

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ «ЈАТОВА»

Руководство по настройке. Часть 30.
Запись событий информационной безопасности.
Компонент «ja_seceventlog»

643.72410666.00067-07 98 01-30

Листов 46

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

АННОТАЦИЯ

В документе приведены сведения, необходимые для установки и эксплуатации компонента «ja_seceventlog» (далее по тексту – «компонент»), предназначенного для формирования событий безопасности СУБД в соответствии с ГОСТ-Р-59548-2022 «Национальный стандарт Российской Федерации. Защита информации. Регистрация событий безопасности. Требования к регистрируемой информации» и хранения их в отдельном каталоге.

Настоящее руководство предназначено для администраторов СУБД.



Примеры в данном документе приведены для СУБД «Jatoba» версии ядра 6.

Для СУБД «Jatoba» версий ядра 5 и 6 используется версия компонента — 3.1.

Степени важности примечаний, применяемые в документе:



Важная информация – указания, требующие особого внимания



Дополнительная информация – указания, позволяющие упростить работу с изделием

СОДЕРЖАНИЕ

1. Назначение компонента	5
1.1. Условия применения	5
1.2. Ограничения	5
2. Установка и настройка	6
2.1. Установка в ОС GNU Linux	6
2.2. Настройка конфигурационного файла «postgresql.conf» для компонента «ja_seceventlog»	7
2.3. Параметры регистрации событий безопасности компонентом «ja_seceventlog»	10
2.3.1. ja_seceventlog.log	11
2.3.2. ja_seceventlog.log_destination	12
2.3.3. ja_seceventlog.log_relation	13
2.3.4. ja_seceventlog.log_directory	13
2.3.5. ja_seceventlog.log_rotation_size	13
2.3.6. ja_seceventlog.log_rotation_age	14
2.3.7. ja_seceventlog.code_desc_cache_limit	14
2.3.8. ja_seceventlog.source_desc_cache_limit	14
2.3.9. ja_seceventlog.log_parameter_max_size	14
2.3.10. ja_seceventlog.log_destination_table	15
2.3.11. ja_seceventlog.db_name	15
2.3.12. ja_seceventlog.jasel_syslog_facility	16
2.3.13. ja_seceventlog.jasel_syslog_ident	16
2.3.14. ja_seceventlog.jasel_syslog_sequence_numbers	16
2.3.15. ja_seceventlog.jasel_split_messages	17
2.3.16. ja_seceventlog.maxage	17
2.3.17. ja_seceventlog.maxsize	17
2.3.18. ja_seceventlog.log_size_soft_limit	18
2.3.19. ja_seceventlog.log_size_hard_limit	18
2.3.20. ja_seceventlog.max_queue_size	19
2.3.21. ja_seceventlog.log_filename	20
2.3.22. ja_seceventlog.jasel_syslog_split_messages	21
2.3.23. ja_seceventlog.log_autoclose_minutes	21
2.4. Установка расширения «ja_seceventlog»	21
2.5. Директория хранения журнала событий безопасности компонента «ja_seceventlog»	24
3. Функциональные возможности компонента	26
3.1. Фильтрация событий информационной безопасности	26
3.1.1. Функция создания фильтра (create_filter)	27
3.1.2. Функция отображения фильтров (show_filters)	29
3.1.3. Функция удаления фильтра (drop_filter)	29

3.1.4. Функция удаления всех фильтров (reset_filters)	29
3.2. Запись событий ИБ в таблицу БД	29
4. Обновление компонента	32
4.1. Обновление пакета компонента из репозитория	32
4.2. Обновление расширения с использованием	32
4.3. Настройка расширения «ja_seceventlog» после обновления с версии 2.0 до 3.x	32
4.3.1. Параметр ja_seceventlog.log	33
4.3.2. Параметр ja_seceventlog.max_filters	34
4.3.3. Параметр ja_seceventlog.log_connections	35
4.3.4. Параметр ja_seceventlog.log_disconnections	36
4.3.5. Параметр ja_seceventlog.log_catalog	36
4.3.6. Параметр ja_seceventlog.log_relation	36
4.3.7. Параметр ja_seceventlog.role	37
4.3.8. Параметр ja_seceventlog.log_parameter	38
4.3.9. Параметр ja_seceventlog.log_statement	38
4.3.10. Параметр ja_seceventlog.log_statement_once	39
4.3.11. Параметр ja_seceventlog.log_client	39
4.3.12. Параметр ja_seceventlog.log_level	40
4.3.13. Параметр ja_seceventlog.log_parameter_max_size	40
4.3.14. Параметр ja_seceventlog.log_rows	40
4.3.15. Параметр ja_seceventlog.log_statement	40
5. Удаление компонента	42
5.1. Удаление расширения	42
5.2. Удаление пакета	42
Приложение 1	43
Перечень сокращений	45

1. НАЗНАЧЕНИЕ КОМПОНЕНТА

Компонент «ja_seceventlog» предназначен для формирования событий безопасности СУБД в соответствии с ГОСТ-Р-59548-2022 «Национальный стандарт Российской Федерации. Защита информации. Регистрация событий безопасности. Требования к регистрируемой информации» и хранения их в отдельном каталоге.

Компонент устанавливает собственные и независимые параметры регистрации событий и вырезает события безопасности из журнала аудита СУБД, складывая их в свой журнал.

1.1. Условия применения

Компонент «ja_seceventlog» может использоваться с СУБД «Jatoba» версий 5.x и выше, под управлением операционных систем Windows и GNU/Linux.

1.2. Ограничения

На данной стадии реализации отсутствует интеграция компонента «ja_seceventlog» с разделом «Уведомления» компонента Jatoba Data Safe.

События журналов безопасности хранятся в формате JSON.

При обновлении компонента «ja_seceventlog» с версии 2.0 до 3.x необходимо создать фильтры формирования событий безопасности СУБД, так как это указано в разделе 4.

2. УСТАНОВКА И НАСТРОЙКА

Установка компонента должна производиться от имени пользователя, обладающего административными привилегиями в операционной системе. Данный компонент штатным образом может быть установлен только с СУБД «Jatoba» (см. документ «Защищенная система управления базами данных «Jatoba». Руководство по установке).

2.1. Установка в ОС GNU Linux

Компонент возможно установить при первичной установке СУБД «Jatoba», либо доустановить.

Установку компонента возможно провести двумя способами:

- 1) установка из локального репозитория (CDROM) – производится из файлов, записанных на компакт-диск или скопированных с него;
- 2) установка непосредственно из deb/rpm-файлов – производится опционально, по усмотрению пользователя.

Компонент выполнен в виде отдельного deb или rpm-пакета. Установка компонента осуществляется средствами пакетного менеджера ОС. Для разных типов пакетных менеджеров команда установки немного отличается. Ниже приведены основные типы:

- для систем на основе пакетного менеджера APT (к таким системам относятся все ОС семейства Debian, использующие deb-пакеты) команда установки, следующая:

```
apt-get install jatoba6-ja-seceventlog
```

- для систем на основе пакетных менеджеров YUM/DNF (к таким системам относятся все ОС семейства RedHat и вышедшие из нее, использующие rpm-пакеты) команда установки, следующая:

```
yum install jatoba6-ja_seceventlog
```

Отдельного уточнения требуют операционные системы ALT Linux и openSUSE.

- ALT Linux использует пакетный менеджер APT, но распространяется в виде rpm-пакетов и для нее команда установки выглядит аналогично Debian:

```
apt-get install jatoba6-ja_seceventlog
```

– openSUSE также распространяется в виде rpm-пакетов, но использует собственный пакетный менеджер zypper, для нее команда установки выглядит следующим образом:

```
zypper install jatoba6-ja_seceventlog
```

Установка компонента в составе других версий СУБД «Jatoba» осуществляется аналогично. Отличие будет только в номере версии СУБД, в составе которой он распространяется. Например, jatoba5-ja-seceventlog и т.п.

Удаление модуля также осуществляется средствами пакетного менеджера ОС. Вместо команды install нужно использовать соответствующую данному пакетному менеджеру команду удаления (remove, purge, erase и т.п.).

Для получения детальной информации по пакетному менеджеру рекомендуется обратиться к документации по ОС.

2.2. Настройка конфигурационного файла «postgresql.conf» для компонента «ja_seceventlog»

Функциональные возможности компонента позволяют сохранять события безопасности:

- в системный журнал операционной системы (SYSLOG);
- в отдельный каталог (JSON);
- в таблицу БД;
- параллельно в отдельный каталог и в таблицу БД.

Данные функциональные возможности выделяют ключевые, зависимые и дополнительные параметры компонента. Правильная установка которых напрямую влияет на работоспособность компонента.

Взаимосвязи ключевых и зависимых параметров компонента представлены на рисунке 2.1.

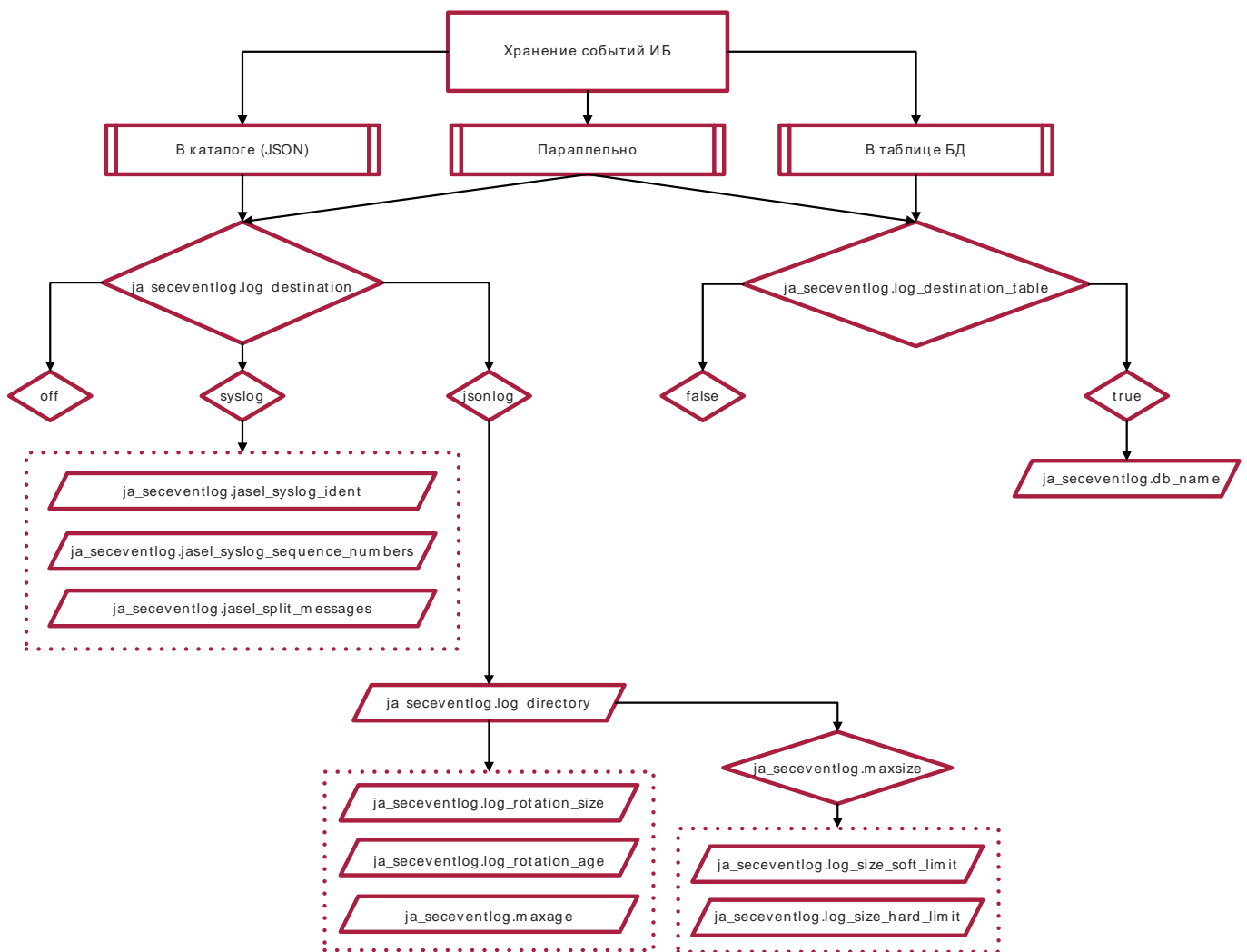


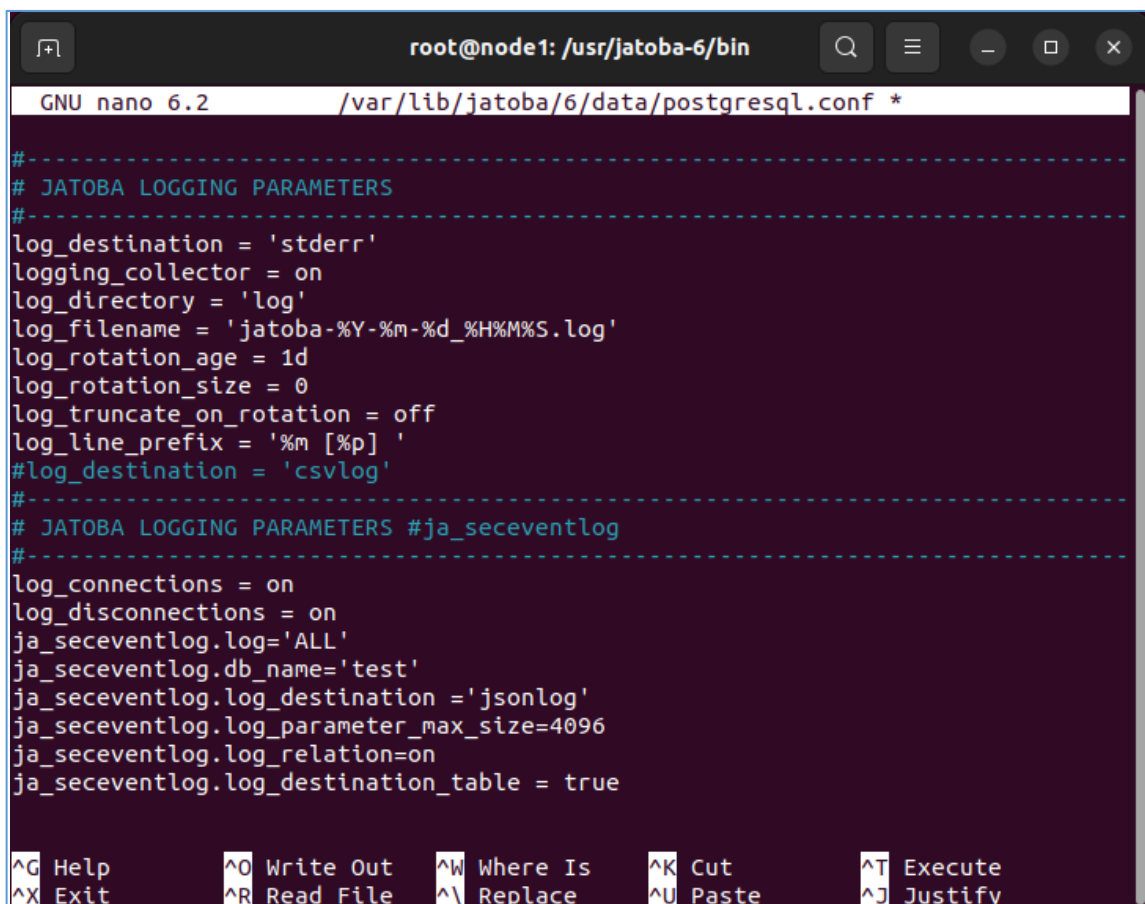
Рисунок 2.1 – Схема зависимости параметров компонента

Установка параметров компонента осуществляются или через конфигурационный файл «postgresql.conf», или после установки расширения SQL-командой, как описано в п.п. 2.3.

Загрузка библиотеки компонента, для последующей установки расширения, выполняется в разделе «Shared Library Preloading» установкой параметра:

```
shared_preload_libraries = 'ja_seceventlog'
```

В случае конфигурирования компонента через конфигурационный файл СУБД, потребуется создать раздел «JATOBA LOGGING PARAMETERS # ja_seceventlog» и установить параметры регистрации событий в СУБД компонента «ja_seceventlog».



```
root@node1: /usr/jatoba-6/bin
GNU nano 6.2 /var/lib/jatoba/6/data/postgresql.conf *
#-----
# JATOBA LOGGING PARAMETERS
#-----
log_destination = 'stderr'
logging_collector = on
log_directory = 'log'
log_filename = 'jatoba-%Y-%m-%d_%H%M%S.log'
log_rotation_age = 1d
log_rotation_size = 0
log_truncate_on_rotation = off
log_line_prefix = '%m [%p] '
#log_destination = 'csvlog'
#-----
# JATOBA LOGGING PARAMETERS #ja_seceventlog
#-----
log_connections = on
log_disconnections = on
ja_seceventlog.log='ALL'
ja_seceventlog.db_name='test'
ja_seceventlog.log_destination = 'jsonlog'
ja_seceventlog.log_parameter_max_size=4096
ja_seceventlog.log_relation=on
ja_seceventlog.log_destination_table = true

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Рисунок 2.2 – Параметры регистрации событий безопасности компонентом «ja_seceventlog»



Параметр сохранения событий безопасности `ja_seceventlog.log_directory` рекомендуется закомментировать или не включать в конфигурационный файл при первоначальном конфигурировании для исключения ошибки. Поскольку расширение на данном этапе еще не установлено, каталог хранения может быть не создан и не установлены права на каталог.

```
#ja_seceventlog.log_directory = '/audit_log'
```

Пример минимальных параметров компонента «ja_seceventlog» в конфигурационном файле СУБД приведен в следующем листинге:

```
#-----
# JATOBA LOGGING PARAMETERS #ja_seceventlog
#-----
log_connections = on
log_disconnections = on
ja_seceventlog.log='ALL'
ja_seceventlog.db_name='test'
ja_seceventlog.log_destination = 'jsonlog'
```

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

```
ja_seceventlog.log_parameter_max_size=4096  
ja_seceventlog.log_relation=on  
ja_seceventlog.log_destination_table = true
```

В разделе «Locale and Formatting» обеспечить следующее значение кодировки сообщений:

```
lc_messages = 'en_US.UTF-8'
```

Применение установленных параметров в конфигурационных файлах выполняется перезагрузкой службы СУБД:

- в ОС Windows:

```
net stop JatobaServer  
net start JatobaServer
```

- в GNU Linux:

```
systemctl restart jatoba-<ver>  
systemctl status jatoba-<ver>
```

2.3. Параметры регистрации событий безопасности компонентом «ja_seceventlog»

После установки расширения компонента становится доступным вывод параметров SQL-командой:

```
SELECT name,setting from pg_settings where name like  
'%seceventlog%';
```

```
root@node1: /usr/jatoba-6/bin

postgres@node1:~$ psql
Password for user postgres:
psql (16.9)
Type "help" for help.

postgres=# SELECT name,setting from pg_settings where name like '%seceventlog%';
          name          |          setting
-----+-----
ja_seceventlog.code_desc_cache_limit | 200
ja_seceventlog.db_name | test
ja_seceventlog.jasel_syslog_facility | local0
ja_seceventlog.jasel_syslog_ident | jatoba
ja_seceventlog.jasel_syslog_sequence_numbers | off
ja_seceventlog.jasel_syslog_split_messages | on
ja_seceventlog.log | ALL
ja_seceventlog.log_autoclose_minutes | 0
ja_seceventlog.log_destination | jsonlog
ja_seceventlog.log_destination_table | on
ja_seceventlog.log_directory | log
ja_seceventlog.log_filename | audit-%Y-%m-%d_%H%M%S
ja_seceventlog.log_parameter_max_size | 4096
ja_seceventlog.log_relation | on
ja_seceventlog.log_rotation_age | 1440
ja_seceventlog.log_rotation_size | 10240
ja_seceventlog.log_size_hard_limit | 0
ja_seceventlog.log_size_soft_limit | 0
ja_seceventlog.max_filters | 128
ja_seceventlog.max_queue_size | 100000
ja_seceventlog.maxage | 30
ja_seceventlog.maxsize | 10240
ja_seceventlog.source_desc_cache_limit | 20
(23 rows)

postgres=#
```

Рисунок 2.3 – Вывод параметров компонента

Установить или изменить параметры компонента в СУБД возможно от имени и с правами привилегированного пользователя SQL-командой:

```
ALTER SYSTEM
```

В этом случае параметр и его значения запишутся в конфигурационный файл СУБД «postgresql.auto.conf» и применить их, без перезагрузки СУБД, возможно SQL-командой:

```
SELECT pg_reload_conf ();
```

Компонент «ja_seceventlog» использует параметры регистрации событий безопасности СУБД приведенные ниже.

2.3.1. ja_seceventlog.log

Параметр определяет какие классы операторов будут регистрироваться в журнале событий безопасности СУБД.



В зависимости от версии компонента имеются отличия в значениях параметров `ja_seceventlog.log`

Значения параметра для версии 3.x могут быть следующими:

- NONE – события аудита не регистрируются в независимости от установленных фильтров (см. п.п. 3.1);
- ALL – регистрируются все события.

```
ja_seceventlog.log = 'ALL'
```

Значения параметра для версии 2.0 могут быть следующими:

- READ – регистрируются SQL-команды SELECT, COPY в случае если источником является отношение или запрос;
- WRITE – регистрируются SQL-команды INSERT, UPDATE, DELETE, TRUNCATE, и COPY;
- FUNCTION – регистрируются функции CALLS и DO;
- ROLE – регистрируются SQL-команды, относящиеся к ролям и системным привилегиям такие как, GRANT, REVOKE, CREATE/ALTER/DROP ROLE;
- DDL – регистрируются SQL-команды DDL не относящиеся к параметру ROLE;
- MISC – регистрируются прочие команды SQL-команды, такие как DISCARD, FETCH, CHECKPOINT, VACUUM, SET;
- MISC_SET – регистрируются SQL-команды типа SET.

Создание фильтров событий в версии 3.x, соответствующих версии 2.0 приводится в п.п. **Ошибка! Источник ссылки не найден..**

2.3.2. `ja_seceventlog.log_destination`

Параметр, определяет формат записи журнала событий безопасности СУБД.

Расширение `ja_seceventlog` поддерживает несколько методов для регистрации сообщений сервера, включая `jsonlog` (JSON). При настройке параметра необходимо выбрать один из следующих методов:

- jsonlog – логирование в отдельный каталог в формате JSON;
- syslog – логирование системный журнал операционной системы (SYSLOG);
- off – логирование в отдельный каталог (JSON) или в системный журнал операционной системы (SYSLOG) отключено.

```
ja_seceventlog.log_destination = 'jsonlog'
```

Значение по умолчанию – jsonlog.

Параметр csvlog отключен.

2.3.3. ja_seceventlog.log_relation

Параметр указывает, должно ли ведение журнала аудита сессии создавать отдельную запись журнала для каждого отношения (TABLE, VIEW и т. д.), на которое ссылается оператор SELECT или DML.

```
ja_seceventlog.log_relation = on
```

2.3.4. ja_seceventlog.log_directory

Параметр указывает на директорию хранения журнала событий безопасности

```
ja_seceventlog.log_directory = '/audit_log'
```


Значение по умолчанию – LOG (/var/lib/jatoba/<ver>/data/log).

2.3.5. ja_seceventlog.log_rotation_size

Параметр, определяющий ротацию журнала по достижению установленного размера.

- B – байт (bytes);
- kB – килобайт (kilobytes);
- MB – мегабайт (megabytes);
- GB – гигабайт (gigabytes);
- TB – терабайт (terabytes).


```
ja_seceventlog.log_rotation_size = 0
```

-  Не рекомендуется ставить небольшие значения, ротация журналов в таком случае будет происходить не чаще чем 1 раз в секунду.

2.3.6. ja_seceventlog.log_rotation_age

Параметр определяет переключение на другой файл журнала каждые N минут.

```
ja_seceventlog.log_rotation_age = 0
```

-  Не рекомендуется ставить небольшие значения, ротация журналов в таком случае будет происходить не чаще чем 1 раз в секунду.

2.3.7. ja_seceventlog.code_desc_cache_limit

Параметр устанавливает максимальное количества значений secEventId (из таблицы ja_seceventlog.secevent_code_desc), хранящихся в кеше.

Параметр 2.3.7. ja_seceventlog.code_desc_cache_limit меняется только при изменении состава событий расширения. Не должен быть меньше количества записей в служебной таблице ja_seceventlog.secevent_code_desc

Значение по умолчанию – 200

```
ja_seceventlog.code_desc_cache_limit = 200
```

2.3.8. ja_seceventlog.source_desc_cache_limit

Параметр устанавливает максимальное количество описаний событий безопасности в кеше.

Значение по умолчанию – 20.

```
ja_seceventlog.source_desc_cache_limit = 20
```

2.3.9. ja_seceventlog.log_parameter_max_size

Параметр определяет максимальный размер параметров, которые будут регистрироваться в журнале аудита СУБД. Если значение равно 0 (по умолчанию), все параметры регистрируются независимо от длины.

```
ja_seceventlog.log_parameter_max_size = 0
```

Значение по умолчанию – 4096.

Диапазон значений от 0 до $2^{30}-1$.

Если при выполнении запроса использовались параметры SQL-запроса, то значение параметра SQL-запроса будет попадать в журнал только если его (параметра) длина (в байтах) меньше или равна значению указанного параметра `ja_seceventlog.log_parameter_max_size`. Если длина (в байтах) значения параметра SQL-запроса больше заданного значения `ja_seceventlog.log_parameter_max_size`, то вместо значения параметра в журнал будет выводиться предупреждающее сообщение "long param suppressed".

2.3.10. `ja_seceventlog.log_destination_table`

Параметр определяет запись событий информационной безопасности в таблицу `ja_seceventlog.secevent_log` и непосредственно связан с параметром «`ja_seceventlog.db_name`».

```
ja_seceventlog.log_destination_table = true
```

Значение по умолчанию – true.

2.3.11. `ja_seceventlog.db_name`

При включенном протоколировании в таблицу БД, то есть при установленном параметре `ja_seceventlog.log_destination_table = true`, определяет имя базы данных, в которой содержится таблица.

```
ja_seceventlog.db_name = postgres
```

Значение по умолчанию – postgres.



Если изначально в конфигурационном файле `postgresql.conf` был установлен параметр записи событий безопасности в таблицу «`ja_seceventlog.log_destination_table`» и не установлен параметр определяющий размещение таблицы в БД «`ja_seceventlog.db_name`», то по умолчанию будет использована БД по умолчанию «postgres».

В последующем добавление либо изменение параметра «ja_seceventlog.db_name» невозможно, так как это потребует переустановки расширения «ja_seceventlog».

2.3.12. ja_seceventlog.jasel_syslog_facility

Параметр «ja_seceventlog.jasel_syslog_facility» является зависимым от параметра «ja_seceventlog.log_destination», который определяет формат записи событий безопасности, если установлено значение «syslog».

Параметр используется для указания типа протоколируемой программы (facility). Используются допустимые значения LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7.

```
ja_seceventlog.jasel_syslog_facility = LOCAL0
```

Значение по умолчанию – LOCAL0.

2.3.13. ja_seceventlog.jasel_syslog_ident

Параметр «ja_seceventlog.jasel_syslog_ident» является зависимым от параметра «ja_seceventlog.log_destination», который определяет формат записи событий безопасности, если установлено значение «syslog».

Параметр «ja_seceventlog.jasel_syslog_ident» задаёт имя программы, которое будет использоваться для идентификации сообщений.

```
ja_seceventlog.jasel_syslog_ident = jatoba
```

Значение по умолчанию – jatoba.

2.3.14. ja_seceventlog.jasel_syslog_sequence_numbers

Параметр «ja_seceventlog.jasel_syslog_sequence_numbers» является зависимым от параметра «ja_seceventlog.log_destination», который определяет формат записи событий безопасности, если установлено значение «syslog». При данном значении параметр «ja_seceventlog.jasel_syslog_sequence_numbers» включен по умолчанию.

Параметр «ja_seceventlog.jasel_syslog_sequence_numbers» при выводе событий безопасности в системный журнал операционной системы (SYSLOG) последовательно увеличивает номера сообщений

Значение по умолчанию – false.

2.3.15. ja_seceventlog.jasel_split_messages

Параметр «ja_seceventlog.jasel_split_messages» является зависимым от параметра «ja_seceventlog.log_destination», который определяет формат записи событий безопасности, если установлено значение «syslog».

Когда параметр «ja_seceventlog.jasel_split_messages» имеет значение «syslog», т.е. активен вывод в системный журнал операционной системы (SYSLOG) событий безопасности, этот параметр определяет, как будут доставляться сообщения. .

Если он включён (по умолчанию), сообщения разделяются по строкам, а длинные строки разбиваются на строки не длиннее 1024 байт, что составляет типичное ограничение размера для традиционных реализаций syslog

Значение по умолчанию – true.

2.3.16. ja_seceventlog.maxage

Параметр определяет количество дней, в течение которых будут храниться журналы событий безопасности СУБД в отдельном каталоге.

Если значение равно «0», то удаление файлов по этому параметру производиться не будет.

```
ja_seceventlog.maxage = 30d
```

Значение по умолчанию – 30d

2.3.17. ja_seceventlog.maxsize

Параметр «ja_seceventlog.maxsize» определяет объем дисковой памяти, выделенной администратором для хранения журналов событий безопасности СУБД в отдельном каталоге.

Значения параметра могут задаваться со значениями kB, MB, GB, TB.

```
ja_seceventlog.maxsize = 10MB
```

Значение по умолчанию – 10MB.

Если значение равно «0», то не будут выполняться проверки на программные (soft) и аппаратные (hard) лимиты.

2.3.18. ja_seceventlog.log_size_soft_limit

Параметр «ja_seceventlog.log_size_soft_limit» является зависимым от параметра «ja_seceventlog.maxsize» который устанавливает размер каталога для хранения журналов событий безопасности СУБД.

Параметр «ja_seceventlog.log_size_soft_limit» определяет процент заполнения журналами, от установленного размера каталога в параметре «ja_seceventlog.maxsize», при достижении которого будет записано соответствующее сообщение в журнал событий безопасности СУБД в отдельном каталоге.

```
ja_seceventlog.log_size_soft_limit = 80
```

Значение по умолчанию – 0.

Если значение равно «0», то не будут выполняться проверки на программный (soft) лимит.



Значение параметра должно быть не более 80%.

Данное значение является оптимальным.

2.3.19. ja_seceventlog.log_size_hard_limit

Параметр «ja_seceventlog.log_size_hard_limit» является зависимым параметром от параметра «ja_seceventlog.maxsize», который определяет объем дисковой памяти, выделенной администратором для хранения журналов событий безопасности СУБД в отдельном каталоге.

Параметр «ja_seceventlog.log_size_hard_limit» управляет процент заполнения объема выделенного каталога и при достижении порогового значения удаляет старый файл журнала

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

событий безопасности. О выполненной операции выполняется делается запись в актуальном файле журналов событий безопасности СУБД

```
ja_seceventlog.log_size_hard_limit = 0%
```

Значение по умолчанию – 0%.

Если значение равно «0», то не будут выполняться проверки на аппаратный (hard) лимит.



Значение параметра «log_size_hard_limit» должно быть больше значения параметра log_size_soft_limit минимум на 10%.

Максимальное значение параметра не должно превышать 90%

2.3.20. ja_seceventlog.max_queue_size

Любой рабочий процесс СУБД (backend), который выполняет какую-то активность и генерирует события безопасности. В качестве получателя сообщений с событиями безопасности выступает специальный рабочий процесс «ja_seceventlog worker», который записывает события в служебную таблицу ja_seceventlog.secevent_log в служебную базу данных, заданную параметром ja_seceventlog.db_name.

Чтобы минимизировать затраты на синхронизацию и передачу сообщений от рабочего процесса к «ja_seceventlog worker» реализован специальный буфер в памяти, в котором реализуется очередь событий.

Размер данной очереди событий регулируется параметром ja_seceventlog.max_queue_size.

Чем больше данный параметр, тем больше рабочих процессов может одновременно записать туда сообщения без значительного влияния на производительность этих рабочих процессов.

```
ja_seceventlog.max_queue_size = 100000
```

Значение по умолчанию – 100000.



Максимальное устанавливаемое значение параметра 10000000, что грубо соответствует размеру очереди событий в ОЗУ 19Гб (количество элементов в

очереди * 2 Кб). В случае если комплекс технических средств не располагает доступным объемом ОЗУ выполняется остановка СУБД с выводом сообщения:

```
FATAL: could not map anonymous shared memory: Cannot  
allocate mamory  
LOG: database system is shut down
```

2.3.21. ja_seceventlog.log_filename

Параметр «ja_seceventlog.log_filename» определяет формат названия журнала событий безопасности.

В значении параметра допускается использовать только следующие спецификаторы:

- %Y;
- %m;
- %d;
- %H;
- %M;
- %S.

Порядок указания спецификаторов в параметре «ja_seceventlog.log_filename» не имеет значения.

```
ja_seceventlog.log_filename = 'audit-%Y-%m-%d_%H%M%S'
```

Значение по умолчанию – audit-%Y-%m-%d_%H%M%S.



В случае в значении параметра «ja_seceventlog.log_filename» пропущен один или несколько спецификаторов СУБД выдает сообщение о некорректном значении параметра «ja_seceventlog.log_filename» в свой системный журнал:

```
The *** specifier is missing in the  
ja_seceventlog.log_filename parameter. The parameter  
must "  
"contain the date and time specifiers %%Y, %%m, %%d,  
%%H, %%M, %%S in any order.
```

2.3.22. ja_seceventlog.jasel_syslog_split_messages

Параметр «ja_seceventlog.jasel_syslog_split_messages» является зависимым от параметра «ja_seceventlog.log_destination», который определяет формат записи событий безопасности, если установлено значение «syslog».

Параметр «ja_seceventlog.jasel_syslog_split_messages» определяет, как будут доставляться сообщения:

Включен, сообщения разделяются по строкам, а длинные строки разбиваются на строки не длиннее 1024 байт, что составляет типичное ограничение размера для традиционных реализаций syslog.

Когда он отключён, сообщения передаются службе syslog как есть, и она должна сама корректно воспринять потенциально длинные сообщения.

```
ja_seceventlog.jasel_syslog_split_messages = on
```

Значение по умолчанию – on.

2.3.23. ja_seceventlog.log_autoclose_minutes

Параметр «ja_seceventlog.log_autoclose_minutes» определяет количество минут бездействия сервера, по достижению которых произойдет автоматическое закрытие журнального файла.

2.4. Установка расширения «ja_seceventlog»

Войти в СУБД от имени и с правами пользователя «SUPERUSER», выполнить SQL-команду:

```
CREATE EXTENSION ja_seceventlog;
```

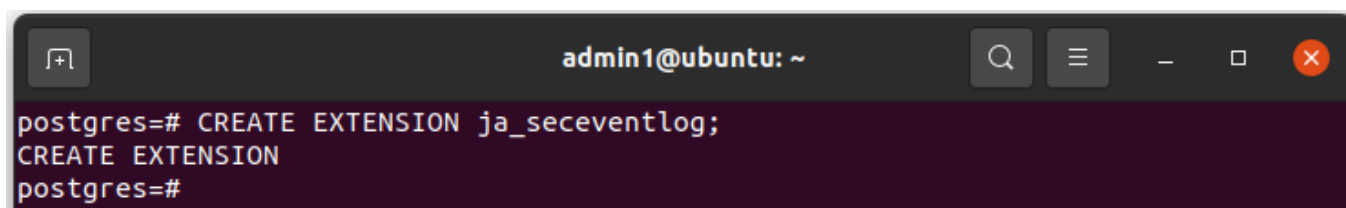


Рисунок 2.4 – Установка расширения «ja_seceventlog»

В результате будет создана одноименная схема данных и служебные таблицы:

- ja_seceventlog.secevent_code_desc, содержащую спецификацию событий безопасности СУБД (см. рисунок 2.5);
- ja_seceventlog.secevent_source_desc, содержащую спецификацию компонентов СУБД (см. рисунок 2.6);
- secevent_log – таблица временного хранения событий безопасности.

admin1@ubuntu: ~

```
postgres=# SELECT * FROM ja_seceventlog.secevent_code_desc;
```

seceventtypegost	seceventtypecode	seceventtypenameshort	seceventnamegost	seceventnamecode	seceventnameshort	seceventpriority	archi
Идентификация и аутентификация субъекта доступа	100	ИАСД	Отказ во входе в связи с тем, что идентификатор не зарегистрирован	100	ИАСД.1	Средний	е
Идентификация и аутентификация субъекта доступа	100	ИАСД	Отказ во входе в связи с тем, что идентификатор заблокирован	101	ИАСД.2	Средний	е
Идентификация и аутентификация субъекта доступа	100	ИАСД	Отказ во входе в связи с неправильным паролем	102	ИАСД.3	Средний	е
Идентификация и аутентификация субъекта доступа	100	ИАСД	Отказ во входе в связи с тем, что превышен лимит попыток ввода пароля	103	ИАСД.4	Средний	е
Идентификация и аутентификация субъекта доступа	100	ИАСД	Отказ во входе в связи с тем, что закончен срок действия пароля	104	ИАСД.5	Средний	е
Идентификация и аутентификация субъекта доступа	100	ИАСД	Успешный вход в систему	105	Успешный вход в систему	Низкий	е
Идентификация и аутентификация субъекта доступа	100	ИАСД	Выход из системы	106	Выход из системы	Низкий	е
Идентификация и аутентификация субъекта доступа	100	ИАСД	Другие события безопасности	107	ИАСД.8	Низкий	е
Идентификация объекта доступа	101	ИОД	Успешная идентификация	108	ИОД.1	Низкий	е
Идентификация объекта доступа	101	ИОД	Ошибка прохождения идентификации	109	ИОД.2	Средний	е
Идентификация объекта доступа	101	ИОД	Другие события безопасности	110	ИОД.3	Низкий	е

Рисунок 2.5 – Таблица «ja_seceventlog.secevent_code_desc»

admin1@ubuntu: ~

```
postgres=# SELECT * FROM ja_seceventlog.secevent_source_desc;
```

seceventcomponentname	seceventcomponentcode	archivestatus	seceventcomponentversion	seceventcomponentdeveloper	seceventcomponenttype	seceventssoftname
jatoba	100	е	5.6.1.1	ООО "Газинформсервис"	Ядро СУЕД	Ядро СУЕД Jatoba
ja_seceventlog	101	е	5.6.1.1	ООО "Газинформсервис"	Расширение	Компонент регистрации событий безопасности
postgres_fdw	102	е	5.6.1.1	Postgres	Расширение	Компонент обращения к данным, находящимся на внешних серверах Jatoba
jaDog	103	е	5.6.1.1	ООО "Газинформсервис"	Расширение	Компонент управления режимом работы узлов кластера
ja_sync_ldap	104	е	5.6.1.1	ООО "Газинформсервис"	Расширение	Компонент синхронизации учетных записей со службами каталогов
SecurityProfile	105	е	5.6.1.1	ООО "Газинформсервис"	Расширение	Компонент управления паролями политиками пользователей СУЕД
ja_CSum	106	е	5.6.1.1	ООО "Газинформсервис"	Расширение	Компонент контроля целостности
jaPooler	107	е	5.6.1.1	ООО "Газинформсервис"	Расширение	Компонент балансировки подключений пользователей к СУЕД
ja_Log	108	е	5.6.1.1	ООО "Газинформсервис"	Расширение	Компонент централизованного сбора записей событий СУЕД

(9 rows)

```
postgres=#
```

Рисунок 2.6 – Таблица «ja_seceventlog.secevent_source_desc»

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

2.5. Директория хранения журнала событий безопасности компонента «ja_seceventlog»

По умолчанию директория хранения журнала событий безопасности компонента «ja_seceventlog» /var/lib/jatoba/<ver>/data/log.

Просмотреть текущую директорию хранения журнала событий безопасности компонента «ja_seceventlog» возможно SQL-командой:

```
SHOW ja_seceventlog.log_directory;
```

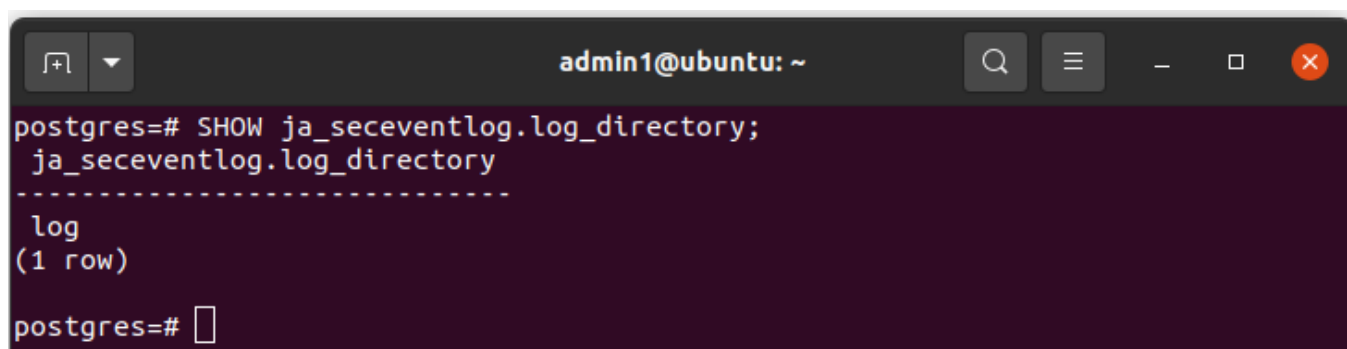


Рисунок 2.7 – Вывод хранения журнала событий безопасности компонента «ja_seceventlog»

Задание директории хранения журнала событий безопасности компонента «ja_seceventlog» возможно только через установку параметра в конфигурационном файле postgresql.conf:

```
ja_seceventlog.log_directory = 'audit_log'
```

Директория хранения может отличаться и в этом случае указывается полный путь к ней.

В рассматриваемом примере директорией хранения журнала событий безопасности компонента «ja_seceventlog» является директория в корневом каталоге «audit_log».

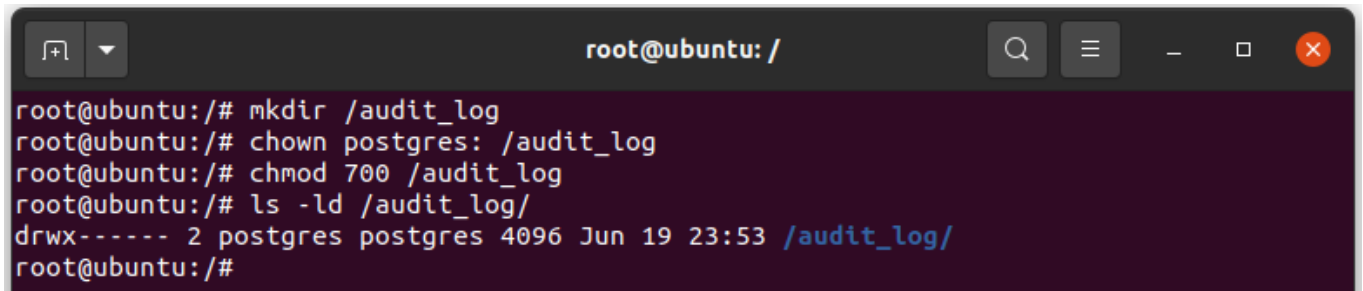
Порядок действий для хранения журнала событий безопасности в отдельной директории будет следующий:

- 1) Создать каталог audit_log:

```
mkdir /audit_log
```

- 2) Задать права на созданный каталог:


```
# chown postgres: /audit_log
# chmod 700 /audit_log
# ls -ld /audit_log/
```



```
root@ubuntu: /
root@ubuntu:/# mkdir /audit_log
root@ubuntu:/# chown postgres: /audit_log
root@ubuntu:/# chmod 700 /audit_log
root@ubuntu:/# ls -ld /audit_log/
drwx----- 2 postgres postgres 4096 Jun 19 23:53 /audit_log/
root@ubuntu:/#
```

Рисунок 2.8 – Создание директории и установка прав на нее

- 3) Установить параметр в конфигурационном файле «postgresql.conf»:

```
ja_seceventlog.log_directory = '/audit_log'
```

- 4) Перезагрузить службу СУБД «Jatoba».

Таким образом журнала событий безопасности будет храниться отдельно от журнала аудита СУБД.

3. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ КОМПОНЕНТА

3.1. Фильтрация событий информационной безопасности

Во время установки расширения будет создана служебная таблица фильтров ja_seceventlog.filter. Таблица ja_seceventlog.filter предназначена для хранения фильтров и имеет следующую структуру:

- db_name (тип данных text) – название БД фильтра;
- seceventid (тип данных text) – код события безопасности фильтра;
- object_type (тип данных text) – тип объекта фильтра;
- object_name (тип данных text) – название объекта фильтра;
- role_name (тип данных text) – название субъекта фильтра;
- comment (тип данных text) – текст комментария к фильтру.

При первоначальной установке расширения в таблице создаются предустановленные фильтры событий безопасности (см. таблицу 3.1).

Таблица 3.1 – Список предустановленных фильтров событий безопасности

id	Название БД фильтра (db_name)	Код события (seceventid)	Тип объекта (object_type)	Название объекта (object_name)	Название субъекта (role_name)	Комментарий (comment)
1	ALL	111162100	ALL	ALL	ALL	Запуск и останов СУБД
2	ALL	119193101	ALL	ALL	ALL	Журнал очищен
3	ALL	119194101	ALL	ALL	ALL	Журналирование отключено
4	ALL	119203101	ALL	ALL	ALL	Предупреждение о заполнении памяти
5	ALL	119205101	ALL	ALL	ALL	Журналирование включено
6	ALL	119206101	ALL	ALL	ALL	Перезапись событий



Расшифровка кодов событий безопасности приведена в документе «Реализация функций безопасности» 643.72410666.00067-07 94 01 в таблице 4.2 – Структура

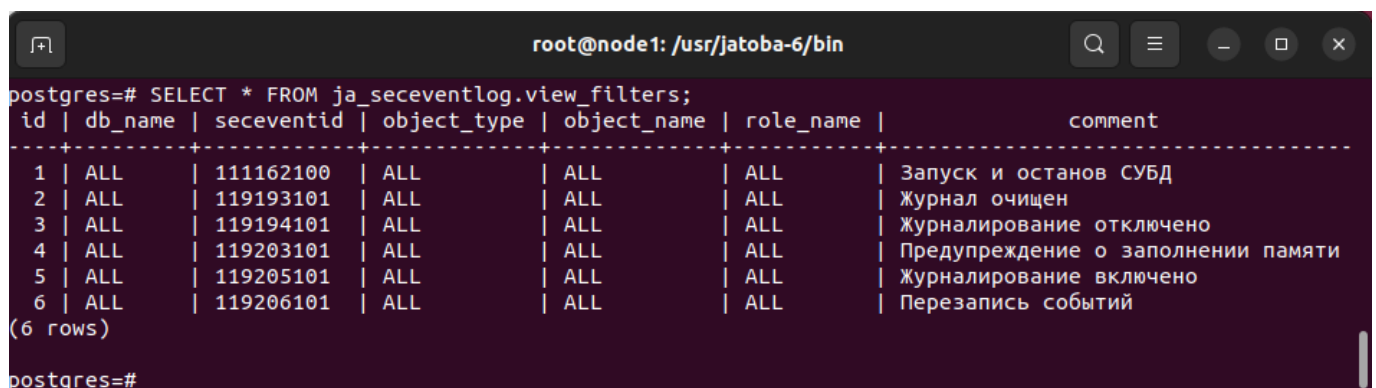
справочной таблицы (ja_seceventlog.secevent_code_desc) с типами и уровнем важности событий безопасности

Расширение предоставляет возможность управления фильтрами при помощи специальных функций:

- create_filter (см. п.п. 3.1.1).
- show_filters ();
- drop_filter (см. п.п. 3.1.3);
- reset_filters ().

Для просмотра установленных или созданных фильтров необходимо воспользоваться представлением ja_seceventlog.view_filters:

```
SELECT * FROM ja_seceventlog.view_filters;
```



id	db_name	seceventid	object_type	object_name	role_name	comment
1	ALL	111162100	ALL	ALL	ALL	Запуск и останов СУБД
2	ALL	119193101	ALL	ALL	ALL	Журнал очищен
3	ALL	119194101	ALL	ALL	ALL	Журналирование отключено
4	ALL	119203101	ALL	ALL	ALL	Предупреждение о заполнении памяти
5	ALL	119205101	ALL	ALL	ALL	Журналирование включено
6	ALL	119206101	ALL	ALL	ALL	Перезапись событий

Рисунок 3.1 – Просмотр установленных или созданных фильтров «ja_seceventlog»

Предустановленные фильтры имеют значение идентификатора (id) от 1 до 999. Пользовательские фильтры начинаются с идентификатора 1000 и более.

3.1.1. Функция создания фильтра (create_filter)

Функция create_filter предназначена для создания нового правила регистрации событий (фильтра) компонента ja_seceventlog.

Созданные фильтры сохраняются в отдельную таблицу filter в схеме ja_seceventlog.

Функция create_filter содержит следующие параметры:

- db_name – символьный параметр. Название БД, для которой задаётся фильтр.

Указание NULL или пустой строки означает, что регистрируются события для всех БД;

– `seceventid` – символьный параметр. Идентификатор события безопасности, для которого задается фильтр. Указание NULL или пустой строки означает, что регистрируются события для всех идентификаторов событий безопасности. Список идентификаторов событий безопасности содержится в документе «Реализация функций безопасности» 643.72410666.00067-07 94 01;

– `object_type` – символьный параметр. Тип объекта, для которого задается фильтр. Указание NULL или пустой строки означает, что регистрируются события для всех типов объектов. Возможные значения: SCHEMA, TABLE, VIEW и так далее. Полный список типов объектов приведен в Приложение 1;

– `object_name` – символьный параметр. Имя объекта, для которого задается фильтр. Указание NULL или пустой строки означает, что регистрируются события для всех объектов;

– `role_name` – символьный параметр. Имя роли, для которой задается фильтр. Если имя роли указано, то регистрируются действия, совершённые указанным пользователем или пользователем, который прямо или косвенно является членом указанной роли. Указание NULL или пустой строки означает, что регистрируются события для всех ролей.

– `comment` – комментарий, описывающий созданный фильтр.

Синтаксис функции создания нового фильтра:

```
ja_seceventlog.create_filter(db_name text, seceventid text,  
object_type text, object_name text, role_name text, comment  
text)
```

Примеры:

```
SELECT ja_seceventlog.create_filter('db1', NULL, NULL, NULL,  
NULL, 'все события для базы db1');  
  
SELECT ja_seceventlog.create_filter('db1', NULL, NULL, NULL,  
'user1', 'все события для пользователя user1 в базе db1');
```

3.1.2. Функция отображения фильтров (show_filters)

Функция `show_filters` выводит полный список текущих фильтров из таблицы фильтров `filter`. Пример отображения фильтров представлен выше (представление `ja_seceventlog.view_filters`, см. п.п. 3.1).

Функция `show_filters` выводит в каждой строке отдельное правило с указанием всех заполненных полей с номером идентификатора этого правила. Предусмотренные фильтры имеют значение идентификатора (`id`) от 1 до 999. Пользовательские фильтры начинаются с идентификатора 1000 и более.

Любое `NULL` значение заменяется на ключевое слово “ALL”, означая, что допускается запись любого события по данному атрибуту фильтра.

3.1.3. Функция удаления фильтра (drop_filter)

Функция `drop_filter` предназначена для удаления правила регистрации событий (фильтра).

При использовании функции `drop_filter` происходит удаление фильтра из таблицы фильтров `filter` в схеме `ja_seceventlog`.

3.1.4. Функция удаления всех фильтров (reset_filters)

Функция `reset_filters` удаляет **все** фильтры из таблицы фильтров `filter`.

3.2. Запись событий ИБ в таблицу БД

Компонент обладает функциональной возможностью записи событий информационной безопасности в отдельную таблицу `ja_seceventlog.secevent_log`. Данная таблица создаётся при установке расширения.

Таблица `ja_seceventlog.secevent_log` может использоваться для интеграции с SIEM системой.

Параметр `ja_seceventlog.log_destination_table` имеет значение «true» по умолчанию:

```
ja_seceventlog.log_destination_table = true
```

Запись событий информационной безопасности в таблицу начинается автоматически. Очистка таблицы выполняется каждый час.

Расположение таблицы в СУБД определяется параметром «ja_seceventlog.db_name». По умолчанию используется значение:

```
ja_seceventlog.db_name = postgres
```

Изменение параметра не рекомендуется, т.к. это приведет к переустановке компонента (см. описание параметра «ja_seceventlog.db_name»).

Таблица имеет 2 поля:

- log_time – штамп времени (во временной зоне сервера, заданной в postgresql.conf параметром timezone). Время сохраняется с точностью до микросекунд;
- log_data – содержание события в формате JSON.

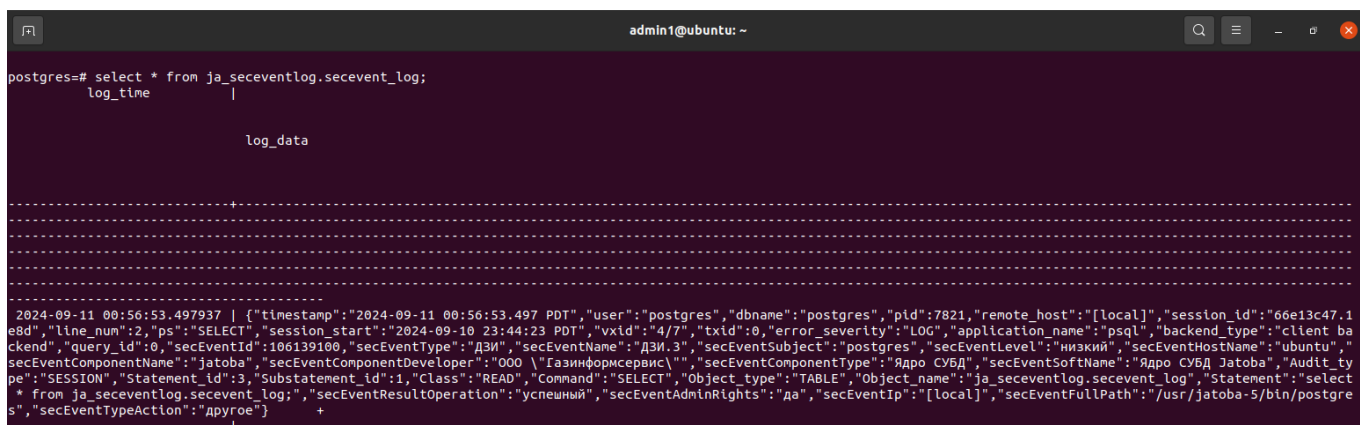


Рисунок 3.2 – Содержание таблицы ja_seceventlog.secevent_log

Запись событий информационной безопасности выполняется параллельно и в таблицу, и в отдельный каталог если установлены параметры:

```
ja_seceventlog.log_destination = jsonlog  
ja_seceventlog.log_destination_table = true
```

В компоненте «ja_seceventlog» доступно представление ja_seceventlog.secevent_log_last_hour. Данное представление выполняет сбор данных по таблицам хранения событий безопасности всех событий за последний час текущего времени сервера.

Представление ja_seceventlog.secevent_log_last_hour содержит следующие поля:

- `log_time` – штамп времени (во временной зоне сервера, заданной в `postgresql.conf` параметром `timezone`). Время сохраняется с точностью до микросекунд;

- `log_data` – содержание события в формате JSON.

Выводимые в представлении `ja_seceventlog.secevent_log_last_hour` записи выбираются в соответствии с временной зоной сервера, в которой записываются события, а не настройках временной зоны, в которой работает пользователь.

Для получения данных из представления `ja_seceventlog.secevent_log_last_hour` можно воспользоваться следующей командой:

```
SELECT * FROM ja_seceventlog.secevent_log_last_hour;
```

```
admin1@node1: ~
postgres@node1:/home/admin1$ psql
Password for user postgres:
psql (16.4)
Type "help" for help.

postgres=# SELECT * FROM ja_seceventlog.secevent_log_last_hour;
      log_time      |
+-----+
2025-01-24 11:17:31.544473 | {"timestamp":"2025-01-24 11:17:31.544 MSK","pid":59842,"session_id":"67934c9b.e9c2","line_num":1,"session_start":"2025-01-24 11:17:31 MSK","txid":0,"error_severity":"LOG","message":"starting PostgreSQL 16.4 on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0, 64-bit","backend_type":"postmaster","query_id":0,"secEventId":111162100,"secEventHostName":"node1","Statement_id":0,"Substatement_id":0,"Command":"UNKNOWN","secEventResultOperation":"успешный","secEventFullPath":"/usr/jatoba-6/bin/postgres","secEventTypeAction":"запуск"}+
2025-01-24 11:17:31.578197 | {"timestamp":"2025-01-24 11:17:31.578 MSK","pid":59849,"session_id":"67934c9b.e9c9","line_num":1,"session_start":"2025-01-24 11:17:31 MSK","vxid":"2/0","txid":0,"error_severity":"LOG","message":"ja_seceventlog: logging is disabled","backend_type":"ja_seceventlog_worker","query_id":0,"secEventId":119194101,"secEventHostName":"node1","secEventResultOperation":"успешный","secEventFullPath":"/usr/jatoba-6/bin/postgres"}+
2025-01-24 11:29:02.712952 | {"timestamp":"2025-01-24 11:29:02.712 MSK","pid":59849,"session_id":"67934c9b.e9c9","line_num":2,"session_start":"2025-01-24 11:17:31 MSK","vxid":"2/0","txid":0,"error_severity":"LOG","message":"ja_seceventlog: partition #10 has been cleared","backend_type":"ja_seceventlog_worker","query_id":0,"secEventId":119193101,"secEventHostName":"node1","secEventResultOperation":"успешный","secEventFullPath":"/usr/jatoba-6/bin/postgres","secEventPartitionClear":10}+
(3 rows)

postgres=#
```

Рисунок 3.3 – Содержание представления `ja_seceventlog.secevent_log_last_hour`

4. ОБНОВЛЕНИЕ КОМПОНЕНТА

4.1. Обновление пакета компонента из репозитория

Перед началом процесса обновления компонента с версии 2.0 до 3.x необходимо:

- 1) Обновить локальный репозиторий согласно документу «Руководство по установке» 643.72410666.00067-07 97 01;
- 2) Выполнить обновление пакета компонента при помощи штатных средств ОС:

```
apt --only-upgrade install jatoba6-ja-seceventlog
```

- 3) Или всех установленных в ОС пакетов:

```
apt-get upgrade
```

4.2. Обновление расширения с использованием

Другим вариантом обновления компонента является установка новой версии 3.x при помощи команды:

```
dpkg -i jatoba6-ja-seceventlog_3.1.0-XXX_amd64.deb
```

После завершения обновления пакетов компонента «ja_seceventlog» необходимо выполнить перезагрузку службы СУБД «Jatoba» и проверить статус работы:

```
systemctl stop jatoba-6  
systemctl start jatoba-6  
systemctl status jatoba-6
```

Расширение «ja_seceventlog» для СУБД «Jatoba» обновляется при помощи SQL-команды:

```
ALTER EXTENSION ja_seceventlog UPDATE TO '3.1';
```

4.3. Настройка расширения «ja_seceventlog» после обновления с версии 2.0 до 3.x

При обновлении версии расширения с версии 2.0 до 3.x в следующих параметрах произошли изменения.

4.3.1. Параметр ja_seceventlog.log

После обновления расширения с версии 2.0 до 3.x необходимо установить значение параметров регистрации в журнале событий безопасности СУБД как это указано в п.п. 2.3:

```
ja_seceventlog.log = 'ALL'
```

Создать фильтр для логирования всех событий безопасности:

```
SELECT ja_seceventlog.create_filter(NULL, NULL, NULL, NULL,  
NULL, 'фильтр для всех событий');
```



В случае задания данного фильтра, задание других фильтров не требуется.

Указанные выше настройки позволяют регистрировать в журнал аудита все события информационной безопасности, как в версии расширения 2.0.

Если параметр ja_seceventlog.log (см. п.п.2.3) в версии 2.0 был установлен в 'NONE', то после обновления до версии 3.x дополнительных действий по созданию фильтров не требуется.

Если параметр ja_seceventlog.log в версии 2.0 был установлен в 'ALL', то достаточно установить фильтр:

```
SELECT ja_seceventlog.create_filter(NULL, NULL, NULL, NULL,  
NULL, 'фильтр для всех событий');
```

Если в параметре ja_seceventlog.log в версии 2.0 были перечислены конкретные значения, то необходимо создать фильтры, подобные следующим:

– READ:

```
SELECT ja_seceventlog.create_filter(NULL, 106139100, NULL,  
NULL, NULL, 'для событий READ');
```

– WRITE:

```
SELECT ja_seceventlog.create_filter(NULL, 106148100, NULL,  
NULL, NULL, 'для событий WRITE');
```

– FUNCTION:

```
SELECT ja_seceventlog.create_filter(NULL, 106139100, NULL,  
NULL, NULL, 'для событий вызова функций');
```

– **ROLE:**

```
SELECT ja_seceventlog.create_filter(NULL, 105133100, NULL,  
NULL, NULL, 'для событий изменений прав доступа');
```

– **DDL:**

```
SELECT ja_seceventlog.create_filter(NULL, 106145100, NULL,  
NULL, NULL, 'для событий создания ресурса');  
  
SELECT ja_seceventlog.create_filter(NULL, 106146100, NULL,  
NULL, NULL, 'для событий изменения ресурса');  
  
SELECT ja_seceventlog.create_filter(NULL, 106147100, NULL,  
NULL, NULL, 'для событий удаления ресурса');
```

– **MISC:**

```
SELECT ja_seceventlog.create_filter(NULL, 106148100, NULL,  
NULL, NULL, 'для событий DISCARD');  
  
SELECT ja_seceventlog.create_filter(NULL, 106139100, NULL,  
NULL, NULL, 'для событий FETCH, VACUUM, CHECKPOINT');
```

– **MISC_SET:**

```
SELECT ja_seceventlog.create_filter(NULL, 108152100, NULL,  
NULL, NULL, 'для событий SET');  
  
SELECT ja_seceventlog.create_filter(NULL, 100105100, NULL,  
NULL, NULL, 'для событий SET ROLE');
```

4.3.2. Параметр **ja_seceventlog.max_filters**

Параметр **ja_seceventlog.max_filters** определяет максимальное количество создаваемых и применяемых фильтров.

Допустимый диапазон значений параметра **ja_seceventlog.max_filters** от 0 до 1024.

```
ja_seceventlog.max_filters = 128
```



В случае превышения числа «ja_seceventlog.max_filters» пользователю отображается предупреждение при создании фильтра, а также при запуске/перезагрузке СУБД.

Если количество созданных фильтров больше, чем указано в «ja_seceventlog.max_filters», компонент обработает только максимально допустимое количество (равное «ja_seceventlog.max_filters»).

Перед созданием нового фильтра рекомендуется выполнить следующий запрос:

```
SELECT COUNT(*) <=
current_setting('ja_seceventlog.max_filters')::integer
FROM ja_seceventlog.filter;
```

В случае если результатом выполнения запроса будет значение FALSE, новый фильтр не поместится в активную память компонента и может потребоваться увеличение значения параметра «ja_seceventlog.max_filters»

Значение параметра ja_seceventlog.max_filters по умолчанию:

- в версии 3.x – 128;
- в версии 2.0 - параметр отсутствует.

4.3.3. Параметр ja_seceventlog.log_connections

Параметр ja_seceventlog.log_connections активирует регистрацию событий попыток подключения к серверу СУБД.

В версии 3.x параметр удален ja_seceventlog.log_connections, а управление регистрации попыток подключения возложено на параметр log_connections СУБД «Jatoba».

Значение параметра ja_seceventlog.log_connections по умолчанию:

- в версии 3.x – параметр отсутствует;
- в версии 2.0 - off.

Если в файле postgresql.conf значение log_connections = on, то события подключений к СУБД будут регистрироваться при наличии следующих фильтров:

```
SELECT ja_seceventlog.create_filter(NULL, '100105100', NULL,
NULL, NULL, 'успешное подключение');
```

```
SELECT ja_seceventlog.create_filter(NULL, '100107100', NULL,  
NULL, NULL, 'получение соединения');
```

4.3.4. Параметр ja_seceventlog.log_disconnections

Параметр включает ja_seceventlog.log_disconnections регистрацию событий завершения сеанса.

Значение параметра ja_seceventlog.log_disconnections по умолчанию:

- в версии 3.x – параметр отсутствует;
- в версии 2.0 - off.

Если в файле postgresql.conf значение log_disconnections = on, то события завершения работы с СУБД будут регистрироваться при наличии следующего фильтра:

```
SELECT ja_seceventlog.create_filter(NULL, '100106100', NULL,  
NULL, NULL, 'завершение сеанса');
```

4.3.5. Параметр ja_seceventlog.log_catalog

В версии компонента 3.x, параметр «ja_seceventlog.log_catalog» исключен.

Если данный параметр был включен в версии 2.0, то происходила регистрация обращений к таблицам системного каталога.

```
ja_seceventlog.log_catalog = true
```

Значение параметра ja_seceventlog.log_catalog по умолчанию:

- в версии 3.x – параметр отсутствует;
- в версии 2.0 – true.

4.3.6. Параметр ja_seceventlog.log_relation

Позволяет ja_seceventlog.log_relation включать/отключать отдельную запись журнала для каждого отношения (TABLE, VIEW, и т.д.), на которое ссылается оператор SELECT или DML.

Значение параметра ja_seceventlog.log_relation по умолчанию:

- в версии 3.x – off;
- в версии 2.0 – off.

После обновления до версии 3.x при значении `ja_seceventlog.log_relation = on` необходимо добавить фильтры для объектов, например:

```
SELECT ja_seceventlog.create_filter('db1', NULL, 'TABLE',  
'public.table1', NULL, 'все события для table1');  
SELECT ja_seceventlog.create_filter('db1', NULL, 'TABLE',  
'public.table2', NULL, 'все события для table2');
```

4.3.7. Параметр `ja_seceventlog.role`

Данный параметр используется для аудита объектов БД.

В версии компонента 3.x, параметр «`ja_seceventlog.role`» исключен.

Команды `SELECT`, `INSERT`, `UPDATE` и `DELETE` для конкретного объекта будут регистрироваться в журнале событий безопасности, если они грантованы аудитору (той роли пользователей, для которой указано значение параметра `role = 'auditor'`).

Значение параметра `ja_seceventlog.role` по умолчанию:

- в версии 3.x – исключен;
- в версии 2.0 – " (пустое).

Параметр `ja_seceventlog.role` при обновлении компонента до версии 3.X можно заменить фильтрами или использовать совместно с имеющимися фильтрами.

К примеру, в версии компонента 2.0 был назначен аудит объекта:

```
SET ja_seceventlog.role = 'auditor';  
GRANT SELECT ON 'public.table1' TO auditor;  
GRANT INSERT ON 'public.table2' TO auditor;
```

В журнале событий безопасности регистрировались все запросы с `SELECT` для `'public.table1'` и все запросы `INSERT` для `'public.table2'`.

Тогда в версии компонента 3.x необходимо использовать один из вариантов:

- 1) Воспользоваться фильтрами следующего вида:

```
SELECT ja_seceventlog.create_filter('db1', '106139100',  
'TABLE', 'public.table1', NULL, 'SELECT');
```

```
SELECT ja_seceventlog.create_filter('db1', '106148100',  
'TABLE', 'public.table2', NULL, 'INSERT, UPDATE и DELETE');
```

Для использования данных фильтров необходимо установить параметр `ja_seceventlog.log_relation = on` (см. п.п. 4.3.6).

В фильтре название объекта БД, для которого необходимо регистрировать события, указывается совместно с именем схемы.

После настройки всех необходимых фильтров параметр `role` можно убрать из конфигурационного файла СУБД. Также можно отозвать выданные ранее полномочия выделенной роли аудитора из версии 2.0.

2) Параметр `ja_seceventlog.role` можно использовать совместно с фильтрами следующего вида:

```
SELECT ja_seceventlog.create_filter('db1', NULL, 'TABLE',  
'public.table1', NULL, 'все события для table1');  
  
SELECT ja_seceventlog.create_filter('db1', NULL, 'TABLE',  
'public.table2', NULL, 'все события для table2');
```

Для каждого объекта, для которого ведется аудит объекта необходимо создать соответствующий фильтр, аналогичный указанным выше.

4.3.8. Параметр `ja_seceventlog.log_parameter`

В версии 3.x параметр `ja_seceventlog.log_parameter` исключен из списка настраиваемых параметров. По умолчанию установлено значение `= true` означает: теперь `ja_seceventlog-3.1` в описание события безопасности всегда вносит информацию о параметрах SQL-запроса, если таковые были использованы и заданы в запросе. Это касается атрибута события "Parameter". В него будут записываться имена и значения параметров. Длина значения параметра контролируется параметром `ja_seceventlog.log_parameter_max_size`.

При обновлении компонента `ja_seceventlog` до версии 3.x данный параметр должен быть исключен и конфигурационного файла `postgresql.conf`.

4.3.9. Параметр `ja_seceventlog.log_statement`

В версии компонента 3.x, параметр «`ja_seceventlog.log_statement`» исключен.

Значение параметра `ja_seceventlog.log_statement` по умолчанию:

- в версии 3.x – исключен;
- в версии 2.0 – " (пустое).

Параметр указывает, будет ли протоколирование включать текст инструкции и параметры (если включено). В зависимости от требований журнал аудита может не требовать этого, и журналы становятся менее подробными.

```
ja_seceventlog.log_statement = on
```

4.3.10. Параметр `ja_seceventlog.log_statement_once`

В версии компонента 3.x, параметр «`ja_seceventlog.log_statement_once`» исключен.

Параметр `ja_seceventlog.log_statement_once` предназначен для включения/выключения записи текста запроса (или его части) в событие журнала безопасности только один раз при обработке сложных запросов для подвыражений.

Значение параметра `ja_seceventlog.log_statement_once` по умолчанию:

- в версии 3.x – исключен;
- в версии 2.0 – off.

При обновлении компонента `ja_seceventlog` до версии 3.x данный параметр не требует дополнительных действий в настройке.

4.3.11. Параметр `ja_seceventlog.log_client`

Параметр «`ja_seceventlog.log_client`» предназначался для определения, будут ли сообщения журнала событий безопасности отображаться для клиентского процесса, например такого как `psql`.

В версии 3.X параметр удален.

По умолчанию параметр был отключен.

При обновлении компонента `ja_seceventlog` до версии 3.x, если параметр использовался, то следует его удалить.

4.3.12. Параметр `ja_seceventlog.log_level`

Параметр `ja_seceventlog.log_level` предназначался для указания уровня важности сообщений событий безопасности, которые передаются клиенту.

В версии 3.X параметр удален.

При обновлении компонента `ja_seceventlog` до версии 3.X, если параметр использовался, то следует его удалить.

4.3.13. Параметр `ja_seceventlog.log_parameter_max_size`

Параметр `ja_seceventlog.log_parameter_max_size` определяет максимальный размер (в байтах) параметров, которые могут быть записаны в журнал. Параметры, превышающие указанный размер, заменяются на `<long param suppressed>`. При значении 0 ограничение отключается, и регистрируются все параметры без обрезки. Данная настройка ограничивает длину записываемых значений для `ja_seceventlog.log_parameter`.

Значение параметра `ja_seceventlog.log_level` по умолчанию:

- в версии 3.x – 0;
- в версии 2.0 – 0.

При обновлении компонента `ja_seceventlog` до версии 3.x данный параметр не требует дополнительных действий в настройке.

4.3.14. Параметр `ja_seceventlog.log_rows`

В версии компонента 3.x параметр `ja_seceventlog.log_rows` включен по умолчанию.

Атрибут с текстом выполняемого SQL-запроса будет всегда включаться в событие безопасности.

При обновлении компонента `ja_seceventlog` до версии 3.x данный параметр не требует дополнительных действий в настройке.

4.3.15. Параметр `ja_seceventlog.log_statement`

Параметр `ja_seceventlog.log_statement` предназначен для включения/выключения записи текста запроса (или его части) в событие журнала безопасности.

Значение параметра `ja_seceventlog.log_statement` по умолчанию:

- в версии 3.x – off;

– в версии 2.0 – on.

При обновлении компонента ja_seceventlog до версии 3.x данный параметр не требует дополнительных действий в настройке.

5. УДАЛЕНИЕ КОМПОНЕНТА

5.1. Удаление расширения

Расширение удаляется SQL-командой:

```
DROP EXTENSION ja_seceventlog;
```

В разделе «Shared Library Preloading» конфигурационного файла «postgresql.conf» целесообразно закомментировать строку или удалить имя загружаемой библиотеки «ja_seceventlog» из общего перечня загружаемых библиотек:

```
#shared_preload_libraries = 'ja_seceventlog'
```

5.2. Удаление пакета

Компонент удаляется при помощи штатных средств ОС управления пакетами:

```
apt-get purge jatoba6-ja-seceventlog
```

ПРИЛОЖЕНИЕ 1

(справочное)

Список типов объектов «object_type» компонента «ja_seceventlog»:

- ACCESS METHOD
- AGGREGATE
- CAST
- COLLATION
- CONVERSION
- DATABASE
- DOMAIN
- EVENT TRIGGER
- EXTENSION
- FOREIGN DATA WRAPPER
- FOREIGN TABLE
- FUNCTION
- GROUP
- INDEX
- LANGUAGE
- MATERIALIZED VIEW
- OPERATOR
- OPERATOR CLASS
- OPERATOR FAMILY
- POLICY
- PROCEDURE
- PUBLICATION

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

- ROLE
- RULE
- SCHEMA
- SEQUENCE
- SERVER
- STATISTICS
- SUBSCRIPTION
- TABLE
- TABLE AS
- TABLESPACE
- TEXT SEARCH CONFIGURATION
- TEXT SEARCH DICTIONARY
- TEXT SEARCH PARSER
- TEXT SEARCH TEMPLATE
- TRANSFORM
- TRIGGER
- TYPE
- USER
- USER MAPPING
- VIEW

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

SQL	–	Structured Query Language
БД	–	База данных
ОС	–	Операционная система
СУБД	–	Система управления базами данных

[illegible]

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