

# VIPole Corporate Server

Installation Manual for Linux Debian

v.2.01.2017



**VIPole**  
[www.vipole.com](http://www.vipole.com)

**SECURE AND ENCRYPTED MESSENGER**  
for secure communications and encrypted data storage

Windows ■ Mac OS ■ Linux ■ Android ■ iOS ■ Enterprise solutions

## Определения и требования

Серверная часть состоит из нескольких компонентов (при первоначальной установке требуется только два из них):

1. Основной сервис, в дальнейшем называемый «Сервер». Используется для аутентификации пользователей, организации чатов, передачи файлов и администрирования. Исполняемый файл — vipole-server.
2. Сервис для голосовых и видеозвонков и конференций, в дальнейшем называемый «Медиаретранслятор». Исполняемый файл — vipole-relay.
3. Сервис автоматического конфигурирования клиентского приложения, называемый в дальнейшем Сервер Автоконфигурации можно не устанавливать.

Во время инсталляции потребуется установка дополнительных компонентов:

- MongoDB , версия 2.6
- Redis, версия 2.8

## Требования к системе

- Debian GNU/Linux Debian Wheezy 7.x или Jessie 8.x архитектуры i386 или amd64 (рекомендуется Debian Jessie 8.x amd64)
- Многоядерный процессор (минимум 2 ядра)
- 4 ГБ RAM
- 500 ГБ HDD (RAID1)

## Настройки брандмауэра Linux

Проверьте настройки брандмауэра Linux. Настройте его так, чтобы не блокировались используемые Сервером порты. Номера портов можно посмотреть в файлах конфигурации Сервера и Медиаретранслятора.

**Внимание**, в файле relay.config указывается диапазон портов:

### server.config:

```
# Server port  
listen-port = 37210
```

### relay.config

```
# UDP ports range  
# Minimum UDP port  
min-port=3000
```

```
# Maximum UDP port
max-port=9000
```

#### server-auto.config

```
# Server port
listen-port = 37212
```

### **Этапы установки Сервера**

**1.** Установите Wheezy 7.x или Debian Jessie 8.x архитектуры amd64.

**2.** Установите MongoDB, самую свежую версию из ветки 2.6 (2.6.12) из репозитория Mongo.

**2.1.** Импортируйте публичный ключ для системы управления пакетами:

```
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10
```

**2.2.** Создайте в `/etc/apt/sources.list.d/mongodb.list` ссылку на репозиторий для MongoDB:

```
$ sudo echo 'deb http://downloads-
distro.mongodb.org/repo/debian-sysvinit dist 10gen' |
sudo tee /etc/apt/sources.list.d/mongodb.list
```

**2.3.** Обновите локальную базу пакетов:

```
$ sudo apt-get update
```

**2.4.** Установите последнюю стабильную версию MongoDB из ветки 2.6:

```
$ sudo apt-get install mongodb-org
```

**2.5.** Убедитесь, что MongoDB запущена:

```
$ sudo service mongod status
```

**3.** Установка Redis, версия 2.8 .

**3.1.** Установка готового пакета в Debian Wheezy 7.x.

Так как для дистрибутива Debian Wheezy в его репозитории есть только Redis 2.6 , то установить его придётся из стороннего репозитория.

Скачайте пакеты для redis-tools и redis-server в соответствии с архитектурой вашей ОС и установите их

для i386:

```
$ wget
http://archives.dotdeb.org/dists/wheezy/redis/2.8.17/binary-i386/redis-tools\_2.8.17-1~dotdeb.1\_i386.deb
```

```
$ wget
http://archives.dotdeb.org/dists/wheezy/redis/2.8.17/binary-i386/redis-server\_2.8.17-1~dotdeb.1\_i386.deb
```

```
$ sudo dpkg -I redis-tools_2.8.17-1~dotdeb.1_i386.deb
```

```
$ sudo dpkg -I redis-server_2.8.17-1~dotdeb.1_i386.deb
```

для amd64:

```
$ wget
http://archives.dotdeb.org/dists/wheezy/redis/2.8.17/binary-amd64/redis-tools\_2.8.17-1~dotdeb.1\_amd64.deb
```

```
$ wget
http://archives.dotdeb.org/dists/wheezy/redis/2.8.17/binary-amd64/redis-server\_2.8.17-1~dotdeb.1\_amd64.deb
```

```
$ sudo dpkg -i redis-tools_2.8.17-1~dotdeb.1_amd64.deb
```

```
$ sudo dpkg -i redis-server_2.8.17-1~dotdeb.1_amd64.deb
```

### 3.2. Установка готового пакета Redis в Debian Jessie 8.x.

```
$ sudo apt-get install redis-server
```

## 4. Установка Сервера.

### 4.1. Установите пакет программ Сервера:

```
$ sudo dpkg -i vipoleserver-standalone_1.1.0_amd64.deb
```

**4.2.** Проинициализируйте базу данных Сервера, создав учетные записи для администратора Сервера и сервиса Медиаретранслятора. Прочтите и подтвердите лицензионное соглашение. У вас будут запрошены пароли для создаваемых учетных записей.

```
$ vipoleadm --domain <доменное имя> --dbname <имя базы данных> \  
--operation init \  
--login <admin_login>\  
--relayserver-login <relay_server_login>
```

Если вы решили проинициализировать базу заново, необходимо сначала удалить старую. Для этого надо запустить программу mongo, указав в качестве параметра имя базы, указанное при инициализации.

```
$ mongo <имя базы данных>
```

в ответ на приглашение программы mongo, необходимо выполнить команду `db.dropDatabase()` ;

а затем  
`exit`

**ВАЖНО!** При инициализации базы изменяется уникальный идентификатор, к которому привязывается лицензия. Поэтому повторная инициализация допускается только до выполнения п.4.5

Инициализация базы после генерации запроса на лицензию приведет к тому, что ваша лицензия становится недействительной. Восстановить её будет невозможно и вам придется приобретать ее заново.

**4.3.** Теперь необходимо настроить лицензирование

**4.3.1.** Сгенерируйте запрос на лицензию:

```
$ vipoleadm --domain <доменное имя> --dbname <имя базы данных> \  
--operation license_request \  
--license-request-file license_request.req
```

**Важно!** Лицензия будет привязана к проинициализированной базе.

Обязательные параметры:

```
--operation license_request
--domain — ваш домен, указанный в п.4.2
--dbname — имя базы данных, указанное в п.4.2
--license-request-file — имя (путь к файлу), куда будет записан запрос
на лицензию. Обратите внимание, что имя (путь к файлу) не должно содержать
символов национальных алфавитов и полностью состоять из латинских
символов и/или цифр.
```

**Файл запроса лицензии необходимо отправить нам по электронной почте по адресу [contact@vipole.com](mailto:contact@vipole.com) . Как только файл запроса лицензии будет получен, мы пришлем вам обратно файл лицензии.**

В дальнейшем при добавлении лицензий файл запроса не потребуется. Просто отправьте нам письмо с указанием количества дополнительных лицензий. После оплаты вы получите файл с дополнительными лицензиями.

**4.3.2.** Получив обратно файл лицензии (к примеру, **`vipole_license.dat`**), сначала проверьте его валидность и просмотрите его содержимое:

```
$ vipoleadm --domain <доменное имя> --dbname <имя базы
данных> \
--operation verify_license \
--license-file vipole_license.dat
```

Обязательные параметры:

```
--operation verify_license
--domain — ваш домен, указанный в п.4.2
--dbname — имя базы данных, указанное в п.4.2
--license-file — имя (путь к файлу) лицензии. Обратите внимание, что
имя и путь к файлу не должны содержать символов национальных алфавитов.
```

**4.3.3.** Для активации лицензии выполните команду:

```
$ vipoleadm --domain <доменное имя> --dbname <имя базы
данных> \
--operation activate_license \
--license-file vipole_license.dat
```

Обязательные параметры:

**--operation verify\_license**  
**--domain** — ваш домен, указанный в п.4.2  
**--dbname** — имя базы данных, указанное в п.4.2  
**--license-file** — имя (путь к файлу) лицензии. Обратите внимание, что имя (путь к файлу) не должно содержать символов национальных алфавитов.

**4.3.4.** В любой момент вы можете просмотреть текущий статус всех лицензионных пакетов:

```
$ vipoleadm --domain <доменное имя> --dbname <имя базы данных> \ --operation print_server_license
```

Если лицензия одна, то будет выведено ее содержимое. Если лицензионных пакетов несколько, то будет выведено содержимое всех лицензионных пакетов, а также совокупная лицензия, сформированная объединением всех актуальных для текущей даты и версии сервера лицензионных пакетов.

**4.4.** Сгенерируйте сертификат Сервера, используя OpenSSL:

**4.4.1.** Сгенерируйте ключ для Сервера

```
$ openssl genrsa -des3 -out cakey.pk 3072
```

На запрос «Enter pass phrase for» придумайте и введите пароль. Он в дальнейшем будет использоваться при конфигурировании Сервера.

**4.4.2.** Сгенерируйте запрос сертификата:

```
$ openssl req -new -key cakey.pk -out server.csr
```

На запрос «Enter pass phrase for» введите пароль из п.4.4.1.

**4.4.3.** На Сервере допустимо использование самоподписанных сертификатов:

```
$ openssl x509 -req -days 3650 -in server.csr -signkey cakey.pk -out cacert.pk
```

На запрос «Enter pass phrase for» введите пароль из п.4.4.1.

**4.5.** Отредактируйте **/etc/vipole/server.config** .

Укажите домен, IP-адрес Сервера, полные пути к сертификату и приватному ключу, а также пароль приватного ключа.

Приватный ключ сервера в формате PEM, полученный в п.4.4.1

```
private-key-file = /etc/vipole/cakey.pk
```

Пароль для приватного ключа из п.4.4.1

```
private-key-file-passphrase = secret_passphrase
```

Сертификат в формате PEM, полученный в п.4.4.3

```
certificate-chain-file = /etc/vipole/cacert.pk
```

IP адрес интерфейса, на котором Сервер будет принимать подключения. Адрес 0.0.0.0 означает «все интерфейсы». Можно так же указать конкретный IP-адрес внешнего интерфейса сервера Debian.

```
listen-address = 0.0.0.0
```

Домен Сервера тот же, что и в п.4.2

```
domain = vipole.example
```

#### **4.6. Отредактируйте /etc/vipole/relay.config .**

Укажите логин и пароль учетной записи Медиаретранслятора, внешний IP-адрес Сервера и полный путь к сертификату Сервера.

Внешний IP-адрес, к которому будут подключаться клиентские приложения. Обычно это адрес внешнего сетевого интерфейса сервера Debian. В случае, если вы установили Сервер за NAT-ом и используете проброс портов (port forwarding), в качестве external-listen-address следует указывать внешний адрес NAT'a.

```
external-listen-address=192.168.1.80
```

Логин для подключения Медиаретранслятора к Серверу. Этот логин был создан в п.4.4 опцией --relayserver-login. Если вы используете несколько Медиаретрансляторов, они должны использовать общий логин

```
login=relay_server_login
```

Пароль учетной записи Медиаретранслятора, установленный в п.4.4.

```
passwd=secret_password
```



Домен Сервера тот же, что и в п.4.2

```
domain=vipole.example
```

Сертификат Сервера, полученный в п.4.5.3

```
certificate=/etc/vipole/cacert.pk
```

#### 4.7. Запуск Сервера.

Сначала запустите Сервер с консоли, чтобы убедиться, что в файле конфигурации нет ошибок:

```
$ vipole-server --config /etc/vipole/server.config
```

Если в течение 30 секунд не появилось приглашение командной строки, значит сервер успешно стартовал. Прервите его работу, нажав CTRL+C.

Затем запустите его командой:

```
$ vipole-server --config /etc/vipole/server.config &
```

#### 4.8. Запустите Медиаретранслятор:

Сначала запустите Медиаретранслятор с консоли, чтобы убедиться, что в файле конфигурации нет ошибок:

```
$ vipole-relay --config /etc/vipole/relay.config
```

Если в течение 30 секунд не появилось приглашение командной строки, значит, Медиаретранслятор успешно стартовал. Прервите его работу, нажав CTRL+C.

Затем запустите его командой:

```
$ vipole-relay --config /etc/vipole/relay.config &
```

5. Настройте Сервер Автоконфигурации. Он используется для автоматической настройки параметров подключения клиентских проложений.

**При первом запуске и ознакомлении с возможностями Сервера настройку и запуск Сервера Автоконфигурации можно пропустить и перейти к п.6.**

##### 5.1. Сгенерируйте ключ для Сервера Автоконфигурации

```
$ openssl genrsa -des3 -out autoconfkey.pk 3072
```

## 5.2. Сгенерируйте запрос сертификата

```
$ openssl req -new -key autoconfkey.pk -out autoconf.csr
```

**5.3. Файл запроса сертификата необходимо направить нам по электронной почте по адресу [contact@vipole.com](mailto:contact@vipole.com). Обратно мы пришлем вам подписанный сертификат. Только он может быть использован для работы сервера автоконфигурации.**

## 5.4. Отредактируйте файл `/etc/vipole/client_autoconf.config`:

Это файл сертификата, который вы получили обратно в п. 5.3

```
autoconf-sign-certificate-  
file=/etc/vipole/vipole_autoconf.cert
```

Это приватный ключ Сервера Автоконфигурации, полученный в п.5.1

```
autoconf-sign-private-key-file=  
/etc/vipole/autoconfkey.pk
```

Поставьте значение этого параметра 1, если хотите, чтобы клиенты вводили код доступа при автоконфигурации

```
autoconf-use-access-code=0
```

Здесь надо указать код доступа, если он используется

```
#autoconf-access-code=
```

Это файл сертификата Сервера, полученный в п. 4.5

```
vipole-server-certificate-file=/etc/vipole/cacert.pk
```

Укажите DNS имя или IP адрес вашего Сервера (IP адрес должен совпадать со значением параметра `external-listen-address`, установленным в п.4.8)

```
vipole-server-host=192.168.1.80
```

Номер порта Сервера

```
vipole-server-port=37210
```

### 5.5. Сохраните настройки автоконфигурации в базе данных

```
$ vipoleadm --domain <доменное имя> --dbname <имя базы  
данных> \  
--operation save_clients_autoconf \  
--clients-autoconf-file client_autoconf.config
```

На запрос «Enter passphrase to decrypt private key» введите пароль из п.5.1

Если вам потребуется сменить настройки автоконфигурации, отредактируйте файл `client_autoconf.config` и опять сохраните настройки автоконфигурации в базе данных. После сохранения параметров автоконфигурации необходимо перезапустить только Сервер Автоконфигурации.

### 5.6. Отредактируйте файл `/etc/vipole/server-auto.config`.

Укажите полный путь к файлу приватного ключа, полученного в п. 5.1

```
private-key-file = /etc/vipole/autoconfkey.pk
```

Укажите пароль приватного ключа, полученного в п. 5.1

```
private-key-file-passphrase = secret_passphrase
```

Укажите полный путь к файлу сертификата, который вы получили в п 5.3

```
certificate-chain-file =  
/etc/vipole/vipoledemoautoconf.cert
```

Укажите домен вашего Сервера

```
domain = vipole.example
```

Укажите IP-адрес Сервера Автоконфигурации. Здесь необходимо указать IP адрес, на котором Сервер Автоконфигурации будет принимать подключения. Адрес 0.0.0.0 означает «все интерфейсы». Можно так же указать конкретный IP-адрес внешнего интерфейса сервера Debian.

```
listen-address = 0.0.0.0
```

Укажите номер порта Сервера Автоконфигурации

```
listen-port = 37212
```

**Следующие настройки должны совпадать с настройками Сервера:**

Укажите IP адрес MongoDB сервера

```
mongo-address = 127.0.0.1
```

TCP порт MongoDB сервера

```
mongo-port = 27017
```

Имя базы данных

```
dbname = vipole_server
```

### **5.7. Запуск Сервера Автоконфигурации.**

Сначала запустите Сервер Автоконфигурации с консоли, чтобы убедиться, что в файле конфигурации нет ошибок :

```
vipole-server --autoconf --config /etc/vipole/server-  
auto.config
```

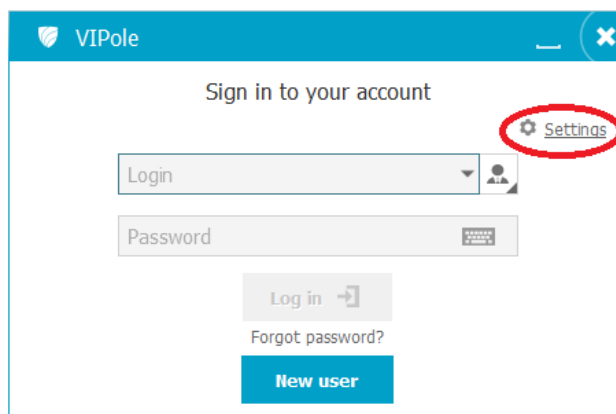
Если в течение 30 секунд не появилось приглашение командной строки, значит, сервер успешно стартовал. Прервите его работу, нажав CTRL+C.

Затем запустите его командой:

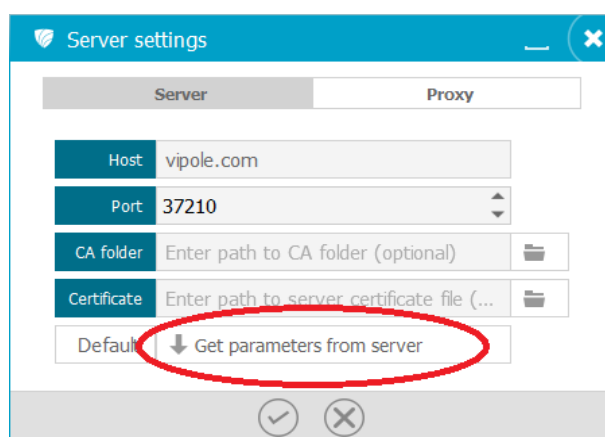
```
vipole-server --autoconf --config /etc/vipole/server-  
auto.config &
```

Проверьте его работу, сконфигурировав клиента, как описано в следующем пункте.

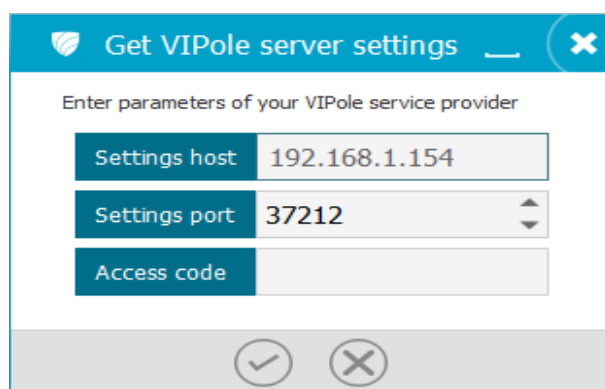
**5.8.** Для использования автоконфигурации в клиентском приложении, нажмите ссылку «⚙ Параметры» при незаполненном поле «Логин».




**5.8.1.** Нажмите на ссылку «⬇ Загрузить параметры с сервера»



**5.8.2.** В открывшемся окне укажите IP адрес сервера конфигурации и номер порта. Если вы указали использование кода доступа, то надо ввести и его.



После нажатия на кнопку  клиентское приложение получит с сервера все сетевые настройки.

**5.9.** Для автоматического запуска и рестарта в случае сбоев сервисов используется vipole-launcher.

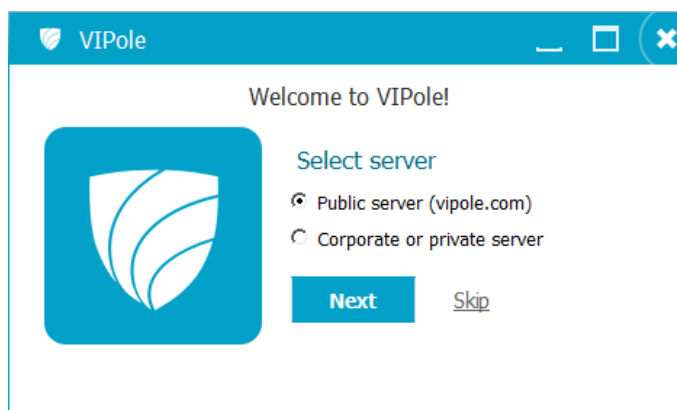
При установке пакета Сервера добавляется файл /etc/init.d/vipoleserver  
Файл конфигурации используется /etc/vipole/launcher.config  
Если вы не используете Сервер Автоконфигурации, необходимо  
закомментировать (символ # в начале строки) всё секцию [application.2]

Перед запуском скрипта /etc/init.d/vipoleserver убедитесь, что вы остановили  
Сервер и Медиаретранслятор, запущенные из командной строки.

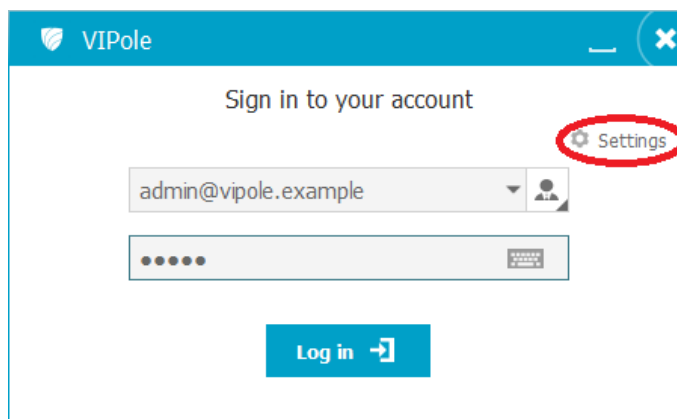
Для предотвращения бесконечного перезапуска сервиса и разрастания логов  
vipole-launcher производит ограниченное количество перезапусков (10  
неудачных подряд)

## 6. Установите и запустите клиентское приложения.

При первом запуске клиентского приложения нажмите «Пропустить» в  
появившемся окне.




Введите логин@домен и пароль администратора, которые вы указали в п.4.5.

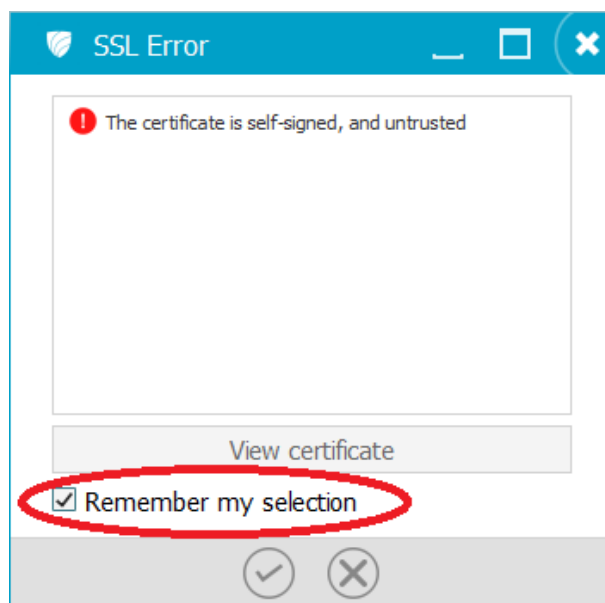


Перейдите по ссылке «⚙ Параметры» и введите в поле «Хост» доменное имя  
или IP-адрес вашего Сервера.

Обратите внимание, если вы ввели IP адрес вашего Сервера вместо доменного имени, в поле «Логин» вы должны указать логин пользователя вместе с названием домена, например, admin\_login@vipole.example

Нажмите . Нажмите «Вход»

Так как сейчас мы используем самоподписанный сертификат и не указали путь к его локальной копии, при подключении вы получите сообщение:



Поставьте галочку «Запомнить мой выбор» и нажмите .

**6.1.** Сгенерируйте ключи шифрования и выберите секретную фразу (как при первом подключении клиента).

**6.2.** Когда клиент запустится, перейдите в главное меню программы в пункт «Расширения», а там выберите «Администратор VIPOLE».

Затем создайте административный пароль и введите его для управления доменом безопасности.

The screenshot shows the 'VIPole Administrator' window with the 'Online users' tab selected. It displays a table with columns: Login, Desired login, EMail, Nickname, Security Domain, Last logged in, Last IP, and Registered. Two users are listed: 'admin' and 'relay\_server'. The interface includes search filters, a 'Refresh' button, and a 'Reset filters' button. The bottom status bar indicates '1 of total 1 pages (2 records)'.

	Login	Desired login	EMail	Nickname	Security Domain	Last logged in	Last IP	Registered
1	admin	admin				2016.08.03 17:49:48	192.168.1.126	2016.08.03 11:39:37
2	relay_server	relay_server				2016.08.03 17:11:09	127.0.0.1	2016.08.03 11:39:37

Обратите внимание, что на данный момент у вас не создано ни одного домена безопасности и вы управляете всеми пользователями сервера. Вы можете только добавлять пользователей, просматривать журнал подключений и список пользователей онлайн. Дополнительные возможности управления пользователями см. «VIPOLE. Руководство администратора домена безопасности».

Расширенные возможности управления пользователями (назначение секретной фразы, установка настроек безопасности, управление контактами) возможно только внутри домена безопасности. Добавить домен безопасности можно, используя консольную утилиту **vipoleadm**:

```
$ vipoleadm --operation create_security_domain \
--domain vipole.example --dbname vipole_server \
--login <логин администратора домена безопасности> \
--security-domain-user-count <число пользователей>
```

Пользователю, назначили администратором домена безопасности, необходимо выйти и заново войти в «Администратор VIPOLE».

Параметром **--login <логин админа домена безопасности>** указывается логин пользователя, который будет администратором домена безопасности. Это может быть администратор сервера или любой вновь добавленный пользователь.

Запустив расширение «Администратор VIPOLE» в клиентской программе, он получит расширенные возможности управления пользователями, включенными в домен безопасности.

Параметром **--security-domain-user-count <количество\_пользователей>** - устанавливается лимит пользователей домена безопасности.

Можно завести несколько доменов безопасности, например, для различных подразделений вашей организации.



После создания домена безопасности, в случае необходимости, можно увеличить лимит пользователей домена безопасности.

```
vipoleadm --operation add_security_domain_user_limit \  
--domain vipole.example --dbname vipole_server \  
--security-domain-user-count <количество_пользователей> \  
--security-domain-id <идентификатор>
```

**--domain** - ваш домен

**<количество пользователей>** - на сколько увеличивается лимит пользователей домена безопасности.

**<идентификатор>** - идентификатор домена безопасности, лимит пользователей которого изменяется. Идентификатор можно посмотреть на странице «Параметры» расширения «Администратор VIPOLE».

**Суммарное количество пользователей доменов безопасности не должно превышать общего лимита пользователей демо лицензии в 10 лицензий .**

Так же администратор может удалить пользователей из домена безопасности.

```
vipoleadm --operation move_user_from_security_domains \  
--domain vipole.example --dbname vipole_server --login \  
<логин>
```

**--domain** - ваш домен

**<логин>** - имя удаляемого из домена безопасности пользователя

Если превышен лимит пользователей на сервере, можно заблокировать ненужных пользователей командой:

```
vipoleadm --operation block_user ^ \  
--domain vipole.example --dbname vipole_server --login \  
<логин>
```

**--domain** - ваш домен

**<логин>** - имя блокируемого пользователя

Лицензией учитываются только активные пользователи.

## ПРИЛОЖЕНИЕ 1. Установка сертификата, выданного Удостоверяющим Центром

Кроме сертификатов, подписанных вашим собственным удостоверяющим центром, Вы можете использовать SSL сертификаты сторонних удостоверяющих центров, например, Comodo или Thawte.

1. Генерируем ключ в соответствии с требованиями Удостоверяющего центра (в частности, длина ключа должна соответствовать требованиям УЦ)

```
openssl genrsa -des3 -out cakey.pk 2048
```

2. Создаем запрос сертификата.

```
openssl req -new -key cakey.pk -out server.csr
```

Заполняем поля в соответствии с требованиями Удостоверяющего центра. Обязательно указываем Common Name (здесь надо указать доменное имя) и Email Address (контактный e-mail)

3. Отправляем запрос сертификата server.csr на подпись в Удостоверяющий центр.

4. Обратно мы получаем свой подписанный сертификат (к примеру, myserver.crt) и несколько сертификатов, составляющих цепочку (где каждый сертификат подписан следующим). Цепочка заканчивается корневым сертификатом.

5. Собираем цепочку сертификатов в один файл, начиная с сертификата нашего сервера и заканчивая корневым.

```
cat myserver.crt intermediate1.crt intermediate2.crt  
rootCA.crt > crt-chain.crt
```

6. Настраиваем Сервер на использование цепочки сертификатов

a. В файле конфигурации сервера указываем файл ключа

```
private-key-file = cakey.pk
```

b. Указываем парольную фразу ключа

```
private-key-file-passphrase = <SecretPhrase>
```

c. Указываем файл цепочки сертификатов

```
certificate-chain-file = crt-chain.crt
```

d. В файле автоконфигурации клиента указываем конечный корневой сертификат

```
vipole-server-certificate-file = rootCA.crt
```

e. Обновляем параметры автоконфигурации клиента

```
vipoleadm --domain <ваш домен> --dbname <имя базы> ^  
-o save_clients_autoconf  
--clients-autoconf-file client_autoconf.config
```

f. Перезапускаем все сервисы

Клиентское приложение должно получить корневой сертификат цепочки или с помощью Сервера Автоконфигурации или надо указать путь к папке и файл корневого сертификата при подключении в окне «Параметры сервера». В противном случае появится сообщение о самоподписанном сертификате или невозможности проверить издателя сертификата.

### **Рассмотрим установку на примере SSL сертификата Comodo PositiveSSL**

Сначала Вы должны сгенерить ключ, запрос сертификата и отправить его в Удостоверяющий центр Comodo. Обрато Вы получите комплект сертификатов из 4 файлов:

1. AddTrustExternalCARoot.crt — сертификат корневого центра сертификации
2. COMODORSADomainValidationSecureServerCA.crt — сертификат самого Comodo
3. COMODORSADomainValidationSecureServerCA.crt — сертификат сервера Comodo, валидирующего домены
4. mydomain\_ru.crt — сертификат вашего сайта

Необходимо объединить сертификаты в цепочку начиная с сертификата Вашего сервера и заканчивая корневым

```
cat mydomain_ru.crt  
COMODORSADomainValidationSecureServerCA.crt  
COMODORSADomainValidationSecureServerCA.crt AddTrustExternalCARoot.crt >  
mydomain_ru.ca-bundle.crt
```

В файле конфигурации сервера server.config указываем файл секретного ключа

```
private-key-file = cakey.pk
```

указываем парольную фразу секретного ключа

```
private-key-file-passphrase = <SecretPhrase>
```

Указываем файл цепочки сертификатов

```
certificate-chain-file = mydomain_ru.ca-bundle.crt
```

В файле автоконфигурации клиента client\_autoconf.config указываем конечный корневой сертификат

```
vipole-server-certificate-file =  
AddTrustExternalCARoot.crt
```

обновляем параметры автоконфигурации клиента

```
vipole adm --domain <ваш домен> --dbname <имя_базы> ^
```

```
-o save_clients_autoconf  
--clients-autoconf-file client_autoconf.config
```

и перезапускаем Сервер Автоконфигурации.

## ПРИЛОЖЕНИЕ 2. Обновление Сервера

Если возникла необходимость обновить исполняемые файлы Сервера, необходимо выполнить следующие шаги.

1. Остановить все сервисы

```
service vipoleserver stop
```

2. обязательно сделать копию файлов конфигурации /etc/vipole/\*

3. Желательно сделать резервную копию базы данных

```
mongodump --host 127.0.0.1 --port 27017 -db vipole_server
```

бекап запишется в текущую папку в dump/

4. Выполнить обновление. (на вопросы об обновлении файлов конфигурации отвечать N )

```
dpkg -i vipoleserver-standalone_1.1.0_amd64.deb
```

5. Запустить сервисы

```
service vipoleserver start
```