Базы данных

- PostgreSQL
- ClickHouse (сбор событий аудита БД)

PostgreSQL

> Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

> Официальная документация по данному разделу приведена в Онлайн-справке на продукт: https://support.kaspersky.com/help/KUMA/2.1/ru-RU/251880.htm

Установка плагина pgAudit

Чтобы установить плагин pgAudit:

1. В командном интерпретаторе выполните команды под учетной записью с правами администратора:

```
sudo apt update
sudo apt -y install postgresql-<версия базы данных PostgreSQL>-pgaudit
```

Версию плагина необходимо выбрать в зависимости от версии СУБД PostgresSQL.
Информацию о версиях СУБД PostgreSQL и необходимых версиях плагина см.по ссылке:
https://github.com/pgaudit/pgaudit#postgresql-version-compatibility.

> Пример:
> `sudo apt -y install postgresql-12-pgaudit`

2. Найдите конфигурационный файл `postgres.conf`. Для этого **в командной строке СУБД PostgresSQL** выполните команду:

```
SHOW config_file;
```

В ответе будет указано расположение конфигурационного файла.

3. Создайте резервную копию конфигурационного файла `postgres.conf`.

Откройте файл `postgres.conf` и скопируйте или замените имеющиеся значения на указанные ниже.

```
## pgAudit settings
shared_preload_libraries = 'pgaudit'
## database logging settings
log_destination = 'syslog'
## syslog facility
syslog_facility = 'LOCAL0'
## event ident
syslog_ident = 'Postgres'
## sequence numbers in syslog
syslog_sequence_numbers = on
## split messages in syslog
syslog_split_messages = off
## message encoding
lc_messages = 'en_US.UTF-8'
## min message level for logging
client_min_messages = log
## min error message level for logging
log_min_error_statement = info
## log checkpoints (buffers, restarts)
log_checkpoints = off
## log query duration
log_duration = off
## error description level
log_error_verbosity = default
## user connections logging
log_connections = on
## user disconnections logging
log_disconnections = on
## log prefix format
log_line_prefix = '%m|%a|%d|%p|%r|%i|%u| %e '
## log_statement
log_statement = 'none'
## hostname logging status. dns bane resolving affect
#performance!
log_hostname = off
## logging collector buffer status
#logging_collector = off
## pg audit settings
pgaudit.log_parameter = on
pgaudit.log='ROLE, DDL, MISC, FUNCTION'
```

4. Перезапустите службу СУБД PostgreSQL при помощи команды:

```
sudo systemctl restart postgresql
```

5. Чтобы загрузить плагин pgAudit в СУБД PostgreSQL, в командной строке СУБД PostgreSQL выполните команду:

```
CREATE EXTENSION pgaudit;
```

Настройка Syslog для отправки событий

Чтобы настроить передачу событий от сервера, на котором установлена PostgreSQL, в коллектор:

1. Проверьте, что на сервере источника событий установлен сервис rsyslog. Для этого выполните следующую команду:

```
sudo systemctl status rsyslog.service
```

Если сервис rsyslog не установлен на сервере, установите его, выполнив следующие команды:

```
yum install rsyslog
sudo systemctl enable rsyslog.service
sudo systemctl start rsyslog.service
```

В каталоге `/etc/rsyslog.d/` создайте файл `pgsql-to-siem.conf` со следующим содержанием:

```
If $programname contains 'Postgres' then @<IP-адрес коллектора>:<порт коллектора>
```

Если вы хотите отправлять события по протоколу TCP, содержимое файла должно быть таким:

```
If $programname contains 'Postgres' then @@<IP-адрес коллектора>:<порт коллектора>
```

В конфигурационный файл `/etc/rsyslog.conf` добавьте следующие строки:

```
$IncludeConfig /etc/rsyslog.d/pgsql-to-siem.conf
$RepeatedMsgReduction off
```

Перезапустите сервис rsyslog, выполнив следующую команду:

```
sudo systemctl restart rsyslog.service
```

Настройка KUMA

После того как параметры передачи событий настроены, требуется создать коллектор в веб-интерфейсе KUMA для событий pgAudit.

1. На шаге **Транспорт** укажите тип и порт в соответствии с настройками на стороне pgAudit.

2. На шаге **Парсинг** событий выберите нормализатор **[OOTB] PostgreSQL pgAudit syslog.**

3. На шаге **Маршрутизация** проверьте, что в набор ресурсов коллектора добавлены следующие точки назначения:

- **Хранилище**. Для отправки обработанных событий в хранилище.
- **Коррелятор**. Для отправки обработанных событий в коррелятор.

Если точки назначения **Хранилище** и **Коррелятор** не добавлены, создайте их.

4. На шаге **Проверка параметров** нажмите **Сохранить и создать сервис**.

5. Скопируйте появившуюся команду для установки коллектора KUMA.

Полезные ссылки

Настройка получения событий pgAudit (онлайн-справка KUMA):

https://support.kaspersky.com/help/KUMA/2.1/ru-RU/251880.htm

ClickHouse (сбор событий аудита БД)

> Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

Для настройки базового аудита Clickhouse понадобится:

1. Для логирования обычных запросов (в том числе grant, create, drop) включить логирование в основном файле (по умолчанию включено):

```
/etc/clickhouse-server/config.xml
```

в секции logger как минимум необходимо задать формат information

```
########################################
<clickhouse>
    <logger>
        <level>information</level>
########################################
```

2. Для логирования подключений, таких как:

- Login пользователя
- Failure logon
- Logout пользователя

потребуется создать отдельный файл

```
/etc/clickhouse-server/config.d/session_log.xml
```

с содержимым:

```
<clickhouse>
    <session_log>
        <database>system</database>
        <table>session_log</table>
```

```
        <flush_interval_milliseconds>7500</flush_interval_milliseconds>
    </session_log>
</clickhouse>
```

И отдельный файл

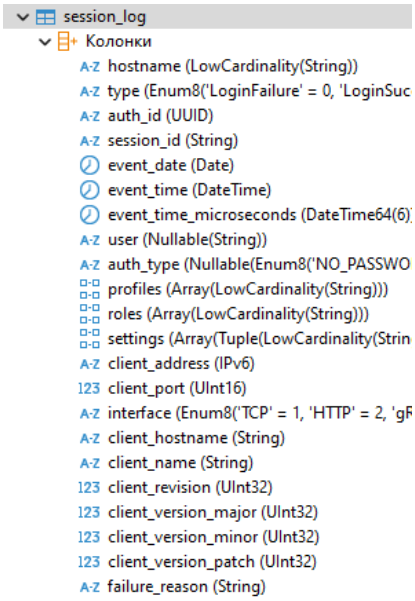
```
/etc/clickhouse-server/config.d/listen_port.xml
```

С содержимым (для возможности удалённого подключения к БД):

```
<clickhouse>
  <listen_host>::</listen_host>
</clickhouse>
```

3. Перезапустить службу командой systemctl restart clickhouse-server

В базе system появится новая таблица со следующими колонками:



Создаём пользователя для подключения к БД для KUMA и выдаём ему необходимые права следующим командами (с использованием ранее настроенного аудита):

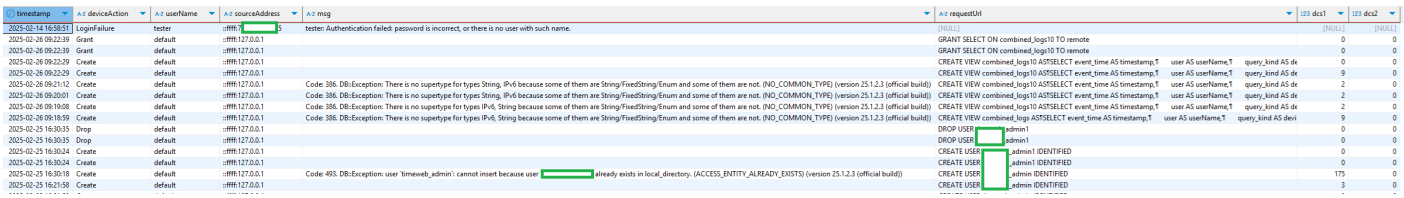
```
###Шаг 1. Создание пользователя для коллектора KUMA
CREATE USER kuma HOST IP '10.10.10.1/32' IDENTIFIED WITH sha256_password BY
'supersecretpassword';

####Шаг 2. Создание VIEW для вывода нескольких запросов
CREATE VIEW combined_logs
```

```
AS SELECT
    event_time AS timestamp,
    query_kind AS deviceAction,
    user AS userName,
    toString(initial_address) AS sourceAddress,
    exception AS msg,
    query AS requestUrl,
    query_duration_ms AS dcs1,
    memory_usage AS dcs2
FROM system.query_log
WHERE (query_kind IN ('Grant', 'Create', 'Drop')) OR (query_duration_ms > 600000) OR
(memory_usage > 1000000000)
UNION ALL
SELECT
    event_time AS timestamp,
    type AS deviceAction,
    user AS userName,
    toString(client_address) AS sourceAddress,
    failure_reason AS msg,
    NULL AS requestUrl,
    NULL AS dcs1,
    NULL AS dcs2
FROM system.session_log
WHERE type IN ('LoginFailure', 'LoginSuccess')
ORDER BY timestamp DESC

####Шаг 3. Назначение прав на VIEW
GRANT SELECT ON combined_logs TO kuma
GRANT SELECT ON system.session_log TO kuma
GRANT SELECT ON system.query_log TO kuma
```

Пример как выглядит вывод VIEW:



Настройка инстанса завершена, можно приступать к подключению логов в KUMA.

Итого данной View мы выводим следующие события:

- Login пользователя
- Failure logon
- Logout пользователя
- Длительный запрос в базу (более 10 минут, как пример)(условие query_duration_ms > '600000' в запросе)
- Большое потребление памяти при запросе (более 1 Гб, как пример)(условие memory_usage > '1000000000')
- Создание пользователя
- Назначение прав пользователю
- Удаление пользователя

В KUMA необходимо создать коллектор с транспортом sql (плейсхолдер для Clickhouse - ?) и параметрами как на скриншоте (для примера указаны разные версии):

Transport

Add a source from which you want to receive events. For details see **Online Help**.

Basic settings | Advanced settings

Connector: Create new

Kind*: sql

Default query*:

```
SELECT * FROM combined_logs
WHERE timestamp > parseDateTimeBestEffort(?)
```

Reconnect to DB each time the request is sent: off

Poll interval, sec: 0

Connection

Database kind*: ClickHouse

URL*: clickhouse:// 10.10.10.10:9000

Authorization*: Plain

Secret*: kuma_clickhouse

TLS mode: Disabled

Identity column*: timestamp

Identity seed*: 2025-12-22 21:46:18

Query: Add a query that you want to use instead of a default query

Poll interval, sec: 0

Корректный запрос:

```
SELECT * FROM combined_logs
WHERE timestamp > parseDateTimeBestEffort(?)
```

Выбираем нормализатор Clickhouse (доступен в Community pack), добавляем необходимые точки назначения и инсталлируем службу коллектора.

Либо, если нам **не нужны события входа в БД** можем использовать запрос только в таблицу по умолчанию:

```
###Шаг 1. Создание пользователя для коллектора KUMA
CREATE USER kuma HOST IP '10.10.10.1/32' IDENTIFIED WITH sha256_password BY
'supersecretpassword';

####Шаг 2. Назначение прав на выполнение SELECT к system.query_log
GRANT SELECT ON system.query_log TO kuma

####Шаг 3. Используем запрос для KUMA коллектора
SELECT event_time AS timestamp,
       query_kind AS deviceAction,
       user AS userName,
       toString(initial_address) AS sourceAddress,
       exception AS msg,
       query AS requestUrl,
   query_duration_ms AS dcs1,
   memory_usage AS dcs2
FROM system.query_log
WHERE query_kind IN ('Grant', 'Create', 'Drop') OR query_duration_ms > '600000' OR
memory_usage > '1000000000' AND timestamp > ?
ORDER BY timestamp DESC
```