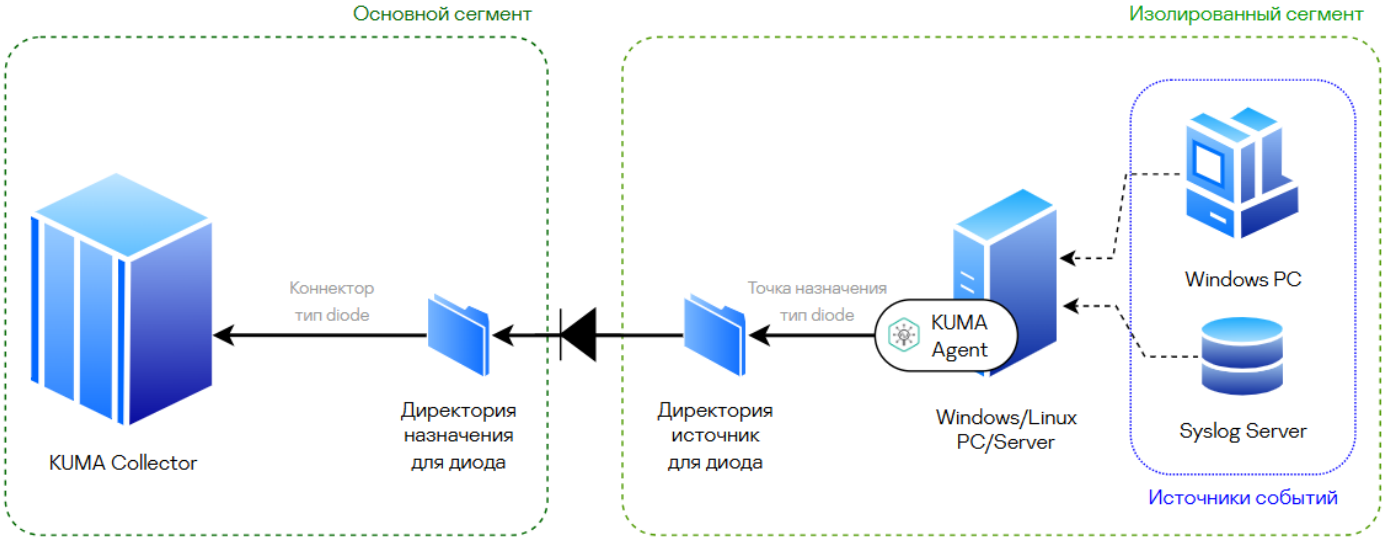
????????? ?????? ? ?????? ???? (Diode)

????? ?????? ????? ? ?????? diode



Агент, находящийся в изолированном сегменте сети, собирает события с источников и перемещает их в директорию источник, откуда их забирает диод (дата-диод) *(сторонний производитель, вне решения KUMA)*. Диод переносит файлы в директорию назначения основной сети, удаляя их директории источника. Из директории назначения события собирает коллектор, удаляя их после считывания.

Для осуществления описанного способа передачи событий через диод используется пара destination и connector типа diode, destination на агенте и connector на коллекторе. Агент может иметь любые из возможных типов connector, а коллектор - любые из возможных destination.

Агент при работе накапливает события в буфер. Как только буфер становится размером >=bufferSize (по умолчанию 64 Мб), или с момента предыдущей записи буфера в файл проходит > FlushInterval (по умолчанию 10 сек):

- Агент записывает события в файл во временной директории, указанную пользователем
- Агент переносит файл из временной папки в "Директорию, из которой диод данных получает события (Data diode source directory)" , попутно переименовывая файл. Название файла содержит sha256 хеш содержимого для возможности осуществления проверки целостности.

Точки назначения

Основные параметры | Дополнительные параметры

Точка назначения	Создать
Название*	destinationDiode
Состояние	ON
Тип*	diode
Директория, из которой диод данных получает события*	/data
Временная директория*	/temp

По умолчанию сжатие файлов не происходит, но его можно осуществлять, если выбрать в настройках destination данную опцию. В этом случае для корректной работы тот же алгоритм должен быть указан в соответствующем diode connector.

При считывании (diode connector) sha256 хеш содержимого файла сравнивается с хешем из имени файла, при несоответствии файл удаляется и создается событие аудита.

> Ресурс точки назначения в агенте должен иметь тип diode. В этом ресурсе необходимо указать директорию, из которой диод данных будет перемещать файлы во внешний сегмент сети.

> Для diode-агента невозможно выбрать коннекторы типа sql или netflow.

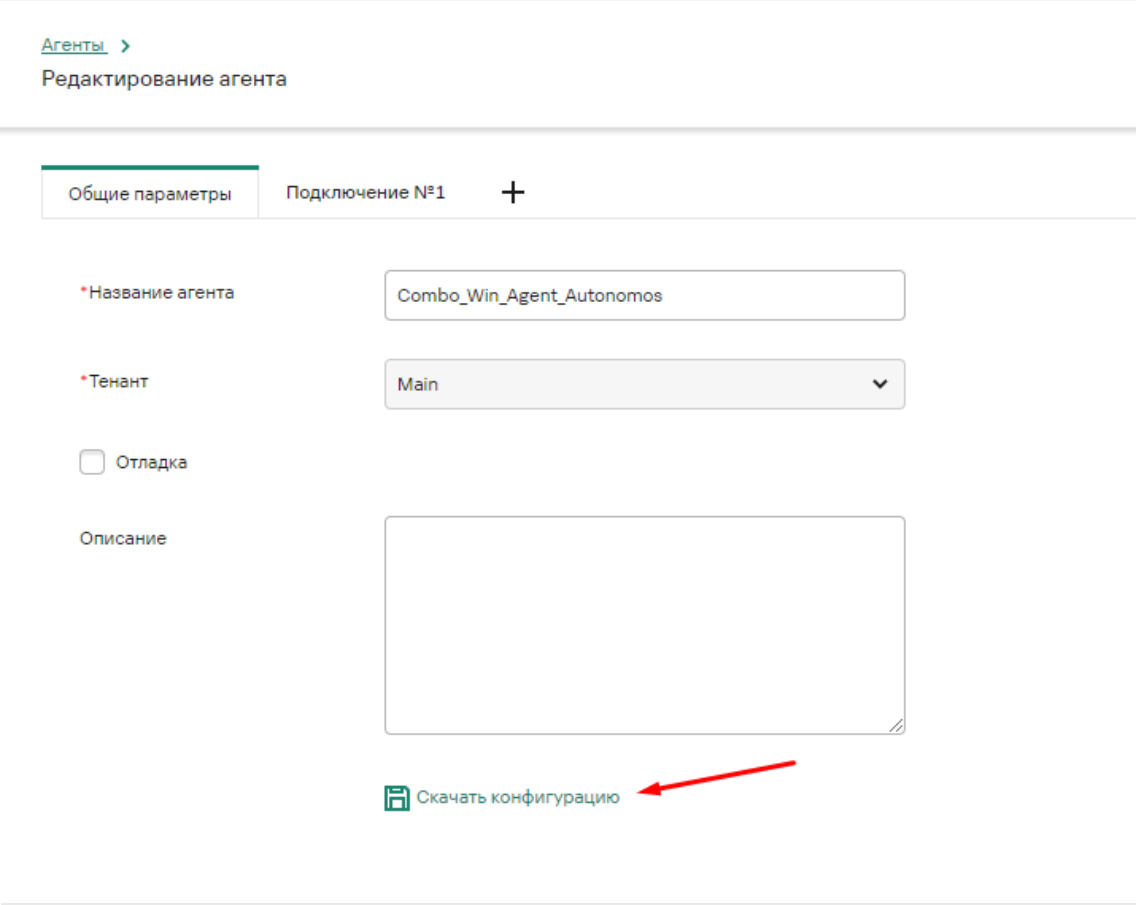
????????????????? ????

Конфигурацинный файл агента сохраняется в человекочитаемом виде для возможности добавления секретов вручную. Для избежания сохранения секретов в открытом виде, содержимое реальных секретов заменяется на шаблоны соответствующего типа секрета, также в конфигурационном файле генерируются шаблоны в местах, где это явно указано (см п. Шаблоны секретов). В ресурсах внутри агента, использующих секреты указан UUID секрета, содержимое секретов находится отдельно в поле secrets. Данные секретов можно заполнить вручную в конфигурационном файле, изменяя поля секретов.

Далее в таблице описаны поля секрета.

Имя поля	Тип	Описание
user	строка	Имя пользователя
password	строка	Пароль
token	строка	Токен
urls	массив строк	Список url
publicKey	строка	Публичный ключ (используется в PKI)
privateKey	строка	Приватный ключ (используется в PKI)
pfx	строка, содержащая base64 закодированное содержимое pfx	Содержимое pfx файла, закодированное в base64. На linux получить base64 кодировку файла можно при помощи команды base64 -w0 src > dst
pfxPassword	строка	Пароль от pfx
securityLevel	строка	Используется в snmp3. Возможные значения: NoAuthNoPriv, AuthNoPriv, AuthPriv
community	строка	Используется в snmp1
authProtocol	строка	Используется в snmp3. Возможные значения: MD5, SHA, SHA224, SHA256, SHA384, SHA512
privacyProtocol	строка	Используется в snmp3. Возможные значения: DES, AES
privacyPassword	строка	Используется в snmp3
certificate	строка, содержащая base64 закодированное содержимое pem	Содержимое pem файла, закодированное в base64. На linux получить base64 кодировку файла можно при помощи команды base64 -w0 src > dst

Конфигурационный файл скачивается из веб-интерфейса ядра KUMA в части настроек агента:



?????? ?????? (?????????? ?????)

> При установке агента его конфигурационный файл не должен находиться в директории, в которую устанавливается агент.

>

> Необходимо при помощи списка контроля доступа (ACL) настроить права доступа к конфигурационному файлу так, чтобы доступ на чтение файла был только у пользователя, под которым будет работать агент.

> TLS не работает, тк требуется подключение к ядру.

Справочная информация об установщике доступна по команде: `kuma.exe help agent`

Linux

Примите лицензионное соглашение: `/opt/kaspersky/kuma/kuma license`

Для запуска агента требуется скопировать файл /opt/kaspersky/kuma/kuma с машины, где установлена KUMA на машину с linux, где будет запущен агент и запустить его: `./kuma agent --cfg <path to config file> --wd <path to working directory>`
Через опцию --wd указывается путь для хранения файлов агента, по умолчанию они будут храниться в текущей директории.

Конфигурационный файл может содержать секреты, его следует защищать при помощи ACL, позволяющих чтение только пользователю KUMA (600).

Windows

Примите лицензионное соглашение: `/opt/kaspersky/kuma/kuma.exe license`

Без установки

```
kuma.exe agent --cfg <path to config file>
```

```
kuma.exe  agent --cfg <путь к конфигурационному файлу агента> --wd <путь к директории, где будут размещаться файлы устанавливаемого агента. Если не указывать этот флаг, файлы будут храниться в директории, где расположен файл kuma>
```

С установкой

```
kuma.exe agent --cfg <path to config file> --user <user to start service as> --install
```

При установке используемый конфигурационный файл перемещается в рабочую директорию (`ProgramData\Kaspersky Lab\KUMA\agent\<serviceID>\`) (ID берется из ресурса агента в конфигурационном файле), `kuma.exe` перемещается в рабочую директорию (`Program Files\Kaspersky Lab\KUMA`).

В дальшейшем используется перемещенный конфигурационный файл.

Конфигурационный файл может содержать секреты, его следует защищать при помощи ACL, позволяющих чтение только пользователю от чьего лица запускается KUMA.

????????

```
kuma.exe agent --cfg <path to config file> --uninstall
```

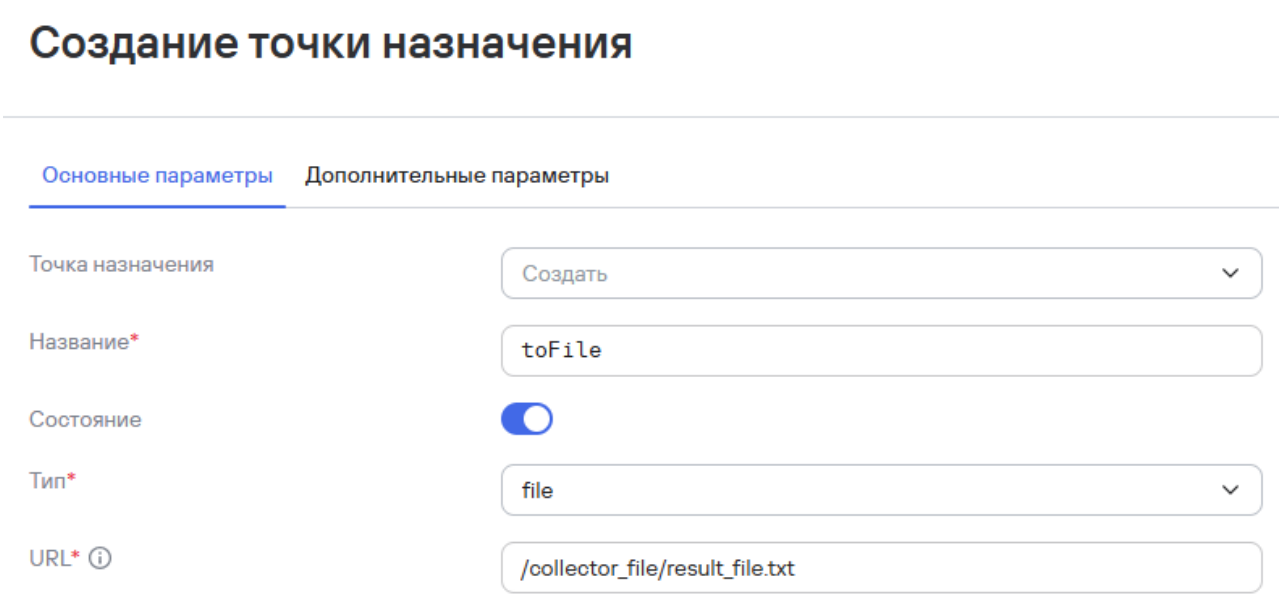
или

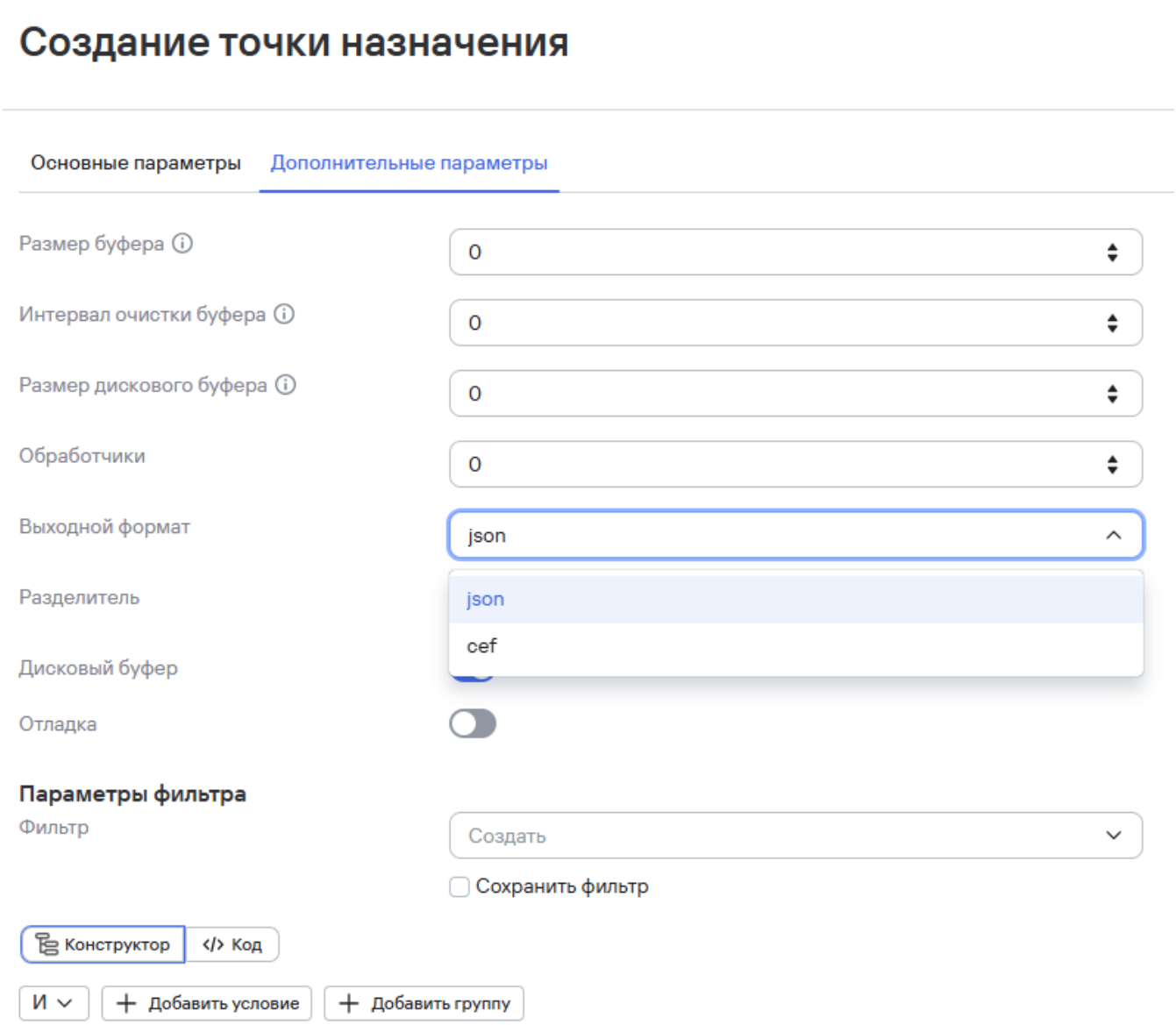
```
kuma.exe agent --id <идентификатор сервиса агента, созданного в KUMA> --uninstall
```

???? ??????????? ?? ??????????????? ????????

Для установки коллектора (т.к. агент поддерживает не все типы коннекторов) необходима связь с ядром, для этого ставим дополнительное отдельное ядро в изолированный сегмент, которое будет управлять этим коллектором. Далее коллектор в маршрутизации пишет результат обработки события НЕ в хранилище, а в файл (например: результирующий файл по работе с БД это JSON формат или CEF). Предварительно необходимо создать путь (папку) и дать права для пользователя kuma:

```
chown -R kuma:kuma /collector_file/
chown -R kuma:kuma /collector_file/result_file.txt
```





Далее этот файл забирает автономный агент и отправляет по диоду в корп сегмент по схеме в начале статьи.

Revision #15
Created 11 August 2023 14:23:49 by Boris Rzr
Updated 5 September 2025 07:45:03 by Boris Rzr