

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ «ЈАТОВА»

Руководство по настройке. Часть 4.
Инструкция по безопасной настройке кластера на основе
компонента «jaDog»

643.72410666.00067-07 98 02-04

Листов 35

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

Настоящий документ является дополнением к существующему документу «Руководство по безопасности. Часть 27» и описывает рекомендации по безопасной настройке СУБД Jatoba.

Настоящее руководство предназначено для администраторов СУБД, специалистов по информационной безопасности и носит рекомендательный характер.

Степени важности примечаний, применяемые в документе:



Важная информация – указания, требующие особого внимания



Дополнительная информация – указания, позволяющие упростить работу с изделием



Все примеры в данном документе приведены для СУБД «Jatoba» версии ядра 6.x, для других версий все шаги выполняются аналогично, разница состоит в именах директорий.

Например, СУБД «Jatoba» версии 6.x по умолчанию устанавливается в директорию:

- ОС Windows – «C:\Program Files\GIS\Jatoba\6\bin»;
- ОС Linux – «/usr/jatoba-6/bin».



Важная информация

Для сертифицированной версии СУБД «Jatoba» поддерживается работа только на ОС, указанных в формуляре на поставку!

СОДЕРЖАНИЕ

1. Актуальность версий	4
2. Подключение	6
2.1. Минимизация количества пользователей/ролей, которым разрешено подключение к БД	6
2.2. Минимизация количества узлов/подсетей, с которых разрешено подключение к БД	7
2.3. Минимизация количества БД, к которым разрешено подключение	7
2.4. Минимизация количества УЗ, которым разрешено подключение к компоненту jaDog	8
2.5. Минимизация количества узлов/подсетей, с которых разрешено подключение к компоненту jaDog ..	9
2.6. Измените номер порта, используемый Jatoba на узлах кластера, на нестандартный	10
2.7. Измените номера портов, используемые jaDog, на нестандартные	11
3. Применение TLS/SSL	13
3.1. Создание СА ключей	13
3.2. Создание сертификатов сервера и администратора (пользователя)	13
3.2.1. Создание серверного сертификата	13
3.2.2. Создание пользовательского сертификата	14
3.2.3. Проверка сертификатов TLS при запуске сервиса jaDog	14
3.3. Настройка проверки SSL-сертификатов при подключении к узлам кластера	15
3.4. Настройка проверки SSL-сертификатов при подключении к компоненту jaDog	16
3.5. Настройка подключения jaDog к СУБД с помощью SSL-сертификатов	17
3.6. Настройка подключения к REST API с помощью SSL-сертификатов	18
3.7. Настройка минимальной версии протокола TLS не ниже рекомендованной	19
3.8. Настройка шаблона файла ответов с применением SSL-сертификатов	20
4. Аутентификация	24
4.1. Используйте надёжный метод аутентификации при подключении к узлам кластера	24
4.2. Используйте надёжный метод аутентификации при подключении к компоненту jaDog	24
4.3. Измените пароли всех технических и административных УЗ при переводе системы в эксплуатацию	25
5. Журналирование	26
5.1. Настройте параметры журналирования компонента jaDog	26
5.2. Настройте журналирование компонентом jaDog событий информационной безопасности	26
6. Хеширование и маскирование	28
6.1. Настройте стойкий алгоритм хеширования паролей в БД на узлах кластера	28
7. Контроль целостности	30
7.1. Установите расширение ja_csum и зафиксируйте эталонные контрольные суммы	30
8. Резервное копирование	31
8.1. Храните резервную копию файла ответов со структурой кластера в удалённом сетевом хранилище	31
Термины и определения	32
Перечень сокращений	34

1. АКТУАЛЬНОСТЬ ВЕРСИЙ

Используйте актуальную версию ПО СУБД «Jatoba» и компонентов, регулярно проверяйте выпуск обновлений.

Чем старше версия используемого программного обеспечения, тем больше времени было у злоумышленников на то, чтобы найти в ней уязвимости и «эксплойты» и, соответственно, тем уязвимее будет информационная система.

Чтобы защитить системы от потенциальных угроз, необходимо своевременно устанавливать обновления безопасности, закрывающие известные уязвимости в программном обеспечении.

Для проверки используемой версии СУБД «Jatoba» можно воспользоваться командой в терминале ОС:

Пример команды

```
/usr/jatoba-6/bin/postgres --version
```



Мажорные версии PostgreSQL и Jatoba соотносятся следующим образом:

- PostgreSQL 14 - Jatoba 4;
- PostgreSQL 15 - Jatoba 5;
- PostgreSQL 16 - Jatoba 6
- PostgreSQL 18 - Jatoba 18.

При подключении к СУБД, можно воспользоваться следующей функцией:

Пример функции

```
SELECT jatoba_version();
```

Для получения версий установленных компонентов необходимо подключиться к БД, в которую они установлены, и выполнить запрос к системному каталогу:

Пример запроса

```
SELECT extname, extversion FROM pg_catalog.pg_extension;
```

В терминале psql можно воспользоваться метакомандой:

Пример метакоманды

```
\dx
```

Процедуры обновления СУБД «Jatoba» и компонентов приведены в документе "Руководство по обновлению" или руководствах на компоненты.



Из соображений требований ИБ начиная с мажорной версии 4 компонента jaDog его процессы в ОС выполняются не от имени УЗ root, а от имени технологической УЗ, от которой запущен сервер СУБД «Jatoba».

2. ПОДКЛЮЧЕНИЕ

2.1. Минимизация количества пользователей/ролей, которым разрешено подключение к БД

Ограничение количества учётных записей, имеющих возможность подключения к СУБД, снижает шансы злоумышленника получить доступ к СУБД.

Возможность подключения к СУБД узла кластера настраивается в конфигурационном файле `pg_hba.conf`.

Наиболее безопасным вариантом будет указание в файле `pg_hba.conf` возможности подключения только технической учётной записи для взаимодействия jaDog с СУБД (`db_connection_settings:user`) и минимально необходимого списка учётных записей администраторов и технических учётных записей приложений.



В процессе развёртывания кластера создаётся (в случае развёртывания с помощью `jadog0` и файла ответов - автоматически, в случае ручного развёртывания - при вызове функции `grant_jadog_role_to_jadog_user('<username>')`) групповая роль `jadog_repl_acc`, являющаяся членом (с параметрами `SET` и `INHERIT`) встроенной роли `pg_read_all_stats` и имеющая атрибуты `INHERIT` и `REPLICATION`. Техническая учётная запись для взаимодействия JaDog с СУБД (примерах ниже - `jadog_user`) включается в роль `jadog_repl_acc`.

В приведённом ниже примере конфигурации настроена возможность подключения только для технической УЗ JaDog (`jadog_user`) и УЗ администратора инстанса СУБД (`db_admin`):

Пример конфигурационного файла `pg_hba.conf`

```
# TYPE      DATABASE  USER          ADDRESS          METHOD
# "local" is for Unix domain socket connections only
local      [db_name] db_admin              peer
hostssl    [db_name] jadog_user          127.0.0.1/32     cert
hostssl    [db_name] jadog_user          192.168.239.131/32 cert
hostssl    replication jadog_user          127.0.0.1/32     cert
```

hostssl replication jadow_user	192.168.239.131/32	cert
--------------------------------	--------------------	------

2.2. Минимизация количества узлов/подсетей, с которых разрешено подключение к БД

В дополнение к предыдущему пункту, ограничение в конфигурационном файле `pg_hba.conf` количества адресов/подсетей, с которых возможно подключение к СУБД на каждом узле кластера, позволяет ещё сильнее уменьшить шансы злоумышленника на проникновение в СУБД. Даже в случае получения данных одной из учётных записей злоумышленнику придётся дополнительно получить контроль над хостом в определённой подсети, чтобы подключиться к СУБД.

По возможности не используйте значение 'all' в поле ADDRESS конфигурационного файла `pg_hba.conf`, такая настройка позволит злоумышленнику попытаться подключиться с любого хоста, над которым у него есть контроль. Вместо этого лучше использовать отдельные адреса узлов, с которых необходимо подключение к экземпляру СУБД, либо ограниченные подсети.

В приведённом ниже примере конфигурации настроена возможность подключения технической УЗ JaDog (`jadow_user`) только с `localhost` и с внешнего адреса самого узла.

Пример конфигурационного файла `pg_hba.conf`

#	TYPE	DATABASE	USER	ADDRESS	METHOD
	hostssl	[db_name]	jadow_user	127.0.0.1/32	cert
	hostssl	[db_name]	jadow_user	192.168.239.131/32	cert
	hostssl replication		jadow_user	127.0.0.1/32	cert
	hostssl replication		jadow_user	192.168.239.131/32	cert

2.3. Минимизация количества БД, к которым разрешено подключение

В дополнение к предыдущим пунктам, ограничение в конфигурационном файле `pg_hba.conf` количества баз данных, к которым возможно подключение, дополнительно снижает шансы злоумышленника на проникновение в СУБД. Даже в случае получения данных одной из учётных записей и обретения контроля над одним из хостов в определённой подсети злоумышленнику придётся подобрать название базы для подключения.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

По возможности не используйте значение 'all' в поле DATABASE конфигурационного файла pg_hba.conf, такая настройка позволит злоумышленнику попытаться подключиться данным именем УЗ к любой базе данных.



Данная рекомендация не относится к техническому пользователю для взаимодействия JaDog с СУБД (db_connection_settings:user), ему как раз нужна возможность подключения ко всем базам данных.

В приведённом ниже примере конфигурации настроена возможность подключения технической УЗ приложения (appuser) только к БД этого приложения (appdb).

Пример конфигурационного файла pg_hba.conf

#	TYPE	DATABASE	USER	ADDRESS	METHOD
hostssl		appdb	appuser	192.168.239.101/32	cert

2.4. Минимизация количества УЗ, которым разрешено подключение к компоненту jaDog

Ограничение количества учётных записей, имеющих возможность подключения к компоненту JaDog, снижает шансы злоумышленника получить контроль над кластером. Возможность подключения к компоненту JaDog узла кластера настраивается в конфигурационном файле jadog_hba.cfg (см. документ «Руководство по настройке. Часть 1. Управление режимом работы узлов кластера. Компонент «jaDog»).

В случае использования парольной аутентификации между узлами кластера наиболее безопасным вариантом будет указание в файле конфигурации jadog_hba.cfg возможности подключения только технической учётной записи для взаимодействия с другими JaDog-сервисами (main:interconnect_user) и минимально необходимого списка учётных записей администраторов.

В приведённом ниже примере конфигурации настроена возможность подключения только УЗ для взаимодействия с другими JaDog-сервисами (admin) и УЗ администратора (administrator).

Пример конфигурационного файла `jadog_hba.cfg`

#	USER	ADDRESS	METHOD
	admin	127.0.0.1/32	sha-256
	admin	192.168.239.0/24	sha-256
	administrator	192.168.239.0/24	sha-256

В случае же использования аутентификации между узлами кластера с использованием SSL-сертификатов помимо озвученных выше УЗ в файле конфигурации `jadog_hba.cfg` на каждом узле кластера обязательно должна быть указана возможность подключения для пользователей, имя которых совпадает с полем CN серверных сертификатов остальных узлов кластера (т.к. в таком случае компонент JaDog одного узла кластера при подключении будет представляться другому узлу именем, указанным в поле CN серверного сертификата, и при отсутствии таких записей в `jadog_hba.cfg` межузловое взаимодействие будет невозможно).

В приведённом ниже примере конфигурации настроена возможность подключения только УЗ для взаимодействия с другими JaDog-сервисами (admin) и CN серверных сертификатов других узлов кластера (jadog-node2 и jadog-node3).

Пример конфигурационного файла `jadog_hba.cfg`

#	USER	ADDRESS	METHOD
	admin	127.0.0.1/32	ssl
	admin	192.168.239.0/24	ssl
	jadog-node2	192.168.239.132/32	ssl
	jadog-node3	192.168.239.133/32	ssl

По возможности не используйте значение 'all' в поле USER конфигурационного файла `jadog_hba.cfg` – такая настройка увеличивает для злоумышленника шанс подобрать имя учётной записи методом перебора.

2.5. Минимизация количества узлов/подсетей, с которых разрешено подключение к компоненту jaDog

В дополнение к предыдущему пункту, ограничение в конфигурационном файле `jadog_hba.cfg` количества адресов/подсетей, с которых возможно подключение к компоненту

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

jaDog на каждом узле кластера позволяет ещё сильнее уменьшить шансы злоумышленника на получение контроля над кластером. Даже в случае получения данных одной из учётных записей злоумышленнику придётся дополнительно получить контроль над хостом в определённой подсети, чтобы подключиться к кластеру (см. документ "Руководство по настройке. Часть 1. Управление режимом работы узлов кластера. Компонент «jaDog»" п. 4.2.1.).

По возможности не используйте значение 'all' в поле ADDRESS конфигурационного файла jadog_hba.cfg, такая настройка позволит злоумышленнику попытаться подключиться с любого хоста, над которым у него есть контроль. Вместо этого лучше использовать отдельные адреса узлов, с которых необходимо подключение к компоненту jaDog, либо ограниченные подсети.

В приведённом ниже примере конфигурации настроена возможность подключения УЗ для взаимодействия с другими jaDog-сервисами (admin) только с localhost и подсети кластера, а CN серверных сертификатов других узлов кластера (jadog-node2 и jadog-node3) - только с адресов соответствующих узлов.

Пример конфигурационного файла jadog_hba.cfg

#	USER	ADDRESS	METHOD
	admin	127.0.0.1/32	ssl
	admin	192.168.239.0/24	ssl
	jadog-node2	192.168.239.132/32	ssl
	jadog-node3	192.168.239.133/32	ssl

2.6. Измените номер порта, используемый JatoBa на узлах кластера, на нестандартный

Изменение стандартного номера порта на случайный усложняет для злоумышленника проведение автоматизированных атак (с помощью ботов или автоматизированных скриптов).

Для изменения номера порта откройте конфигурационный файл postgresql.conf и измените значение параметра port:

Пример конфигурации

```
port = 5432
```

Измените стандартное значение 5432 на любой другой номер порта, не задействованный на этом сервере.

Значение параметра port также можно изменить при помощи команды ALTER SYSTEM:

Пример запроса

```
ALTER SYSTEM SET port = 5433;
```

После изменения значения параметра port нужно перезагрузить экземпляр СУБД (т.к. параметр имеет контекст postmaster).



Обратите внимание на то, что после изменения номера порта в конфигурации СУБД «Jatoba» нужно указать этот порт в параметрах подключения jaDog к СУБД «Jatoba».

Для того, чтобы указать компоненту jaDog на каком порту работает экземпляр Jatoba, можно воспользоваться командой утилиты jadog_ctl:

Пример команды

```
set parameter db_connection_settings:port = '5433'
```

2.7. Измените номера портов, используемые jaDog, на нестандартные

Аналогично предыдущему пункту, стандартные порты, используемые компонентом jaDog для своей работы, могут быть использованы злоумышленником в автоматизированной атаке, поэтому их также следует изменить на любые не задействованные на данном сервере.

Для изменения портов, используемых jaDog, а также адреса, который прослушивает REST API, можно воспользоваться командами утилиты jadog_ctl:

Пример команд

```
set parameter main:port = '12345'  
set parameter main:user_interface_port = '54321'  
set parameter rest_api:listen_port = '54443'  
set parameter rest_api:listen_address = '127.0.0.1'
```

Также можно ознакомиться с параметрами, указанными в секциях `main` и `rest_api` в файле конфигурации `jadog.yml`.

Пример конфигурации

```
main:  
  port: 12345  
  user_interface_port: 54321  
rest_api:  
  rest_api_listen_port: 54443  
  rest_api_listen_address: 127.0.0.1
```

3. ПРИМЕНЕНИЕ TLS/SSL



Использование самоподписанных сертификатов TLS (SSL) допустимо только при проектировании и отладке кластера. В целях обеспечения информационной безопасности в промышленной эксплуатации допускается использование только сертификатов, выданных удостоверяющим центром (CA).



В конфигурациях компонентов СУБД «Jatoba» рекомендуется применение сертификатов TLS (SSL) версии 3.

Применение сертификатов TLS (SSL) для компонента «jaDog» приводится в документе «Руководство по безопасности СУБД Jatoba» 643.72410666.00067-07 97 01-27.

3.1. Создание CA ключей

```
openssl req -x509 -days 365 -newkey rsa:4096 -sha256 -nodes -  
keyout ca-key.crt -out ca-cert.crt -subj "/C=RU/ST=SPB/L=Saint-  
Petersburg /CN=Self-signed CA"
```

3.2. Создание сертификатов сервера и администратора (пользователя)

При подготовке сертификатов в качестве значений «CN=» может указываться название сервера (hostname) или имя (логин) администратора, в зависимости от назначения сертификата. В таком случае сертификаты подразделяются на серверные и клиентские.

3.2.1. Создание серверного сертификата

Серверный сертификат SSL создается следующим образом:

```
openssl req -newkey rsa:4096 -nodes -keyout server1-key.crt -  
out server1-req.crt -subj "/extendedKeyUsage=serverAuth  
/subjectAltName=DNS:jatoba-10,DNS:jatoba-  
11,DNS:localhost,IP:127.0.0.1,IP:192.168.72.10,IP:192.168.72.11  
/CN=jatoba-11"
```

Аналогично сертификаты создаются локально на всех остальных узлах кластера:

```
openssl req -newkey rsa:4096 -nodes -keyout server3-key.crt -  
out server3-req.crt -subj "/extendedKeyUsage=serverAuth  
/subjectAltName=DNS:jatoba-  
10,DNS:DNS_name_server,DNS:localhost,IP:127.0.0.1,  
IP:public_address,IP:IP_address_server /CN=server_name_N"
```

Подписание выпущенного сертификата для узла кластера:

```
openssl x509 -req -in server1-req.crt -days 365 -CA root.crt -  
CAkey ca-key.crt -CAcreateserial -extfile openssl.cnf -  
extensions v3_req -out server1-cert.crt
```

Аналогично сертификаты подписываются для всех остальных узлов кластера:

```
openssl x509 -req -in server3-req.crt -days 365 -CA root.crt -  
CAkey ca-key.crt -CAcreateserial -extfile openssl.cnf -  
extensions v3_req -out serverN-cert.crt
```

Проверка корректности выпущенного сертификата узла кластера:

```
openssl x509 -in server1-cert.crt -noout -text
```

3.2.2. Создание пользовательского сертификата

Создание сертификата для администратора (пользователя) кластера:

```
openssl req -newkey rsa:4096 -nodes -keyout user_name-key.crt -  
out user_name-req.crt -subj "/extendedKeyUsage=clientAuth  
/CN=user_name"
```

Подписание выпущенного сертификата для администратора (пользователя) кластера:

```
openssl x509 -req -in user-req.crt -days 365 -CA ca-cert.crt -  
CAkey ca-key.crt -CAcreateserial -extfile openssl.cnf -  
extensions v3_req -out user_name-cert.crt
```

Значение user_name может быть персонифицированной УЗ администратора кластера, либо технологической УЗ (interconnect_user, jadog_user).

Проверка корректности выпущенного сертификата администратора (пользователя) кластера:

```
openssl x509 -in user_name-cert.crt -noout -text
```

3.2.3. Проверка сертификатов TLS при запуске сервиса jaDog

При запуске сервис jaDog проверяет серверный TLS-сертификат. Для успешного запуска необходимо, чтобы имя хоста, с которого запускается сервис, совпадало с одним из допустимых имён в сертификате.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

Источники проверки имён в сертификате:

- Common Name (CN) — основное имя хоста;
- Subject Alternative Name (SAN) — список альтернативных имён.



Сервис последовательно проверяет оба поля (сначала SAN, затем CN). Таким образом, в случае если хотя бы одно из полей заполнено не корректно служба компонента «jaDog» не будет запущена.

Формат допустимых имён в сертификате:

- имя должно быть полным и явным (FQDN или IP-адрес);
- использование шаблонов с подстановочными знаками (wildcards, например *.example.com) не поддерживается и приведёт к отказу в запуске службы компонента «jaDog».

Сценарии, приводящие к отказу в запуске службы компонента «jaDog»:

- недействительный или просроченный серверный сертификат;
- имя хоста сервера отсутствует как в поле CN, так и в списке SAN сертификата;
- параметр запуска -h содержит название узла или IP-адрес, которых нет в CN или списке SAN сертификата;
- в списке имён (CN или SAN) сертификата обнаружен шаблон с wildcard (*).

3.3. Настройка проверки SSL-сертификатов при подключении к узлам кластера

Наиболее безопасным методом аутентификации при подключении к узлам кластера является метод с использованием сертификатов SSL, предполагающий проверку подлинности как клиента сервером, так и сервера клиентом, а также задействующий шифрование данных, передаваемых между клиентом и сервером. Применение этого метода аутентификации позволяет защититься от атак типа MITM (Man-in-the-Middle).

Таблица 3.1 – Названия сертификатов и места их хранения

Файл	Путь/настройка конфигурации
	jadog.yml – конфигурационный файл
	/var/lib/jatoba/ssl_jadog/ - каталог хранения сертификатов
root.crt	db_connection_settings:ssl_ca_file: /var/lib/jatoba/ssl_jatoba/root.crt
jadog_service.crt	db_connection_settings:ssl_cert_file: /var/lib/jatoba/ssl_jadog/jadog_service.crt

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

Файл	Путь/настройка конфигурации
jadog_service.key	db_connection_settings:ssl_key_file: /var/lib/jatoba/ssl_jadog/jadog_service.key
root.crt	tls:ca_file: /var/lib/jatoba/ssl_jatoba/root.crt
interconnect.crt	tls:cert_file: /var/lib/jatoba/ssl_jadog/interconnect.crt
interconnect.key	tls:key_file: /var/lib/jatoba/ssl_jadog/interconnect.key
root.crt	rest_api:ca_file: /var/lib/jatoba/ssl_jatoba/root.crt
rest_api_server.crt	rest_api:cert_file: /var/lib/jatoba/ssl_jadog/server.crt
rest_api_server.key	rest_api: key_file: /var/lib/jatoba/ssl_jadog/server.key

Для примера настройки см. документ "Руководство по безопасности. Часть 27." п. 7.1. При добавлении записей в конфигурационный файл pg_hba.conf на узлах кластера руководствуйтесь принципами, описанными в п. 4.1 - 4.3 настоящего руководства.

Пример указания сертификатов в конфигурационном файле postgresql.conf.

Пример конфигурационного файла postgresql.conf

```
ssl = on
ssl_ca_file = '/var/lib/jatoba/ssl_jatoba/root.crt'
ssl_cert_file = '/var/lib/jatoba/ssl_jatoba/server.crt'
ssl_key_file = '/var/lib/jatoba/ssl_jatoba/server.key'
```

Пример настройки проверки сертификатов при подключении технической УЗ (jadog_user) в конфигурационном файле pg_hba.conf.

Пример конфигурационного файла pg_hba.conf

```
# TYPE      DATABASE     USER        ADDRESS          METHOD
hostssl replication jadog_user  127.0.0.1/32    cert clientcert=verify-
full
hostssl replication jadog_user  192.168.239.131/32 cert clientcert=verify-
full
```

3.4. Настройка проверки SSL-сертификатов при подключении к компоненту jaDog

Для примера настройки см. документ «Руководство по безопасности. Часть 27.» п. 7.1. При добавлении записей в конфигурационный файл jadog_hba.cfg на узлах кластера руководствуйтесь принципами, описанными в п. 2.1 - 2.5 данного документа.

Для указания сертификатов, используемых jaDog для соединения между узлами, можно воспользоваться командами утилиты jadog_ctl.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

Пример команд

```
set parameter tls:tls = 'true'
set parameter tls:ca_file =
'/var/lib/jatoba/ssl_jatoba/root.crt'
set parameter tls:cert_file =
'/var/lib/jatoba/ssl_jadog/interconnect.crt'
set parameter tls:key_file =
'/var/lib/jatoba/ssl_jadog/interconnect.key'
```

Также можно ознакомиться с параметрами, указанными в секции `tls` в файле конфигурации `jadog.yml`

Пример конфигурации

```
tls:
  tls: true
  ca_file: /var/lib/jatoba/ssl_jatoba/root.crt
  cert_file: /var/lib/jatoba/ssl_jadog/interconnect.crt
  key_file: /var/lib/jatoba/ssl_jadog/interconnect.key
```

Пример настройки проверки сертификатов при подключении УЗ для взаимодействия с другими `jadog`-сервисами (`admin`) и CN серверных сертификатов других узлов кластера (`jadog-node2` и `jadog-node3`) в конфигурационном файле `jadog_hba.cfg`.

Пример конфигурационного файла `jadog_hba.cfg`

#	USER	ADDRESS	METHOD
	admin	127.0.0.1/32	ssl
	admin	192.168.239.0/24	ssl
	jadog-node2	192.168.239.132/32	ssl
	jadog-node3	192.168.239.133/32	ssl

3.5. Настройка подключения jaDog к СУБД с помощью SSL-сертификатов

Для примера настройки см. документ "Руководство по безопасности. Часть 27." п. 7.2.

Для указания сертификатов, используемых jaDog для подключения к СУБД можно воспользоваться командами утилиты `jadog_ctl`.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

Пример команд

```
set parameter db_connection:auth_method = 'ssl'  
set parameter db_connection_settings:ssl_mode = 'verify-full'  
set parameter db_connection_settings:ssl_ca_file =  
'/var/lib/jatoba/ssl_jatoba/root.crt'  
set parameter db_connection_settings:ssl_cert_file =  
'/var/lib/jatoba/ssl_jadog/jadog_service.crt'  
set parameter db_connection_settings:ssl_key_file =  
'/var/lib/jatoba/ssl_jadog/jadog_service.key'
```

Пример конфигурации

```
db_connection:  
auth_method: ssl
```

А также изменить пути до сертификатов, указанные в строке подключения в секции

```
db_connection_settings: → ssl_mode  
db_connection_settings: → ssl_ca_file  
db_connection_settings: → ssl_cert_file  
db_connection_settings: → ssl_key_file.
```

3.6. Настройка подключения к REST API с помощью SSL-сертификатов

Для примера настройки см. документ «Руководство по безопасности. Часть 27.» п. 7.3.1.

Для указания сертификатов, используемых jaDog при подключении к REST API можно воспользоваться командами утилиты jadog_ctl.

Пример команд

```
set parameter rest_api:ca_file =  
'/var/lib/jatoba/ssl_jadog/root.crt'  
set parameter rest_api:cert_file =  
'/var/lib/jatoba/ssl_jadog/jadog_user.crt'
```

```
set parameter rest_api:key_file =  
'/var/lib/jatoba/ssl_jadog/jadog_user.key'
```

Также можно ознакомиться с параметрами, указанными в секции `rest_api` в файле конфигурации `jadog.yml`.

Пример конфигурации

```
rest_api:  
  ca_file: /var/lib/jatoba/ssl_jadog/root.crt  
  cert_file: /var/lib/jatoba/ssl_jadog/server.crt  
  key_file: /var/lib/jatoba/ssl_jadog/server.key
```

3.7. Настройка минимальной версии протокола TLS не ниже рекомендованной

Версии протокола TLS ниже чем TLSv1.2 имеют известные уязвимости, считаются устаревшими и не рекомендуются к использованию без крайней на то необходимости.

Для ограничения минимальной версии протокола TLS откройте конфигурационный файл `postgresql.conf` и измените значение параметра `ssl_min_protocol_version`.

Пример конфигурации

```
ssl_min_protocol_version = 'TLSv1.2'
```

Значение параметра `ssl_min_protocol_version` также можно изменить при помощи команды `ALTER SYSTEM`.

Пример запроса

```
ALTER SYSTEM SET ssl_min_protocol_version = 'TLSv1.2';
```

После изменения значения параметра необходимо перечитать конфигурацию инстанса СУБД.

Пример функции

```
SELECT pg_reload_conf();
```

Также можно перечитать конфигурацию с использованием команды утилиты `jadog_ctl`.

Пример команды

```
reload dbs on node 'node_name'
```

Есть возможность перечитать конфигурацию инстанса СУБД сразу на всех узлах кластера с использованием команды утилиты `jadog_ctl`.

Пример команды

```
reload dbs on cluster
```

3.8. Настройка шаблона файла ответов с применением SSL-сертификатов

При использовании автоматизированного развертывания кластера с применением SSL-сертификатов необходимые параметры указываются непосредственно в шаблоне файла ответов.

Подготовка самоподписанных SSL-сертификатов для проектирования, отладки и непромышленной эксплуатации кластера описана в третьей части документа «Краткое руководство по настройке. Компонент «jaDog» 643.72410666.00067-07 98 02-03.

Шаблоны файлов ответов для автоматизированного развертывания кластера располагаются в директории `/usr/jatoba-<ver>/share/doc/jadog/clusters_kits`, где `<ver>` - номер версии СУБД «Jatoba». Шаблоны с применением SSL-сертификатов содержат в названии суффикс «ssl».

Применение SSL-сертификатов в компоненте «jaDog» подразумевает подготовку трех типов SSL-сертификатов:

- для подключения при инициализации БД (здесь и далее называются `postgres`), расположение в директории `/var/lib/jatoba/ssl_jatoba`;
- для подключения к БД (здесь и далее имеют префикс `jadog_server`), расположение в директории `/var/lib/jatoba/ssl_jatoba`;
- для межузлового взаимодействия (здесь и далее имеют префикс `jadog_user`), расположение в директории `/var/lib/jatoba/ssl_jadog`.

Для межузлового взаимодействия в компоненте «jaDog» применяется специализированный пользователь с типом «interconnect user». По тексту инструкции для

примера таким пользователем определен `jadog_user`. Описание процедур создания пользователя `jadog_user` приводится в первой части документа «Управление режимом работы узлов кластера. Компонент «jaDog» 643.72410666.00067-07 98 02-01.

В секции `default_cluster_params:db_init_conn_string` параметров первоначальной инициализации СУБД определяются пути к директориям, которые содержат необходимые SSL-сертификаты.

```
default_cluster_params:
    db_init_conn_string: host=127.0.0.1 port=5432 dbname=postgres
    user=postgres sslmode=verify-full
    sslrootcert=/var/lib/jatoba/ssl_jatoba/root.crt
    sslcert=/var/lib/jatoba/ssl_jatoba/postgres.crt
    sslkey=/var/lib/jatoba/ssl_jatoba/postgres.key

    initdb:
        initdb_options: "--locale=ru_RU.utf8 --encoding=UTF-8 --set ssl='on'
--set ssl_ca_file='/var/lib/jatoba/ssl_jatoba/root.crt' --set
ssl_cert_file='/var/lib/jatoba/ssl_jatoba/jatoba_server.crt' --set
ssl_key_file='/var/lib/jatoba/ssl_jatoba/jatoba_server.key'"
```

В секции `pg_hba.conf` шаблона необходимо указать параметры подключения «interconnect user» к базам данных:

```
pg_hba.conf:  # Параметры будут установлены при формировании кластера в
файл pg_hba.conf

- local      all          postgres                                trust
- hostssl    all          all          127.0.0.1/32          cert
clientcert=verify-full
- hostssl    all          all          192.168.72.0/24        cert
clientcert=verify-full
- hostssl    replication  jadog_user    127.0.0.1/32          cert
clientcert=verify-full
- hostssl    replication  jadog_user    192.168.72.0/24        cert
clientcert=verify-full
```

В секции `cluster_settings:jadog_users` для каждого из пользователей необходимо указать метод аутентификации, в данном случае `ssl`. Например:

```
jadog_users:
- name: admin
  address: all
```

```
method: ssl
- name: jadow_user
address: all
method: ssl
```

В секции `cluster_settings:default_node_params:main` в параметре `interconnect_user` необходимо указать название учетной записи, используемой для межузлового взаимодействия, в данном примере это `jadow_user`. В данной секции также перечисляются пути к директориям, которые содержат SSL-сертификаты для межузлового взаимодействия. Например:

```
default_node_params:
  main:
    interconnect_user:
      name: jadow_user
      ca_file: /var/lib/jatoba/ssl_jadow/root.crt
      cert_file: /var/lib/jatoba/ssl_jadow/jadow_user.crt
      key_file: /var/lib/jatoba/ssl_jadow/jadow_user.key
```

В секции `cluster_settings:default_node_params:tls` необходимо указать пути к директориям, в которых находятся сертификаты SSL:

```
tls:
  tls: true
  ca_file: /var/lib/jatoba/ssl_jadow/root.crt
  cert_file: /var/lib/jatoba/ssl_jadow/jadow_service.crt
  key_file: /var/lib/jatoba/ssl_jadow/jadow_service.key
```

В секции `cluster_settings:default_node_params:db_connection` в параметре `auth_method` необходимо указать метод аутентификации в БД, в данном случае `ssl`:

```
db_connection:
  auth_method: ssl
```

В секции `cluster_settings:default_node_params:db_connection_settings` необходимо указать пути к директориям, в которых находятся сертификаты SSL:

```
db_connection_settings:
  user: jalog_user
  ssl_ca_file: /var/lib/jatoba/ssl_jatoba/root.crt
  ssl_cert_file: /var/lib/jatoba/ssl_jatoba/jalog_user.crt
  ssl_key_file: /var/lib/jatoba/ssl_jatoba/jalog_user.key
  ssl_mode: verify-full
```

4. АУТЕНТИФИКАЦИЯ

4.1. Используйте надёжный метод аутентификации при подключении к узлам кластера

Рекомендованным методом аутентификации при подключении к СУБД на узлах кластера является аутентификация с проверкой SSL-сертификатов - cert (см. п. 0 данного документа). Тем не менее, в случае, если информационная система не поддерживает аутентификацию по сертификату, следует настроить надёжный метод аутентификации в конфигурационном файле `pg_hba.conf` узлов кластера.

Методы аутентификации `trust`, `password`, `ident`, `peer` и `md5` не являются надёжными и не рекомендуются к использованию в промышленной среде. Вместо них рекомендуется использовать один из следующих методов, поддерживаемых JaToba: `scram-sha-256`, `gss`, `sspi`, `ldap`, `radius`, `pam`.

В приведённом ниже примере конфигурации настроена возможность подключения технической УЗ (`jadog_user`) с применением метода аутентификации `scram-sha-256`.

Пример конфигурационного файла `pg_hba.conf`

#	TYPE	DATABASE	USER	ADDRESS	METHOD
host		[db_name]	jadog_user	127.0.0.1/32	scram-sha-256
host		[db_name]	jadog_user	192.168.239.131/32	scram-sha-256
host		replication	jadog_user	127.0.0.1/32	scram-sha-256
host		replication	jadog_user	192.168.239.131/32	scram-sha-256

4.2. Используйте надёжный метод аутентификации при подключении к компоненту jaDog

Рекомендованным методом аутентификации при подключении к компоненту jaDog является аутентификация с проверкой SSL-сертификатов - `ssl` (см. п. 3.4 данного документа). В случае, если информационная система не поддерживает аутентификацию по сертификату, следует настроить аутентификацию с помощью метода `scram-sha-256`.

В приведённом ниже примере конфигурации настроена аутентификация УЗ для взаимодействия с другими jaDog-сервисами (`admin`) и УЗ администратора (`administrator`) с применением метода `scram-sha-256`.

Пример конфигурационного файла jadog_hba.cfg

#	USER	ADDRESS	METHOD
	admin	127.0.0.1/32	sha-256
	admin	192.168.239.0/24	sha-256
	administrator	192.168.239.0/24	sha-256

4.3. Измените пароли всех технических и административных УЗ при переводе системы в эксплуатацию

В случае использования парольной аутентификации вместо SSL в процессе настройки компонента jaDog и развёртывания кластера (вручную или с помощью jadog0) в интерфейсе jaDog и файлах ответов указываются пароли технических и административных учетных записей. Эти пароли могут быть скомпрометированы в процессе настройки, поэтому при переводе системы в промышленную эксплуатацию они должны быть изменены.

5. ЖУРНАЛИРОВАНИЕ

5.1. Настройте параметры журналирования компонента jaDog

Помимо журналирования событий инстанса СУБД также нужно настроить журналирование событий компонента jaDog.

Для установки параметров журналирования jaDog можно воспользоваться командами утилиты jadog_ctl.

Пример команд

```
set parameter log:path = '/usr/jatoba-6/var/log/jadog'  
set parameter log:file = 'true'  
set parameter log:mode = '0600'  
set parameter log:file_name = 'jadog-%a'  
set parameter log:type = 'csv, security.csv'  
set parameter log:level = 'info'
```

Также можно ознакомиться с параметрами, указанными в секции log в файле конфигурации jadog.yml.

Пример конфигурации

```
log:  
  path: /usr/jatoba-6/var/log/jadog  
  file: true  
  mode: 0600  
  file_name: jadog-%a  
  level: info  
  type: csv, security.csv
```

5.2. Настройте журналирование компонентом jaDog событий информационной безопасности

Не менее важно настроить журналирование событий ИБ компонента jaDog.

Для установки параметров журналирования событий ИБ jaDog можно воспользоваться командами утилиты jadog_ctl:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

Пример команд

```
set parameter log:type = 'csv, security.csv'  
set parameter security_log:path = '/usr/jatoba-6/var/log/jadog'  
set parameter security_log:file_mode = '0600'  
set parameter security_log:file_name = 'security_jadog-%a'
```



Обратите внимание на то, что тип журналирования событий ИБ указывается в значении параметра type в секции log с помощью конструкции типа '<...>', security.<формат_журналирования>' (например, security.csv). Если его не указать - журнал событий ИБ не будет создан.

Также можно ознакомиться с параметрами, указанными в секциях log и security_log в файле конфигурации jadog.yml.

Пример конфигурации

```
log:  
  type: csv, security.csv  
security_log:  
  path: /usr/jatoba-6/var/log/jadog  
  file_mode: 0600  
  file_name: security_jadog-%a
```

6. ХЕШИРОВАНИЕ И МАСКИРОВАНИЕ

6.1. Настройте стойкий алгоритм хеширования паролей в БД на узлах кластера

В случае использования парольной аутентификации пароль технической УЗ (jadog_user) должен храниться в СУБД в захешированном стойким алгоритмом (с использованием случайной соли) виде. Тогда даже в случае получения злоумышленником хеша пароля использование им радужных таблиц для определения исходного пароля не принесёт результата.

СУБД «Jatoba» поддерживает два алгоритма хеширования: md5 (менее стойкий) и scram-sha-256 (более стойкий).

Для установки стойкого алгоритма хеширования паролей откройте конфигурационный файл postgresql.conf и измените значение параметра password_encryption.

Пример конфигурации

```
password_encryption = scram-sha-256
```

Значение параметра password_encryption также можно изменить при помощи команды ALTER SYSTEM.

Пример запроса

```
ALTER SYSTEM SET password_encryption = 'scram-sha-256';
```

После изменения значения параметра необходимо перечитать конфигурацию инстанса СУБД.

Пример функции

```
SELECT pg_reload_conf();
```

Также можно перечитать конфигурацию с использованием команды утилиты jadog_ctl.

Пример команды

```
reload dbs on node 'node_name'
```

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

Есть возможность перечитать конфигурацию инстанса СУБД сразу на всех узлах кластера с использованием команды утилиты `jadog_ctl`.

Пример команды

```
reload dbs on cluster
```



Обратите внимание на то, что после изменения значения параметра и перечитывания конфигурации следует изменить пароль технической УЗ (`jadog_user`) для того, чтобы он перехешировался указанным алгоритмом.

7. КОНТРОЛЬ ЦЕЛОСТНОСТИ

7.1. Установите расширение `ja_csum` и зафиксируйте эталонные контрольные суммы

Для предотвращения несанкционированного изменения шаблонных баз данных, библиотек, бинарных файлов и файлов конфигурации рекомендуется зафиксировать их контрольные суммы и постоянно наблюдать за фактом их изменения. Для реализации этой функции требуется установить расширение `ja_csum`.

Для примера установки и применения расширения `ja_csum` см. документ «Руководство по настройке. Часть 14. Контроль целостности. Компонент `ja_CSum`».



В случае ручной фиксации эталонных контрольных сумм файлов компонента JaDog неизменными в процессе работы компонента файлами считаются:

- библиотеки;
- бинарные файлы;
- файл конфигурации `jadog.yml`;
- файл конфигурации `jadog_hba.cfg`;
- файл конфигурации `users.yml`.

Фиксация контрольных сумм других файлов расширения может привести к непредвиденным блокировкам УЗ в продуктовой среде!



Обратите внимание на то, что пакет компонента `ja_csum` должен быть установлен на всех хостах до конфигурирования кластера. В случае, если на одном узле кластера расширение будет установлено, а на другом - нет, на узле с отсутствующим пакетом расширения демон Jatoba в определённый момент зафиксирует ошибку и будет остановлен.



Обратите внимание на то, что для функционирования на всех узлах кластера файлы эталонных контрольных сумм должны быть скопированы на все узлы кластера.

8. РЕЗЕРВНОЕ КОПИРОВАНИЕ

8.1. Храните резервную копию файла ответов со структурой кластера в удалённом сетевом хранилище

Компонент «jaDog» имеет встроенную возможность выгрузки структуры кластера в файл, который впоследствии может быть использован как файл ответа для восстановления состояния кластера в случае вывода злоумышленником из строя одного или нескольких узлов кластера (см. документ "Руководство по настройке. Часть 1. Управление режимом работы узлов кластера. Компонент «jaDog»" п. 3.7.).

При планировании политики создания и хранения резервных копий рекомендуется использовать для сохранения резервных файлов ответов не локальную директорию сервера СУБД, а удалённое сетевое хранилище, физически расположенное в другом ЦОД. Тогда в случае вывода злоумышленником из строя сегмента серверной инфраструктуры файлы ответов останутся доступны для восстановления.



Обратите внимание на то, что для функционирования на всех узлах кластера директория, находящаяся на удалённом файловом сервере, должна быть подключена на всех узлах кластера.

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Аутентификационная информация — информация, используемая при аутентификации субъекта доступа или объекта доступа.

Аутентификация – действия по проверке подлинности субъекта доступа и/или объекта доступа, а также по проверке принадлежности субъекту доступа и/или объекту доступа предъявленного идентификатора доступа и аутентификационной информации (ГОСТ Р 58833-2020).

Администратор СУБД – субъект доступа, выполняющий административные функции в СУБД и наделенный правами:

- создавать учетные записи пользователей системы управления базами данных;
- модифицировать, блокировать и удалять учетные записи пользователей системы управления базами данных;
- назначать права доступа пользователям системы управления базами данных к объектам доступа системы управления базами данных;
- управлять конфигурацией системы управления базами данных;
- создавать, подключать базы данных.

Администратор СУБД имеет атрибут SUPERUSER и/или обладает системной учетной записью «postgres».

Администратор БД – субъект доступа, выполняющий административные функции в БД и наделенный правами:

- создавать учетные записи пользователей базы данных;
- модифицировать, блокировать и удалять учетные записи пользователей базы данных;
- управлять конфигурацией базы данных;
- назначать права доступа пользователям базы данных (пользователей информационной системы) к объектам доступа базы данных;

- создавать резервные копии базы данных и восстанавливать базу данных из резервной копии;
- создавать, модифицировать и удалять процедуры (программный код), хранимые в базе данных.

Администратор БД имеет атрибут CREATEROLE, и возможные атрибуты BYPASSRLS, REPLICATION, а также прочие системные привилегии относительно БД, кроме атрибута CREATEDB.

Безусловная блокировка пользователя – это ограничение пользователя в возможности устанавливать новую сессию с СУБД. Безусловная блокировка имеет приоритет над ограничениями, накладываемыми парольными политикам (блокировка вследствие истечения срока действия пароля, временные блокировки при исчерпании попыток ввода пароля и т.п.), применяется независимо от них и не зависит от применяемого метода аутентификации пользователей. Снятие безусловной блокировки не снимает блокировок по парольным политикам и наоборот.

Завершение сессии пользователя – принудительное завершение открытой сессии пользователя с БД/СУБД в заданном режиме.

Пользователь БД - субъект доступа, имеющий доступ к ограниченному перечню БД и объектов БД. Имеющий следующий набор привилегий:

- создавать и манипулировать объектами доступа БД (таблица, запись или столбец, поле, представление и иные объекты доступа);
- выполнять процедуры (программный код), хранимые в БД.

Пользователь БД имеет обязательный атрибут LOGIN.

Пользователь СУБД – см. «Пользователь БД». Для СУБД эти понятия идентичны. СУБД не разграничивает пользователей по отдельным БД. Все пользователи общие, доступ к отдельным БД определяется настройками доступа.

Роль – субъект доступа в БД/СУБД, наделенный определенным набором привилегий (чаще всего употребляется как обобщение группы пользователей для выполнения определенного набора действий в БД/СУБД).

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

SQL	–	Structured Query Language
БД	–	База данных
КС	–	Контрольные суммы
КЦ	–	Контроль целостности
ОС	–	Операционная система
СУБД	–	Система управления базами данных
ФСТЭК России	–	Федеральная служба по техническому и экспортному контролю России

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------