьютер)
ОС	Операционная система
ПО	Программное обеспечение
СУБД	Система управления базами данных

1 Введение

Настоящее руководство предназначено для установки системы защиты корпоративной сотовой связи «SafePhone» (далее по тексту – система «SafePhone» или Система) и содержит указания по установке и настройке программного окружения и серверных компонентов системы «SafePhone».

Система «SafePhone» состоит из следующих компонентов:

Клиентские компоненты:

- мобильный клиент SafePhone[1];
- АРМ Администратора SafePhone.

Серверные компоненты:

- сервер управления SafePhone (Socket server);
- сервер iOS MDM;
- сервер баз данных SafePhone;
- сервер веб-приложений SafePhone;
- сервер VoIP телефонии SafePhone (SIP-сервер);
- сервер ГИС.

2 Назначение и условия применения

2.1 Назначение системы «SafePhone»

Система «SafePhone» представляет из себя аппаратно-программный комплекс для защиты информации, передаваемой по каналам сотовой связи, от несанкционированного доступа.

2.2 Требования к программному обеспечению

Установка серверных компонентов системы «SafePhone» выполняется в предварительно созданной программной среде, включающей следующие элементы:

1. 64-х разрядные ОС: CentOS 7.4, Red Hat Enterprise Linux 7.4, SUSE Linux Enterprise Server 12 SP3;

[1] Мобильный клиент SafePhone состоит из клиентского ПО SafePhone и конфигурационного профиля управления МСК.

2. версии Oracle 11g Release 2 и Oracle 12c Release 2 (для БД Oracle);
3. версия PostgreSQL 9.6.X (для БД PostgreSQL);
4. настроенная синхронизация времени по протоколу NTP;
5. актуальные исправления Oracle, установленные на сервер баз данных SafePhone, касающиеся часовых поясов (time zone).

Для входа в АРМ Администратора требуется один из перечисленных браузеров:

1. Mozilla Firefox актуальной версии;
2. Google Chrome актуальной версии;
3. Microsoft Internet Explorer 11.

2.3 Требования к сетевому окружению

Для установки серверных компонентов системы «SafePhone» требуется следующее сетевое окружение:

1. разрешение подключений сервера iOS MDM к серверам Apple Push Notification Service посредством:
   - доступа к DNS-серверу, разрешающему доменные имена api.push.apple.com, gateway.push.apple.com, feedback.push.apple.com;
   - разрешения прохождения IP-трафика к адресам 17.0.0.0/8, порт 443/TCP, 53/TCP, 53/UDP;
2. разрешение внешних подключений по следующим портам[1] (указаны значения по умолчанию):
   - 443/TCP – для подключения МСК ОС iOS к серверу iOS MDM;
   - 50001/TCP – для подключения МСК, кроме МСК ОС iOS, к серверу управления SafePhone;
   - 8443/TCP, 8080/TCP – для подключения веб-консоли администратора к серверу веб-приложений SafePhone;
   - 5060/UDP, 5060/TCP – для передачи SIP сигнализации на сервер VoIP телефонии;
   - 6110/UDP, 6110/TCP – для передачи SIP сигнализации к МСК;

[1] После стандартной установки CentOS 6.9 и выше в системе включен брандмауэр и разрешены входящие соединения только на порт 22.

- 10 000 – 40 000/UDP – для передачи медиа-трафика VoIP телефонии;
- 123/UDP – для синхронизации с сервером времени.

3. разрешение сетевых подключений между серверными компонентами (указаны значения портов по умолчанию):
   - 1521/TCP – от сервера управления SafePhone, сервера iOS MDM, сервера веб-приложений SafePhone и сервера VoIP телефонии SafePhone в адрес сервера баз данных SafePhone (БД Oracle);
   - 5432/TCP – от сервера управления SafePhone, сервера iOS MDM, сервера веб-приложений SafePhone и сервера VoIP телефонии SafePhone в адрес сервера баз данных SafePhone (БД PostgreSQL);
   - 80/TCP – от рабочего места администратора SafePhone в адрес сервера ГИС.

2.4 Требования к серверу БД PostgreSQL на 1000 абонентов и полгода хранения данных

На хосте, которому предназначена роль сервера базы данных SafePhone, должно быть установлено ПО сервера баз данных PostgreSQL (в терминологии PostgreSQL − кластер PostgreSQL) версии 9.6.X.

Хранилище файлов баз данных SafePhone, должно иметь место для размещения файлов указанной БД объемом не менее 50 ГБ.

Для кластера PostgreSQL предусмотрены два варианта установки ПО БД SafePhone: ***стандартная установка***, посредством скрипта **install.sh,** и ***полная установка***, посредством скриптов **setup.sh** и **install.sh.**

Для проведения ***стандартной установки*** данный кластер PostgreSQL должен также обладать следующими свойствами, необходимыми для работы с SafePhone:

1. На сервере должна быть создана база данных с именем, соответствующим имени БД в файле db.yml (параметр **name**) при установке БД-компонента SafePhone. Кодировка этой БД должна быть UTF8.
2. На сервере должен быть создан пользователь с правами подключения к этой БД и на создание в ней временных таблиц (далее − пользователь SafePhone). Login-имя и пароль этого пользователя должны соответствовать указанным в файле db.yml параметрам **user** и **password**; должна быть

создана схема в этой БД, владельцем которой должен быть назначен пользователь SafePhone.

3. В указанной БД должны быть установлены следующие расширения функциональности:

3.1. В стандартной схеме public должно быть установлено расширение pgcrypto.

3.2. Должен быть установлен планировщик заданий PostgreSQL pgAgent, настроенный для работы под пользователем БД с ограниченными правами:

3.2.1. Должен быть создан пользователь ОС «pgAgent» и его домашний каталог.

3.2.2. Должен быть установлен и настроен демон pgAgent, функционирующий как сервис ОС Linux; сервис должен быть запущен и включен в автозапуск операционной системы.

3.2.3. В БД postgres должна быть установлена схема планировщика заданий: pgAgent.

3.2.4. Необходимо создать файл паролей для подключения агента к БД из п.1.

3.2.5. Должна быть создана роль scheduler для планировщика заданий и пользователь PostgreSQL с тем же именем; пароль этого пользователя должен быть равен тому же паролю, что использован при создании файла паролей (п. 3.2.4.); для простоты рекомендуется сделать его равным параметру **password** из файла db.yml

В связи с ролью scheduler должны быть проведены следующие изменения в настройках кластера PostgreSQL для БД postgres:

GRANT CONNECT ON DATABASE postgres, <ИМЯ БД п. 1.> TO scheduler;

GRANT USAGE ON SCHEMA pgagent TO scheduler;

GRANT USAGE ON ALL SEQUENCES IN SCHEMA pgagent TO scheduler;

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA pgagent TO scheduler;

GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA pgagent TO scheduler.

3.2.6. После этого должны быть изменены данные в файле паролей агента (п. 3.2.4.): данные БД-пользователя postgres должны быть заменены на данные пользователя scheduler.

3.2.7. Последним шагом конфигурирования планировщика заданий должна быть замена в сервис-файле pgAgent: DBUSER=postgres на DBUSER=scheduler

4. Пользователю scheduler должно быть дано право на соединение с БД пользователя SafePhone.

5. Переменная сессии SEARCH_PATH для роли пользователя БД SafePhone (п. 2.) должна содержать: <имя схемы БД SafePhone (п. 2.)>, public.

Для проведения ***полной установки*** предварительная подготовка БД не требуется, так как все требования к серверу БД реализуются запуском скрипта **setup.sh**.

2.5 Требования к серверу БД Oracle на 1000 абонентов и полгода хранения данных

1. Версии Oracle 11g Release 2 и Oracle 12c Release 2.
2. Для работы сервера Oracle с системой «SafePhone» требуется выделить отдельный экземпляр базы данных (database instance), который не должен использоваться для других целей.
3. В СУБД Oracle необходимо создать пользователя, от имени которого SafePhone работает в БД. Этот пользователь должен отвечать следующим требованиям:
   - имя пользователя (UserName) = SPHONE;
   - обладает правами CONNECT, RESOURCE, CREATE VIEW, EXECUTE ON DBMS_CRYPTO, CREATE JOB, MANAGE SCHEDULER.
4. Специфические параметры настройки сервера Oracle:
   - SID (Database name) = SPHONE;
   - табличное пространство для данных – SAFEPHONE_TS;
   - табличное пространство для индексов – SAFEPHONE_IDX;
   - размеры табличных пространств – SAFEPHONE_TS (16 ГБ), SAFEPHONE_IDX (6 ГБ);
   - временное табличное пространство – 4 ГБ;
   - максимальное количество сессий – 500;
   - максимальное количество процессов – 450;
   - максимальное количество транзакций – 550;
   - максимальное число открытых курсоров – 500.
5. При создании или эксплуатации БД Oracle для SafePhone нельзя использовать механизм сжатия: создавать сжатые таблицы, индексы или помещать их в табличные пространства со сжатием по умолчанию.
   На запросы: **select table_name, compression from user_tables;** или **select**

index_name, compression from user_indexes; в столбце **«COMPRESSION»** должно быть значение **«DISABLED»**.

Примечание – Под табличным пространством подразумевается суммарный объем табличного пространства. Его структура и состав файлов определяется ограничениями, накладываемыми СУБД Oracle, и находится в компетенции лица, администрирующего СУБД Oracle.

3 Установка и настройка программного обеспечения для серверных компонентов

Установка, настройка ПО и серверных компонентов осуществляется пользователем «root».

3.1 Получение цифровых сертификатов

Для создания запроса и генерации ключа используется криптографический пакет OpenSSL, при его отсутствии необходимо загрузить и установить OpenSSL с веб-сайта.

3.1.1 Сертификат SKB

При подключении МСК на платформе Android, Tizen и Windows к системе «SafePhone» через портал регистрации потребуется сертификат с приватным ключом для удостоверения инсталляционных SKB-файлов. Сертификат предоставляет НИИ СОКБ по запросу от заказчика.

Для получения сертификата SKB необходимо выполнить следующие действия:

1. Для запуска процесса генерации приватного ключа и формирования запроса в формате CSR выполнить команду, где **XXX** – краткое название организации-заказчика:

```
openssl req -out XXX.csr -new -newkey rsa:2048 -nodes -keyout skb.key
```

2. Заполнить форму запроса, приведенную в приложении А, указав ***skb*** в следующей строке:

```
Common Name (eg, your name or your server's hostname)
[]:skb
```

3. По окончанию генерации ключа и запроса на сертификат будут сформированы два файла:
    - XXX.csr− запрос на сертификат;
    - skb.key − сгенерированный ключ.
4. Файл XXX.csr следует передать в службу технической поддержки системы «SafePhone» по электронной почте для получения сертификата в виде файла skb.chain.
5. Полученные файлы skb.chain и skb.key и поместить в конфигурационную папку /opt/safephone/config/ при условии, что SafePhone будет установлен в /opt/safephone/, совместно с файлом docker-compose.yml.
6. Полученный сертификат выдается на один год и по истечении срока должен быть продлен. Для этого следует сформировать новый запрос с использованием старого ключа следующей командой:

```
openssl req -new -key skb.key -out XXX.csr
```

7. Затем обратиться в службу технической поддержки системы «SafePhone» в соответствии с регламентом, изложенным в этом подразделе.

3.1.2 Сертификат HTTPS

Для активации возможности работы протокола HTTPS, корректной работы портала регистрации и сервера веб-приложений SafePhone потребуются сертификаты и ключи HTTPS:

- iosmdm.crt − сертификат сервера iOS MDM;
- iosmdm.key − приватный ключ сервера iOS MDM;
- arm.crt − сертификат сервера веб-приложений SafePhone;
- arm.key − приватный ключ сервера веб-приложений SafePhone.

Генерация приватных ключей с формированием долгосрочных самоподписанных сертификатов выполняется при запуске скрипта

первоначальной настройки в соответствии с описанием в перечислении 7 раздела 4.

Проверить, что сертификаты и ключи автоматически помещены в конфигурационную папку, а именно:

iosmdm.crt и iosmdm.key в /opt/safephone/config/;

arm.crt и arm.key в /opt/safephone/config/nginx/.

Если серверные компоненты, которым требуются HTTPS сертификаты и ключи расположены на разных серверах, следует в соответствии с описанием в перечислении 7 раздела 4, сгенерировать приватные ключи и сформировать сертификаты, а затем переместить их на целевые серверы в указанные папки.

Для исключения отображения ошибок, корректной работы сервера iOS MDM и сервера веб-приложений SafePhone, следует получить HTTPS-сертификаты в организации, которая обладает правом выдачи цифровых сертификатов. Для этого необходимо выполнить следующие действия:

1. Сгенерировать ключи и сформировать запросы на выпуск сертификатов в формате CSR следующей командой:

   для сервера iOS MDM

   ```
   openssl req -out iosmdm.csr -new -newkey rsa:2048 -nodes -keyout iosmdm.key
   ```

   для сервера веб-приложений SafePhone

   ```
   openssl req -out arm.csr -new -newkey rsa:2048 -nodes -keyout arm.key
   ```

8. Заполнить форму запроса, приведенную в приложении А, указав ***iosmdm*** или ***arm*** в следующей строке:

   ```
   Common Name (eg, your name or your server's hostname) []:iosmdm
   ```

2. Направить запрос в организацию (центр сертификации). После проверки

данных, указанных в запросе будет выписан сертификат со следующей информацией:

- полное (уникальное) имя владельца сертификата;
- открытый ключ владельца;
- дата выдачи HTTPS-сертификата;
- дата окончания сертификата;
- полное (уникальное) имя центра сертификации;
- цифровая подпись издателя.

3. Полученные сертификаты и ключи поместить в конфигурационную папку как было описано в данном подразделе. Сервер автоматически сопоставит выпущенный сертификат со сгенерированным приватным ключом, вследствие чего будет осуществлять зашифрованное и безопасное соединение.

4. Полученные сертификаты имеют ограниченный срок действия, который зависит от вида самого сертификата и организации (центра сертификации).

5. При необходимости получения HTTPS-сертификата для других серверов следует повторить действия в соответствии с указанным регламентом.

6. При использовании сертификата *корпоративного* удостоверяющего центра в файл **«iosmdm.crt»** сертификата сервера iOS MDM необходимо занести всю цепочку сертификатов следующим образом:

```
-----BEGIN CERTIFICATE-----
сертификат сервера
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
промежуточный сертификат
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
корневой сертификат
-----END CERTIFICATE-----
```

3.1.3 Сертификат Push MDM

Для возможности отправки пользователю МСК на платформе iOS MDM push-уведомлений потребуется сертификат и ключ APNs для сервера iOS MDM.

Для ***получения*** сертификата Push MDM необходимо выполнить следующие действия:

1. Для запуска процесса генерации приватного ключа и формирования запроса на сертификат в формате CSR выполнить команду:

```
openssl req -out MdmPush.csr -new -newkey rsa:2048 -nodes -keyout MdmPush.key
```

2. Заполнить форму запроса, приведенную в приложении А, указав ***MdmPush*** в следующей строке:

```
Common Name (eg, your name or your server's hostname) []:MdmPush
```

3. По окончанию генерации ключа и запроса на сертификат будут сформированы два файла:
   - MdmPush.csr− запрос на сертификат;
   - MdmPush.key − сгенерированный ключ.
4. Файл MdmPush.csr следует отправить/передать в службу технической поддержки системы «SafePhone» по электронной почте. Подписанный файл CSR от НИИ СОКБ будет возвращён в формате PLIST.
5. В браузере перейти на страницу https://identity.apple.com/pushcert/ и зайти на портал регистрации сертификатов для push-уведомлений (Apple Push Certificates Portal) посредством своей учетной записи (Apple ID).

   Примечание − Рекомендуется отдельная учетная запись для должности администратора (не персональная) с целью сохранения возможности управления корпоративными сертификатами при увольнении ответственного сотрудника.

6. На портале регистрации сертификатов для push-уведомлений следует выполнить следующие действия:
   - нажать **«Create a Certificate Upload»** (Создать сертификат);
   - ознакомиться и согласиться с предложенными условиями **«I have read and agree to these terms and conditions»,** нажав на **«Accept»** (Принять);
   - нажать **«Choose File»** (Выбрать файл), перейти на подписанный CSR на своем компьютере и нажать **«Upload»** (Загрузить);
   - для получения файла сертификата в формате PEM нажать **«Download»** (Скачать) и скачать файл с названием push.pem.
7. Файлы MdmPush.key и push.pem поместить в конфигурационную папку /opt/safephone/config/, при условии, что SafePhone будет установлен в /opt/safephone/, совместно с файлом docker-compose.yml.
8. Объединить файлы сертификата и приватного ключа в один файл MdmPush.pem:

```
cat push.pem MdmPush.key > MdmPush.pem
```

9. В дальнейшем используется только файл MdmPush.pem, поэтому остальные использованные файлы следует удалить для обеспечения безошибочной работы:

```
rm -f push.pem MdmPush.key
```

10. Полученный сертификат выдается на один год и должен быть своевременно обновлен в соответствии с регламентом, изложенным в этом подразделе.

Для ***обновления*** сертификата Push MDM необходимо выполнить следующие действия:

1. Сформировать новый запрос с использованием старого ключа следующей командой:

```
openssl req -new -key MdmPush.key -out MdmPush.csr
```

2. Заполнить форму запроса, приведенную в приложении А, указав ***MdmPush*** в

следующей строке:

```
Common Name (eg, your name or your server's hostname)
[]:MdmPush
```

3. По окончанию генерации запроса на сертификат, сформированный файл MdmPush.csr следует отправить/передать в службу технической поддержки системы «SafePhone» по электронной почте. Подписанный файл CSR от НИИ СОКБ будет возвращён в формате PLIST.

4. В браузере перейти на страницу https://identity.apple.com/pushcert/ и зайти на портал регистрации сертификатов для push-уведомлений (Apple Push Certificates Portal) посредством своей учетной записи (Apple ID/Password).

5. На портале регистрации сертификатов для push-уведомлений следует выполнить следующие действия:

   – выбрать строку с сертификатом, подлежащим обновлению, и нажать **«Renew»** (Обновить);

   Примечание – При обновлении сертификата не следует нажимать «Download» (Скачать) или «Revoke» (Отозвать), т.к. оба эти параметра потребуют повторной регистрации всех МСК на платформе iOS.

   – нажать **«Choose File»** (Выбрать файл), перейти на подписанный CSR на своем компьютере и нажать **«Upload»** (Загрузить);

   – для получения файла сертификата в формате PEM нажать **«Download»** (Скачать).

6. В конфигурационной папке /opt/safephone/config/ открыть файл MdmPush.pem и скопировать в него строки из обновленного сертификата, заменив информацию об истекшем сертификате, информацию о приватном ключе оставить без изменений. Сохранить внесенные изменения.
   Пример файла MdmPush.pem приведен ниже:

```
-----BEGIN CERTIFICATE-----
вставить содержимое обновленного сертификата
```

```
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
оставить без изменений
-----END RSA PRIVATE KEY-----
```

7. Перезапустить docker-контейнеры для сервера iOS MDM следующей командой:

```
docker-compose restart iosmdm iosmdm-bg
```

8. При необходимости отзыва сертификата Push MDM следует на портале регистрации в строке с выбранным сертификатом нажать **«Revoke»** (Отозвать).

3.1.4 Сертификат Push Monitor

Для обеспечения возможности установки клиентского ПО SafePhone на МСК на платформе iOS, без использования учётных записей Apple Store, и отправки push-уведомлений потребуется сертификат и ключ Push Monitor.

Для получения сертификата Push Monitor необходимо выполнить следующие действия:

1. Для запуска процесса генерации приватного ключа и формирования запроса на сертификат в формате CSR необходимо выполнить команду:

```
openssl req -out MonitorPush.csr -new -newkey rsa:2048
-nodes -keyout MonitorPush.key
```

2. Заполнить форму запроса, приведенную в приложении А, указав ***MonitorPush*** в следующей строке:

```
Common Name (eg, your name or your server's hostname)
[]:MonitorPush
```

3. По окончанию генерации ключа и запроса на сертификат будут сформированы два файла:

   - MonitorPush.csr− запрос на сертификат;
   - MonitorPush.key − сгенерированный ключ.

4. Затем зарегистрироваться и активировать в браузере на странице https://developer.apple.com/programs/enterprise/ программу ADEP, дающую доступ к **Apple Developer Center**.

5. Включить учётную запись разработчика ПО SafePhone (Apple ID) в свою программу ADEP для назначения идентификатора приложения (App ID) клиентскому приложению SafePhone.

6. В **Apple Developer Center** перейти в раздел **«Certificates, Identifiers & Profiles»**, нажать «+» для создания сертификата и выбрать тип **«Certificates» / «Development» / «Apple Push Notification service SSL (Sandbox & Production)**.

7. Выбрать в раскрывающемся списке **App ID** клиентское приложение SafePhone и нажать **«Continue».**

8. Затем загрузить выбранный файл запроса на сертификат (CSR) на своем компьютере.

9. Для получения файла сертификата APS.CER нажать **«Download»** (Скачать).

10. Файлы MonitorPush.key и aps.cer поместить в конфигурационную папку /opt/safephone/config/, при условии, что SafePhone будет установлен в /opt/safephone/, совместно с файлом docker-compose.yml.

11. Выполнить команду конвертации файла сертификата в формат PEM:

```
openssl x509 -in aps.cer -inform DER -out aps.pem
```

12. Объединить файлы сертификата и приватного ключа в один файл MonitorPush.pem:

```
cat aps.pem MonitorPush.key > MonitorPush.pem
```

13. В дальнейшем используется только файл MonitorPush.pem, поэтому остальные использованные файлы следует удалить для обеспечения безошибочной работы:

```
rm -f aps.cer aps.pem MonitorPush.key
```

14. Полученный сертификат выдается на один год и должен быть своевременно обновлен в соответствии с регламентом, изложенным в этом подразделе.

3.2 Установка и настройка БД PostgreSQL

3.2.1 Установка и настройка сервера PostgreSQL для CentOS

Для установки БД PostgreSQL для CentOS требуется выполнить следующие действия:

1. Добавить официальный репозиторий PostgreSQL:

```
yum -y install
https://download.postgresql.org/pub/repos/yum/9.6/redha
t/rhel-7-x86_64/pgdg-centos96-9.6-3.noarch.rpm
```

2. Выполнить установку сервера PostgreSQL CentOS 7 версии 9.6:

```
yum -y install postgresql96-server postgresql96-contrib
pgagent_96
```

3. Инициализировать необходимую БД:

```
/usr/pgsql-9.6/bin/postgresql96-setup initdb
```

4. Посредством редактора **vi** задать адреса TCP/IP, по которым сервер будет принимать подключения клиентских приложений. Значение **«*»** параметра **«listen_addresses»** (прослушиваемые адреса) обозначает, что заданы все имеющиеся IP-интерфейсы:

```
vi /var/lib/pgsql/9.6/data/postgresql.conf
        listen_addresses = '*'
```

5. Разрешить подключения из удалённых хостов с авторизацией по паролю, добавив следующую строчку в редакторе **vi**:

```
vi /var/lib/pgsql/9.6/data/pg_hba.conf
        host    all             all             0.0.0.0
/0                md5
```

6. Настроить запуск сервера PostgreSQL со стартом системы и запустить:

```
systemctl enable postgresql-9.6
systemctl start postgresql-9.6
```

7. Посредством редактора **vi** следует отредактировать конфигурацию сервиса pgAgent, в которой указать работу через ограниченного пользователя **«scheduler»**:

```
vi /etc/pgagent/pgagent_96.conf
        DBUSER=scheduler
```

8. Сохранить пароль пользователя планировщика pgAgent:

```
mkhomedir_helper pgagent
echo *:*:*:*:111 > /home/pgagent/.pgpass
chmod 600 /home/pgagent/.pgpass
chown pgagent /home/pgagent/.pgpass
```

9. Настроить запуск сервиса pgAgent со стартом системы и запустить:

```
systemctl enable pgagent_96
systemctl start pgagent_96
```

3.2.2 Установка и настройка сервера PostgreSQL для SLES

1. Зарегистрировать ПО SLES 12 SP3:

```
SUSEConnect -r $RegCode    # RegCode – код регистрации
подписки на продукт
```

2. Активировать ПО, указав модули в следующем формате:

```
SUSEConnect -p sle-sdk/12.3/x86_64
SUSEConnect -p sle-module-containers/12/x86_64
SUSEConnect -p sle-module-legacy/12/x86_64
```

3. Выполнить установку сервера PostgreSQL версии 9.6:

```
zypper install -y postgresql96-server postgresql96-
```

```
contrib
```

4. Установить планировщик заданий pgAgent 9.6:

```
zypper -n --no-gpg-checks install
http://ftp.gwdg.de/pub/opensuse/distribution/leap/42.3/
repo/oss/suse/x86_64/libwx_baseu-2_8-0-compat-lib-stl-
2.8.12-32.27.x86_64.rpm
zypper -n --no-gpg-checks install
https://download.postgresql.org/pub/repos/zypp/9.6/suse
/sles-12-x86_64/pgagent_96-3.4.0-10.sles12.x86_64.rpm
```

5. Инициализировать необходимую БД:

```
sudo -u postgres initdb -D /var/lib/pgsql/data
```

6. Посредством редактора **vi** задать адреса TCP/IP, по которым сервер будет принимать подключения клиентских приложений. Значение **«*»** параметра **«listen_addresses»** (прослушиваемые адреса) обозначает, что заданы все имеющиеся IP-интерфейсы:

```
vi /var/lib/pgsql/data/postgresql.conf
    listen_addresses = '*'
```

7. Разрешить подключения из удалённых хостов с авторизацией по паролю, добавив следующую строчку в редакторе **vi**:

```
vi /var/lib/pgsql/data/pg_hba.conf
    host    all             all             0.0.0.0/0
                md5
```

8. Посредством редактора **vi** следует отредактировать конфигурацию сервиса pgAgent, в которой указать работу через ограниченного пользователя **«scheduler»**:

```
vi /etc/pgagent/pgagent_96.conf
    DBUSER=scheduler
```

9. Сохранить пароль пользователя планировщика pgAgent:

```
mkhomedir_helper pgagent
echo *:*:*:*:111 > /home/pgagent/.pgpass
chmod 600 /home/pgagent/.pgpass
chown pgagent:pgagent /home/pgagent/.pgpass
```

10. Для корректной работы сервиса pgAgent скопироватьо необходимые расширения:

```
cp -p /usr/pgsql-9.6/share/extension/*
/usr/share/postgresql96/extension
```

11. Настроить запуск сервера PostgreSQL со стартом системы и запустить:

```
systemctl enable postgresql
systemctl start postgresql
```

12. Настроить запуск сервиса pgAgent со стартом системы и запустить:

```
systemctl enable pgagent_96
systemctl start pgagent_96
```

3.3 Установка Docker

Для установки Docker требуется выполнить следующие действия:

1. Осуществить установку и настройку актуальных пакетов, о наличии которых нужно узнать на сайте производителя Docker:

```
yum install -y yum-utils
yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
yum makecache fast
yum -y install docker-ce
```

2. Настроить запуск Docker со стартом системы:

```
systemctl enable docker
```

3. Запустить Docker с помощью команды:

```
systemctl start docker
```

4. Выполнить установку компонентов docker-compose (указанная ниже команда справедлива для версии 1.23.1):

```
curl -L
https://github.com/docker/compose/releases/download/1.2
3.1/docker-compose-`uname -s`-`uname -m` >
/usr/local/bin/docker-compose
```

5. Настроить права доступа к установленным компонентам:

```
chmod +x /usr/local/bin/docker-compose
```

6. Осуществить проверку версии установленных компонентов:

```
docker-compose --version
```

7. Для ограничения параметров логирования выполнить команду создания файла daemon.json с редактированием, посредством редактора **vi**:

```
vi /etc/docker/daemon.json
```

Установить максимальные значения в файле daemon.json:

```
{
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m",
    "max-file": "10"
  }
}
```

4 Установка и настройка серверных компонентов системы «SafePhone»

Комплект ПО для установки системы «SafePhone» состоит из следующих файлов:

- safephone-config.tar.gz;
- safephone-docker.tar.gz;
- db-postgresql-4.1.tar.gz (для БД PostgreSQL);

- db-oracle-4.1.tar.gz (для БД Oracle).

Для установки серверных компонентов следует выполнить следующие операции:

1 Установить docker-образы серверных компонентов из архива **safephone-docker.tar.gz** посредством команды:

```
docker load -i safephone-docker.tar.gz
```

2 Создать папку **«safephone»** и распаковать в нее архив с конфигурацией компонентов SafePhone в Docker (файл **safephone-config.tar.gz**) с помощью команды:

```
mkdir /opt/safephone
```

```
tar -xzvf safephone-config.tar.gz -C /opt/safephone
```

3 В директории **«safephone»** создать папку **«db»** и распаковать в нее один из выбранных архивов (файл **db-postgresql-4.1.tar.gz** − для БД PostgreSQL, файл **db-oracle-4.1.tar.gz** − для БД Oracle).
Например, для БД PostgreSQL команды будут следующими:

```
mkdir /opt/safephone/db
```

```
tar -xzvf db-postgresql-4.1.tar.gz -C /opt/safephone/db
```

4 Установить схему БД. Для этого необходимо запустить скрипт, который находится в папке **«db»**.

Требования к серверу БД при ***стандартной*** и ***полной установке*** описаны в 2.4.

Стандартная установка

***Для БД PostgreSQL**:*

При запуске скрипта следующей командой:

```
./install.sh
```

будет предложена установка схемы БД с параметрами по умолчанию, а именно:

```
File: ./sql/schema.sql   # Название файла для установки
Database: sphone   # Имя базы данных
User: sphone   # Имя пользователя
Scheduler: scheduler   # Имя пользователя планировщика
Schema: sphone   # Название схемы
Continuie (y/n)?
```

Для продолжения работы с предложенными параметрами, следует написать **«y»**. В противном случае написать **«n»** и запустить скрипт с указанием требуемых параметров. Если какой-то из параметров схемы не будет указан, следовательно, в схему будет включен параметр по умолчанию:

```
./install.sh [--user username] [--db dbname] [--scheduler scheduler] [--schema schema] -- [--host=192.168.XX.XX]  # host=192.168.XX.XX - адрес сервера БД
```

Для получения справки по параметрам схемы требуется запустить скрипт с ключом **-h** или **--help.**

Пример команды приведен ниже:

```
./install.sh -- --host=192.168.0.1
```

Затем требуется подтвердить установку схемы БД вводом пароля пользователя.

***Для БД Oracle**:*

При запуске скрипта ***install.sh*** необходимо указать в явном виде схему: пользователя, пароль, IP сервера, имя БД.

Пример команды приведен ниже:

```
./install.sh -- sphone/111@127.0.0.1/sphone
```

Далее после утвердительного ответа **«y»** на вопрос **«Continue (y/n)?»** будет установлена схема БД.

Полная установка для БД PostgreSQL:

Затем осуществить подготовку БД посредством скрипта:

```
./setup.sh
```

Далее запустить скрипт ***install.sh***. Последующие действия будут аналогичными действиям при запуске скрипта ***install.sh*** в стандартной установке.

5 В распакованном архиве конфигурации в папке **«safephone / default / config»** содержатся минимальные конфигурационные файлы компонентов SafePhone.

- Пример конфигурационного файла «**iosmdm.yml**» приведен ниже:

```
iosmdm:
  addr: 192.168.XX.XX    # Адрес или имя сервера
  monitoruid: ru.niisokb.Monitor    # App ID клиентского приложения SafePhone
  defaultunit: Компания    # Подразделение, в котором создаются сотрудники при импорте из AD
  defaultposition: Сотрудник    # Должность, назначаемая сотрудникам, у которых она не указана в AD
  providers:    # Параметр, определяющий способ аутентификации (Код приглашения, AD или один из выбранных способов)
    - code
    - ldap
  ldap:
    addr: 192.168.XX.XX    # Адрес или имя сервера AD
    basedn: dc=company,dc=ru    # Базовый DN (Distinguished name) для поиска объектов
    timeout: 600    # Время блокировки после истечения попыток ввода
    maxfails: 3    # Количество неудачных попыток, после которых временно блокируется аутентификация пользователя
```

```
    group: ""   # Имя группы в формате DN
  log: W   # Уровень логирования. Логируются сообщения, с уровнем равным или выше указанного. Используются следующие уровни логирования в убывающем порядке степени важности: F(fault), E(error), W(warning), I(info), T(trace)
  push:
    interval: 180   # Период опроса устройства
    cleanup: 31104000   # Время неактивности, после которого удаляются данные, необходимые для связи с устройством
  timeouts:
    offline: 900   # Время неактивности, после которого устройство считается отключенным
    longoffline: 259200   # Время неактивности, после которого устройство считается отключенным длительное время
  mdm_cert: /etc/safephone/iosmdm/iosmdm.crt # Адрес сертификата сервера iOS MDM в docker-контейнере
  mdm_key: /etc/safephone/iosmdm/iosmdm.key # Адрес приватного ключа сервера iOS MDM в docker-контейнере
  skb:
    mode: cuks.ini # Режим генерации skb, в Системе поддерживается cuks.ini
    build_signature_files:
      key: /etc/safephone/iosmdm/skb.key # Адрес приватного ключа SKB в docker-контейнере
      chain: /etc/safephone/iosmdm/skb.chain # Адрес сертификата SKB формата chain в docker-контейнере
    ini:
      ss_cert: /etc/safephone/iosmdm/nginx/ss.crt # Адрес сертификата сервера управления SafePhone (SS) в docker-контейнере
```

```
  loader: # Параметры публикации на портале регистрации
приложения «Loader», предназначенного для установки
мобильного клиента на МСК Samsung
    - descr: Samsung mobile type SM #16883
      regex: \bsm-
      url: https://safephone.store/android/loader.apk
    - descr: Samsung mobile type
      regex: \bsamsung
      url: https://safephone.store/android/loader.apk
    - descr: Samsung mobile type GT
      regex: \bgt-
      url: https://safephone.store/android/loader.apk
  # external_log_system: # Параметры подключения к
системе централизованного мониторинга логов
  #   host: 192.168.1.3
  #   port: 33128
  #   root_url: /
  loggers:
    # database
    - name: database, database.caller
      level: WARNING
  lost_mode_messages: # Сообщение, отображаемое на МСК
iOS при блокировке устройства
    message: Устройство заблокировано
    footnote: Обратитесь к администратору


params: # Параметры ключа контейнера Knox для
приложения «Loader»
  knox_key:
"34AA6EFC5855F6E95117838285E73C42BCDCAB0DD37179A0B5B00B
E4B590D63AF0A7DF2AA066DC0F3DA9C01E9E296D1594390DC03E376
9969F37117E2E95C70B"
```

- Пример конфигурационного файла «**db.yml**» приведен ниже:

```
db:
```

```
  type: postgresql   # Тип сервера баз данных (oracle / postgresql)
  user:   # Пользователь БД
  password:   # Пароль пользователя БД
  host: 127.0.0.1   # Адрес сервера БД
  port: 5432   # Порт сервера БД (1521 / 5432)
  name: sphone    # Имя сервера БД
```

- Пример конфигурационного файла «**arm.yml**» приведен ниже:

```
server:
  servlet:
    context-path:   # Путь контекста приложения. Если server.context-path=/safephone, то адрес в строке браузера подобен https://localhost:8443/safephone/
    session-timeout:  30
  use-forward-headers: true   # Включение режима APM для работы с nginx

#Logging
logging:
  level: # Уровни логирования (TRACE, DEBUG, INFO, WARNING, ERROR, FATAL, OFF)
    root: WARNING
    jdbc:
      sqlonly: OFF   # Логирование SQL-запросов
      audit: OFF   # Логирует практически всё для запросов

      connection: OFF   # Логирование открытие/закрытие соединения с БД
      sqltiming: OFF   # Логирование SQL-запросов с указанием времени в миллисекундах, затраченного на выполнение запроса
    com.safephone.dao.resultcode: OFF   # Логирование
```

```
переменной O_RC

session.control:   # Управление пользовательскими
сессиями
  enabled: false   # Включение управления
пользовательскими сессиями, по умолчанию -
false(отключено)
  maximum-sessions: 2   # Максимальное количество
одновременных сессий для одного пользователя, по
умолчанию - 2
  max-session-prevents-login: false   # Политика при
превышении максимального числа конкурентных сессий
одного пользователя. По умолчанию - false. false -
принудительно закрываем одну старую сессию, true -
запрещаем новую сессию

## <ldap>
ad:   # Настройки интеграции с Active Directory
  domain: company.ru   # Имя домена в Active Directory
  url: ldap://192.168.XX.XX   # URL службы Active
Directory
#  search-filter:   # Строка для фильтрации аккаунтов в
Active Directory, по умолчанию -
(&(objectClass=user)(userPrincipalName={0})
(memberOf=CN=GroupName,OU=Groups,OU=root,DC=company,DC=
ru)), т.е. все зарегистрированные в AD пользователи,
входящие в группу
## </ldap>

auth-provider:
  database.on:  true   # Включение аутентификации через
БД, по умолчанию - true(включено)
  active-directory.on:  false   # Включение
аутентификации через Active Directory, по умолчанию -
false(отключено)
```

```
gis:
  servers:   # Список источников карт
    - name: openstreetmap
      label: openstreetmap.org
      url: http://{a-
c}.tile.openstreetmap.org/{z}/{x}/{y}.png
```

- Пример конфигурационного файла «**ss.yml**» приведен ниже:

```
ss:
  unloader:
    count: 50   # Число потоков Unloader. Каждый
Unloader имеет connect к SQL database
  loader:
    task-limit: 200   # Максимальное количество команд
для устройств, загружаемое из БД
    acknowledge-timeout: 60   # Таймаут ожидания ответа
от устройства в секундах
    acknowledge-long-timeout: 180   Таймаут ожидания
ответа от устройства в секундах для долгих команд
    max-blob-size: 1048576   # Максимальный размер
передаваемого тела пакета (0 - без ограничения)
    ping-timeout: 60   # Таймаут активности клиента для
отправки серверного пинга в секундах (0 - не
использовать пинги)
  cmdproc:
count: 10   # Число потоков CommandProcessor. Каждый
поток имеет подключение к серверу БД
  client:
```

```
    recv-timeout: 60    # Время ожидания чтения из сокета в секундах
  tcp:
    keepalive:
      enabled: 1    # Использовать TCP KEEP ALIVE, по умолчанию - включено
      time: 600    # Период простоя TCP-соединения в секундах, после которого начинается процедура KEEP ALIVE в секундах
      interval: 2 # Временная задержка между попытками в секундах
      iprobes: 2    # Число попыток проверить, что соединение еще "живое"
  db:
    reconnect-timeout: 60    # Таймаут между попытками переподключения к базе данных в секундах
    janitor:
      period: 30    # Период проверки соединения сокет-сервера с БД в секундах
      hang: 20    # Таймаут между проверками соединения сокет-сервера с БД в секундах
  logger:
    level: W    # Уровень логирования. Логируются сообщения, с уровнем равным или выше указанного. Используются следующие уровни логирования в убывающем порядке степени важности: F(fault), E(error), W(warning), I(info), T(trace)
    dump:
      enabled: 1    # Включение-отключение дампа сообщений, по умолчанию - включено. В этом случае
```

```
производится логирование сообщений с данными в шестнадцатеричном формате.
      limit: 1024    # Ограничение размера дампа сообщений в байтах (1024). Ограничение действует для всех дампов
  api:
    url: https://example.com/api/    # Адрес или доменное имя сервера iOS MDM
    certificate: /config/iosmdm.crt    # Адрес сертификата iOS MDM
```

- Пример конфигурационного файла «**sipsync.yml**» приведен ниже:

```
sipsync:
  asterisk: tcp://sipsync:sipsync@127.0.0.1:5038    # Адрес сокета управления астериском (AMI)
  peers_conf: /var/lib/asterisk/sip_peers.conf    # Путь к конфигурационному файлу со списком абонентов (не должен меняться)
  interval: 120    # Интервал опроса БД
  number:    # Настройки генерации номеров
    prefix: ''
    begin: 1000
    end: 9999999
    fill: 0
```

6 В папке **«safephone / default»** проверить наличие файла **«docker-compose.yml»**.

Пример файла для БД PostgreSQL приведен ниже, указанные в нем параметры могут отличаться в зависимости от конфигурации Системы:

```
version: '3'
services:

  ## <nginx>
```

```
  nginx:
    image: "docker-registry.niisokb.ru/safephone/nginx:${NGINX_VERSION}"
    restart: always
    labels:
      safephone: "nginx"
    depends_on:
      - arm            ## arm
      - iosmdm         ## iosmdm
      - socket-server ## socket-server
    ports:
      - ${BIND_ADDR}:443:443      ## iosmdm
      - ${BIND_ADDR}:80:80        ## iosmdm
      - ${BIND_ADDR}:8443:8443    ## arm-https
      - ${BIND_ADDR}:8080:8080    ## arm-http
      - ${BIND_ADDR}:8081:8081    ## arm
      - ${BIND_ADDR}:50001:50001 ## socket-server
    volumes:
      - ./config:/config:ro
      - ./config/nginx:/etc/nginx/conf.d:ro
      - iosmdm_cache:/run/safephone/iosmdm/cache:ro  ## iosmdm
      - arm-static:/usr/share/arm:ro ## arm
    tmpfs:
      - /var/cache/nginx
    networks:
      - backend
    logging:
      driver: "json-file"
      options:
```

```
        max-size: "100m"
        max-file: "10"
  ## </nginx>

  ## <asterisk>
  asterisk:
    image: "docker-registry.niisokb.ru/safephone/asterisk:${ASTERISK_VERSION}"
    restart: always
    network_mode: host
    labels:
      safephone: "asterisk"
    volumes:
      - ./config/asterisk:/etc/asterisk:ro
      - asterisk_data:/var/lib/asterisk
    logging:
      driver: "json-file"
      options:
        max-size: "100m"
        max-file: "10"
  ## </asterisk>

  ## <arm>
  arm:
    image: "docker-registry.niisokb.ru/safephone/arm:${ARM_VERSION}"
    restart: always
    labels:
      safephone: "arm"
    depends_on:  ## postgres
```

```
      - postgres ## postgres
    environment:
      - java_heap_size_min=256
      - java_heap_size_max=256
    volumes:
      - ./config:/config:ro
      - arm-static:/static
    tmpfs:
      - /tmp
    networks:
      - backend
    logging:
      driver: "json-file"
      options:
        max-size: "100m"
        max-file: "10"
  ## </arm>

  ## <iosmdm>
  iosmdm-bg:
    image: "docker-
registry.niisokb.ru/safephone/iosmdm:${MDM_SERVER_VERSI
ON}"
    restart: always
    labels:
      safephone: "iosmdm-bg"
    depends_on:  ## postgres
      - postgres ## postgres
    volumes:
      - ./config:/etc/safephone/iosmdm:ro
```

```
      - iosmdm_data:/var/safephone/iosmdm
      - iosmdm_cache:/run/safephone/iosmdm
    command: iosmdm-bg-runner
    networks:
      - backend
    logging:
      driver: "json-file"
      options:
        max-size: "100m"
        max-file: "10"
  ## </iosmdm>

  ## <iosmdm>
  iosmdm:
    image: "docker-
registry.niisokb.ru/safephone/iosmdm:${MDM_SERVER_VERSI
ON}"
    restart: always
    labels:
      safephone: "iosmdm"
    depends_on:  ## postgres
      - postgres ## postgres
    volumes:
      - ./config:/etc/safephone/iosmdm:ro
      - iosmdm_data:/var/safephone/iosmdm
      - iosmdm_cache:/run/safephone/iosmdm
    command: iosmdm-main
    networks:
      - backend
    logging:
```

```
      driver: "json-file"
      options:
        max-size: "100m"
        max-file: "10"
  ## </iosmdm>

  ## <socket-server>
  socket-server:
    image: "docker-
registry.niisokb.ru/safephone/socket-
server:${SOCKET_SERVER_VERSION}"
    restart: always
    labels:
      safephone: "socket-server"
    depends_on:  ## postgres
      - postgres ## postgres
    volumes:
      - ./config:/config:ro
    networks:
      - backend
    logging:
      driver: "json-file"
      options:
        max-size: "100m"
        max-file: "10"
  ## </socket-server>

  ## <asterisk>
  sipsync:
```

```
    image: "docker-registry.niisokb.ru/safephone/sipsync:${SIP_SYNC_VERSION}"
    restart: always
    networks:
      - backend
    depends_on:  ## postgres
      - postgres ## postgres
    labels:
      safephone: "sipsync"
    volumes:
      - ./config:/config:ro
      - asterisk_data:/var/lib/asterisk
    logging:
      driver: "json-file"
      options:
        max-size: "100m"
        max-file: "10"
  ## </asterisk>

  ## <postgres>
  postgres:
    image: "docker-registry.niisokb.ru/safephone/postgres:${POSTGRES_VERSION}"
    restart: always
    networks:
      - backend
    volumes:
      - pgdata:/var/lib/postgresql/data:rw
    labels:
```

```
      safephone: "postgres"
    logging:
      driver: "json-file"
      options:
        max-size: "100m"
        max-file: "10"

  pgagent:
    image: "docker-
registry.niisokb.ru/safephone/postgres:${POSTGRES_VERSI
ON}"
    restart: always
    entrypoint: /usr/bin/pgagent
    command: -f host=postgres dbname=postgres
user=scheduler
    volumes:
      - pgdata:/var/lib/postgresql/data:ro
    depends_on:
      - postgres
    networks:
      - backend
    labels:
      safephone: "pgagent"
    logging:
      driver: "json-file"
      options:
        max-size: "100m"
        max-file: "10"
  ## </postgres>

networks:
```

```
  backend: ## backend


volumes:
  arm-static:      ## arm
  iosmdm_data:     ## iosmdm
  iosmdm_cache:    ## iosmdm
  pgdata:          ## postgres
  asterisk_data:   ## asterisk
```

7 Для работы скрипта первоначальной настройки необходимо ПО Git. При отсутствии указанного продукта на сервере следует выполнить команду:

```
yum install git
```

8 Запустить скрипт первоначальной настройки **setup.sh**, который находится в папке **«safephone»**. После этого необходимо ответить на вопросы системы для создания конфигурационных файлов с заданными параметрами и файла **«docker-compose.yml»** с указанными в ответах серверными компонентами (для ответов на вопросы предоставляются подсказки: y − да, n − нет, q − выход из настройки, ? − справочная информация):

```
ARM [y/n/q/?]? y   # Формировать конфигурацию сервера
веб-приложений SafePhone?

ARM: Use HTTPS [y/n/q/?]? y   # Использовать протокол
HTTPS для сервера веб-приложений SafePhone?

ARM: Create HTTPS certificate [y/n/q/?]? y
# Формировать самоподписанный сертификат HTTPS сервера
веб-приложений SafePhone?

ARM: Common Name (IP or domain name): 192.168.XX.XX
# Адрес или доменное имя сервера веб-приложений
SafePhone

iOS MDM [y/n/q/?]? y   # Формировать конфигурацию
сервера iOS MDM
```

```
You need to manually provide 'config/skb.key' file!

You need to manually provide 'config/skb.chain' file!

You need to manually provide 'config/MdmPush.pem' file!

You need to manually provide 'config/MonitorPush.pem'
file!   # Информация о необходимости самостоятельного
предоставления ключа и сертификата skb, сертификатов
MdmPush и MonitorPush

iOS MDM: Create HTTPS certificate [y/n/q/?]? y
# Формировать самоподписанный сертификат HTTPS сервера
iOS MDM?

iOS MDM: Common Name (IP or domain name): 192.168.XX.XX
# Адрес или доменное имя сервера iOS MDM

Socket server [y/n/q/?]? y # Формировать конфигурацию
сервера управления SafePhone (Socket server)?

Socket server: Create HTTPS certificate [y/n/q/?]? y
Формировать самоподписанный сертификат HTTPS сервера
управления SafePhone (Socket server)?

Socket server: Common Name (IP or domain name):
192.168.XX.XX   # Адрес или доменное имя сервера
управления SafePhone (Socket server)

Asterisk [y/n/q/?]? y   # Формировать конфигурацию
сервера VoIP телефонии SafePhone (SIP-сервер)

Database type [o/p/q/?]? y   # Формировать конфигурацию
сервера баз данных SafePhone

o - Oracle

p - PostgreSQL

q - quit

? - print help
```

```
Database type [o/p/q/?]? p   # Тип сервера баз данных (Oracle / Postgresql)

Database hostname: 127.0.0.1   # Адрес сервера БД

Database port (default: 5432):  5432   # Порт сервера БД (по умолчанию для PostgreSQL - 5432, для Oracle - 1521)

Database username:  sphone    # Пользователь БД

Database password:   # Пароль пользователя БД

Database name:  sphone   # Имя сервера БД
```

При выборе БД Oracle параметры будут отображаться следующим образом:

```
Database type [o/p/q/?]? o

Database hostname:  127.0.0.1

Database port (default: 1521):  1521

Oracle system identifier (SID):  cuks

Database username:  sphone

Database password:
```

9 Запустить установку docker-контейнеров с помощью команды:

```
docker-compose up -d
```

10 Проверить наличие созданных docker-образов и docker-контейнеров следующими командами:

```
docker images -a

docker ps -a
```

Если в результате проверки, кроме созданных компонентов, отобразились docker-образы и docker-контейнеры от более ранних запусков, их следует удалить.

5 Управление серверными компонентами системы «SafePhone»

1. Просмотреть текущие версии установленных компонентов можно следующими командами:

```
docker-compose exec arm rpm -q safephone-arm

docker-compose exec iosmdm rpm -q safephone-iosmdm

docker-compose exec iosmdm-bg rpm -q safephone-iosmdm

docker-compose exec nginx rpm -q safephone-iosmdm-ui-data

docker-compose exec sipsync rpm -q safephone-sipsync

docker-compose exec socket-server rpm -q safephone-socket-server

docker-compose exec asterisk rpm -q asterisk-safephone asterisk
```

2. При изменениях в конфигурации серверных компонентов следует перезапустить docker-контейнеры следующей командой:

```
docker-compose restart <имя компонента>
```

3. Обновление docker-образов осуществляется следующими командами, при этом необходимо сначала остановить и удалить docker-контейнеры, затем обновить версии в файле **«docker-compose.yml»** и запустить docker-контейнеры:

```
docker-compose down

docker load -i safephone-docker.tar.gz

docker-compose up -d
```

4. При внесении изменений в файл **«docker-compose.yml»**, например, добавление/удаление компонентов, следует обновить docker-образы следующей командой:

```
docker-compose up -d
```

6 Обновление системы «SafePhone» до текущей версии

Для обновления уже установленной системы «SafePhone» версии 3.0 до текущей версии дополнительно в установочный комплект входят следующие файлы:

- db-postgresql-patch-4.0.0-4.1.tar.gz (для БД PostgreSQL);
- db-oracle-patch-4.0.0-4.1.tar.gz (для БД Oracle).

Примечание – Перед обновлением системы рекомендуется сделать резервную копию БД.

Чтобы обновить систему следует выполнить следующие операции:

1 Остановить docker-контейнеры следующей командой:

```
docker-compose down
```

2 Установить новые docker-образы серверных компонентов из архива **safephone-docker.tar.gz** посредством команды:

```
docker load -i safephone-docker.tar.gz
```

3 Переименовать директорию **«safephone»** в директорию **«safephone-old»**:

```
mv -R /opt/safephone /opt/safephone-old
```

4 Создать папку **«safephone»** и распаковать в нее архив с конфигурацией компонентов SafePhone в Docker (файл **safephone-config.tar.gz**) с помощью команды:

```
mkdir /opt/safephone
```

```
tar -xzvf safephone-config.tar.gz -C /opt/safephone
```

5 В директории **«safephone»** создать папку **«db»** и распаковать в нее один из выбранных архивов обновлений (файл **db-postgresql-patch-4.0.0-4.1.tar.gz** – для обновления системы с БД PostgreSQL, **db-oracle-patch-4.0.0-4.1.tar.gz –** для обновления системы с БД Oracle)
Например, для БД PostgreSQL команды будут следующими:

```
mkdir /opt/safephone/db
```

```
tar -xzvf db-postgresql-patch-4.0.0-4.1.tar.gz -C
/opt/safephone/db
```

6 Установить схему БД из папки **«db»** посредством скрипта **install.sh**. При запуске скрипта следующей командой:

```
./install.sh
```

будет предложена установка схемы БД с параметрами по умолчанию, а именно:

```
File: ./sql/schema.sql   # Название файла для установки
Database: sphone   # Имя базы данных
User: sphone   # Имя пользователя
Scheduler: scheduler   # Имя пользователя планировщика
Schema: sphone   # Название схемы
Continuie (y/n)?
```

Для продолжения работы с предложенными параметрами, следует написать **«y»**. В противном случае написать **«n»** и запустить скрипт с указанием требуемых параметров. Если какой-то из параметров схемы не будет указан, следовательно, в схему будет включен параметр по умолчанию:

```
./install.sh [--user username] [--db dbname] [--
scheduler scheduler] [--schema schema] -- [--
host=192.168.XX.XX]     # host=192.168.XX.XX - адрес
сервера БД
```

Для получения справки по параметрам схемы требуется запустить скрипт с ключом **-h** или **--help.**

Пример команды для скрипта **install.sh** приведен ниже:

```
./install.sh -- --host=192.168.0.1
```

Затем требуется подтвердить установку схемы БД вводом пароля пользователя.

7 В распакованном архиве конфигурации в папке **«safephone / config»** содержатся минимальные конфигурационные файлы компонентов SafePhone, описание которых приведены в разделе 3 (перечисление 5). Следует сравнить конфигурационные файлы релиза 4.1 в папке **«safephone»** с файлами релиза 4.0.0 в папке **«safephone-old»** и дополнить их, при необходимости, недостающими значениями портов из старых файлов. В конфигурационном файле БД указать параметры в соответствии установленной схемой БД.

8 Скопировать сертификационные файлы формата CSR, KEY, CHAIN, PEM из папки **«safephone-old / config»** в папку **«safephone / config»**. Создание и параметры файлов сертификации описаны в 3.1. Пример команд приведен ниже:

```
cd /opt/safephone-old/config
cp /opt/safephone-old/config/$(ls *.csr *.key *.chain *.pem) /opt/safephone/config/
```

11 Запустить установку docker-контейнеров с помощью команды:

```
docker-compose up -d
```

12 Проверить наличие созданных docker-образов и docker-контейнеров следующими командами:

```
docker images -a
docker ps -a
```

Docker-образы и docker-контейнеры от более ранних запусков следует удалить.

7 Проверка работоспособности системы «SafePhone»

Для контроля работоспособности системы «SafePhone» требуется:

1. Войти в АРМ Администратора SafePhone, для этого в адресной строке браузера ввести https://ip-address:8443, (вместо <ip-address:8443> следует указать адрес сервера, на котором установлена система «SafePhone»). Если АРМ запущен, отобразится страница авторизации.
2. В таблице МСК главного окна выбрать **подключенный, незаблокированный и доступный для управления** комплект в соответствии с рисунком 7.1, у которого:
   - состояние соединения МСК, которое отображается в столбце «Статус», – в сети;
   - состояние блокировки МСК, которое отображается в столбце «Статус», – не заблокирован;
   - состояние управления устройством, которое отображается в столбце «Статус», – доступно для управления.

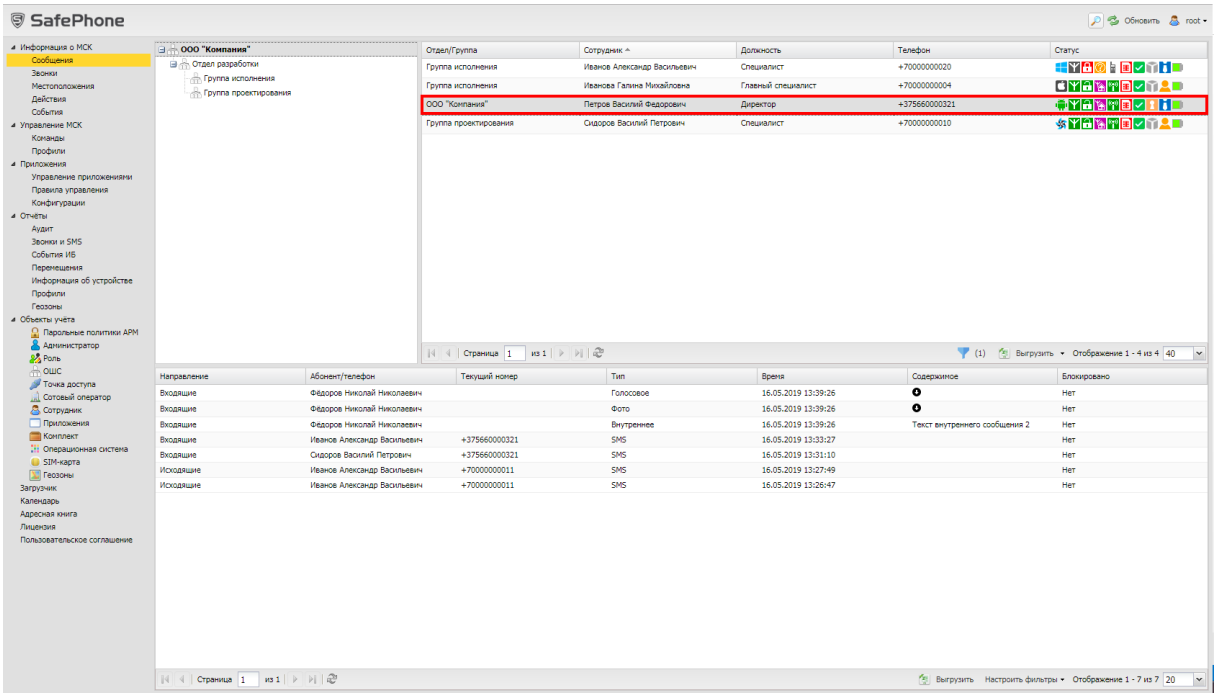

Рисунок 7.1 – Выбор подключенного незаблокированного комплекта

3. В главном меню выбрать раздел «Команды» и отправить на устройство команду «Переподключение» соответствии с рисунком 7.2, с параметром 10 с. Затем в окне «Уведомления» нажать кнопку «ОК».

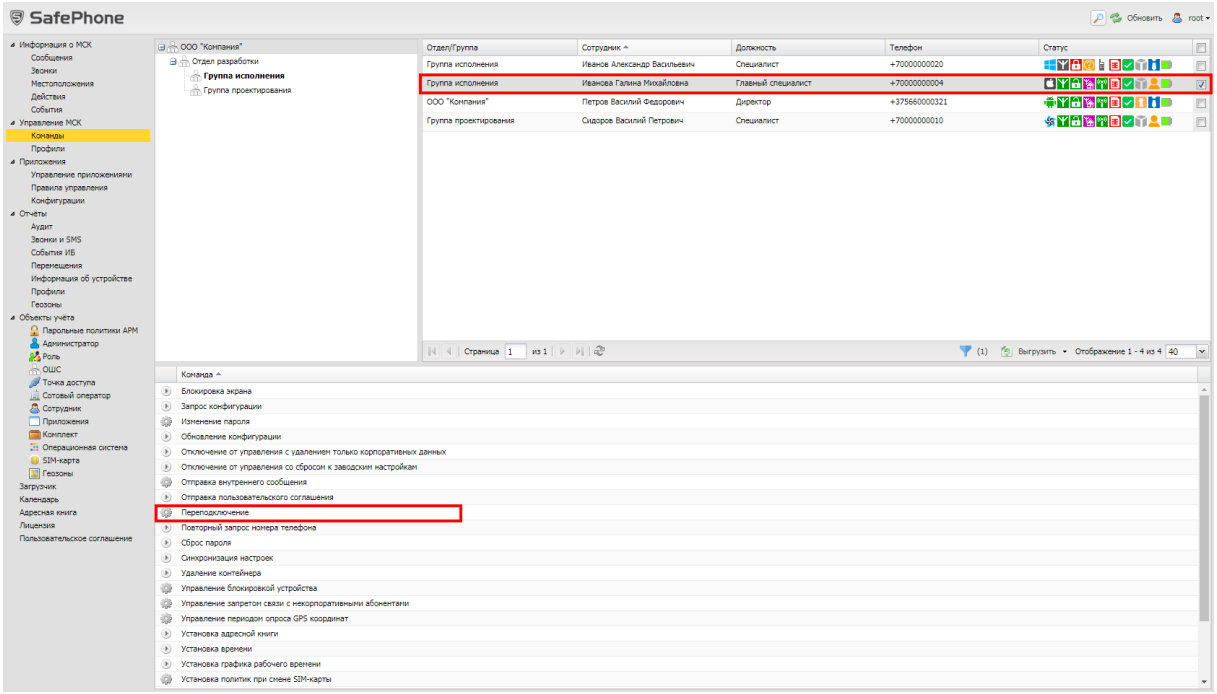

Рисунок 7.2 – Отправка команды «Переподключение»

4. Дождаться результата выполнения действия: когда значение в разделе «Действие» изменится на значение, отличное от «Ожидание результата»:
    - результат «Нормальное завершение» свидетельствует о работоспособности системы «SafePhone» (рисунок 7.3);

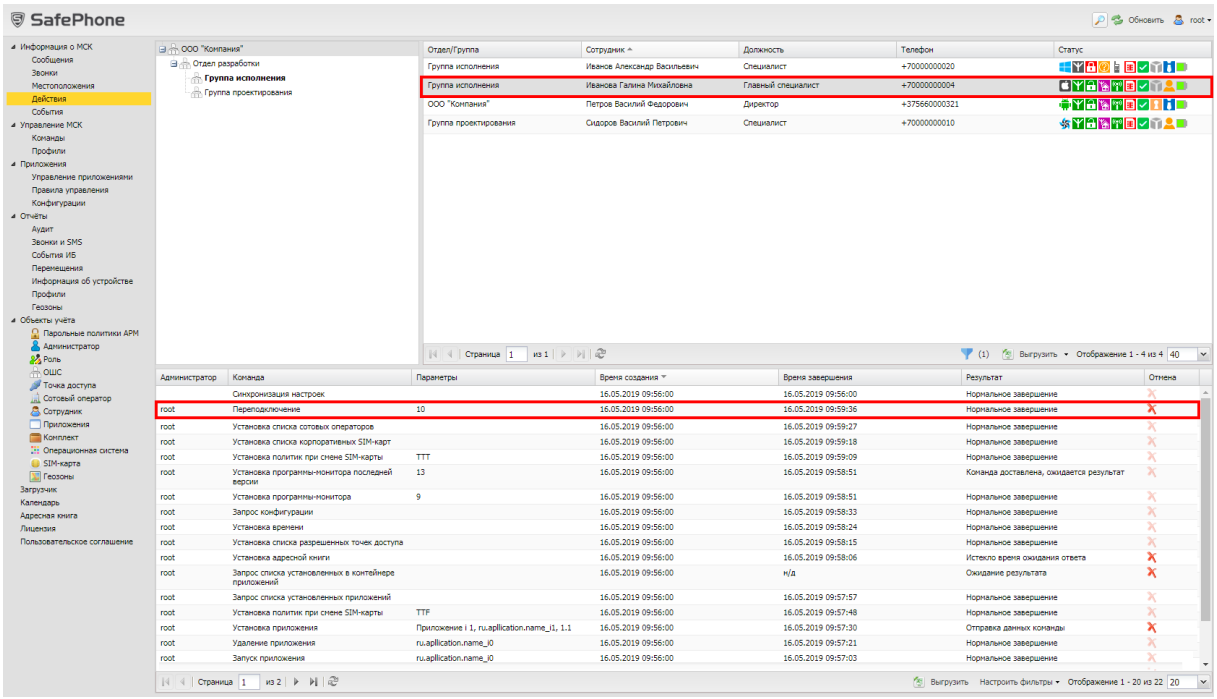

Рисунок 7.3 – Результат команды «Переподключение»

- значение результата, отличное от «Нормальное завершение», свидетельствует о возможном нарушении работоспособности системы.

Приложение А
Форма запроса на получение сертификата

В приложении приведен пример заполнения запроса на сертификат **SKB**, где выделенные параметры обязательны к заполнению и предназначены для идентификации сертификата заказчика. При формировании запроса на сертификат **HTTPS** или **APNs** следует привести данные для требуемых сертификатов в соответствии с описанием в 3.1.

```
    Generating a 2048 bit RSA private key
.....................+++
.....................+++
writing new private key to 'skb.key'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:RU   # Двухбуквенный код
страны
State or Province Name (full name) []:Moscow   # Название
округа
Locality Name (eg, city) [Default City]:Cuchino   # Название
города
Organization Name (eg, company) [Default Company Ltd]:Company
XXX   # Название организации
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:skb
# Общее имя сервера
```

```
Email Address []:...# Адрес электронной почты службы
технической поддержки

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:...# Пароль с запросом
An optional company name []:...# Дополнительное название
компании
```