INFOWATCH

InfoWatch Traffic Monitor

Руководство администратора

05/10/2023

Тел./Факс +7 (495) 229-00-22

http://www.infowatch.ru

СОДЕРЖАНИЕ

1 Введение

В настоящем руководстве содержатся сведения по администрированию InfoWatch Traffic Monitor: настройке системы после установки, восстановлению после сбоев, проведению регламентных работ.

1.1 Аудитория

Документ предназначен для администраторов InfoWatch Traffic Monitor Server, знакомых с основами работы в среде операционных систем Microsoft Windows и Linux, а также обладающих навыками администрирования СУБД Oracle и Postgre SQL.

1.2 Комплект документов

В комплект документации по InfoWatch Traffic Monitor входят:

- «*InfoWatch Traffic Monitor. Руководство по установке*»

Содержит описание порядка установки, настройки, обновления и удаления системы InfoWatch Traffic Monitor.

- «*InfoWatch Traffic Monitor. Руководство администратора*».

Содержит информацию по администрированию Системы (база данных, серверная часть).

- «*InfoWatch Traffic Monitor. Руководство пользователя*».

Содержит описание порядка работы с InfoWatch Traffic Monitor (настройка конфигурации, экспорт/импорт данных, составление политик для обработки объектов).

- «*InfoWatch Traffic Monitor. Справочник по конфигурационным файлам*».

Содержит пояснения к часто используемым конфигурационным файлам.

1.3 Техническая поддержка пользователей

При возникновении проблем и вопросов, связанных с работой Системы, вы можете обратиться в службу технической поддержки:

- если вы приобрели продукт у партнера компании InfoWatch, то обратитесь в службу технической поддержки партнера.
- если продукт приобретен у компании InfoWatch напрямую, то обратитесь в службу технической поддержки компании InfoWatch по адресу support@infowatch.com.

Часы работы Службы технической поддержки – с 7:00 до 21:00 по московскому времени с понедельника по пятницу, исключая официальные выходные и праздничные дни в РФ.
Вы также можете посетить раздел технической поддержки на нашем сайте:
https://www.infowatch.ru/services/support
Перед обращением в службу технической поддержки рекомендуется посетить раздел База знаний на нашем сайте: https://kb.infowatch.com/. Возможно, там уже содержится ответ на интересующий вас вопрос или описано решение возникшей у вас проблемы.

2 Обзор Traffic Monitor

В этой главе:

2.1 Функции InfoWatch Traffic Monitor

InfoWatch Traffic Monitor позволяет контролировать информационные потоки в корпоративной среде для выявления и предотвращения случаев несанкционированного использования конфиденциальных данных.

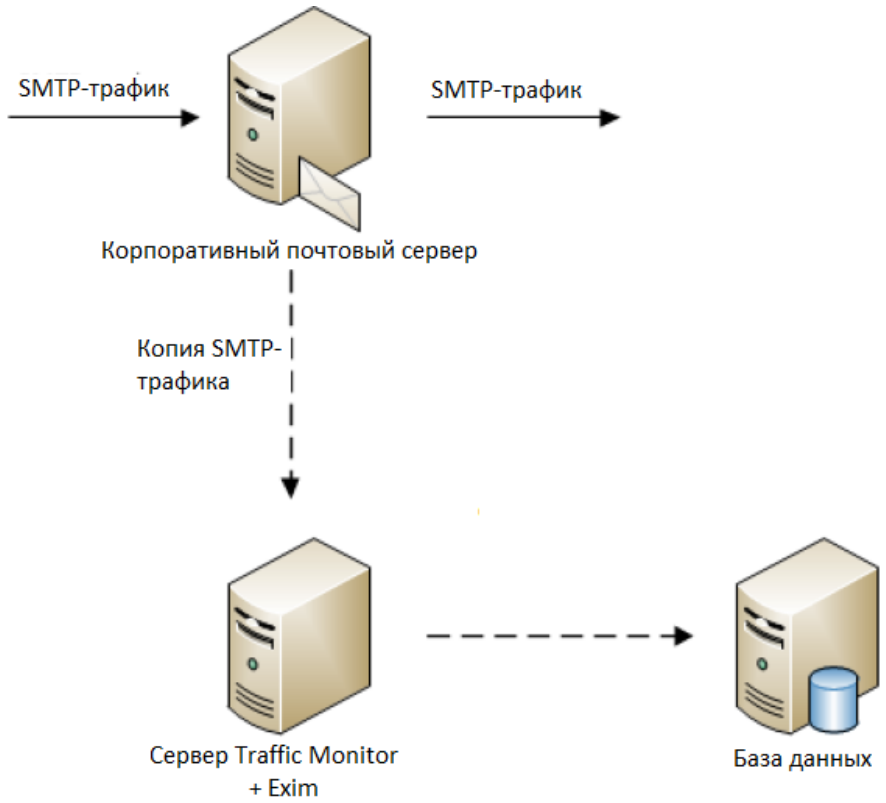
Основные функции InfoWatch Traffic Monitor:

- Перехват в потоке/на шлюзе трафика, передаваемого по протоколу SMTP, с учетом мандатных меток;
- Перехват трафика на рабочих станциях агентами Device Monitor (подробнее см. *"InfoWatch Device Monitor. Руководство пользователя"*).

- Анализ содержимого перехваченного трафика с целью выявления нарушений корпоративной политики безопасности;
- Фильтрация перехваченного трафика путем выдачи разрешения/запрещения на доставку определенных данных.

2.1.1 Схема перехвата SMTP-трафика

На корпоративном почтовом сервере требуется настроить правило, отправляющее скрытую копию (BCC) для каждого отправленного письма. Копия должна отравляться на несуществующий почтовый адрес почтового домена, IP-адрес которого соответствует серверу Traffic Monitor. Данная функция поддерживается большинством почтовых серверов.



О настройке данного функционала см. "Прием копий с почтового сервера"
Преимущества:

- Позволяет анализировать не только внешнюю, но и внутреннюю переписку компании;
- Гарантирует анализ всех писем.

Недостатки:

- Требует внесения изменений в настройки корпоративного почтового сервера;
- Никак не контролируются внешние почтовые серверы (если их использование разрешено);
- При недоступности сервера Traffic Monitor, пользователи получат сообщение об ошибке доставки скрытой копии;
- Дополнительная нагрузка на почтовый сервер.

2.2 Состав InfoWatch Traffic Monitor (Astra Linux)

Подсистема InfoWatch Traffic Monitor	Назначение подсистемы
Подсистема перехвата трафика	Перехват и передача на обработку трафика (объектов или их копий) осуществляется агентами подсистемы Device Monitor.
Подсистема обработки	Извлечение из перехваченных объектов значимой информации и вложений, определение форматов вложений и передача извлеченных текстов в подсистему анализа. **Примечание:** при создании объекта составляется его XML-

Подсистема InfoWatch Traffic Monitor	Назначение подсистемы
	контекст – текстовый файл, включающий содержимое объекта и информацию о нем.
Подсистема анализа	Анализ текстовых данных, извлеченных из перехваченных объектов (текстов писем, сообщений, запросов, а также текстов, извлеченных из вложений). Состоит из следующих технологий: • Категории и термины; • Текстовые объекты; • Эталонные документы; • Векторные изображения; • Бланки; • Печати; • Выгрузки из БД; • Графические объекты; • Автолингвист.
Подсистема применения политик	На основе результатов работы подсистемы анализа и подсистемы обработки выносит вердикт о факте нарушения или не нарушения перехваченным объектом политики информационной безопасности. Также обеспечивает привязку данных о получателе или отправителе объекта к записям справочника сотрудников и рабочих станций. Состоит из следующих модулей: Модуль интеграции с Astra Linux Directory, Модуль принятия решений
Подсистема хранения	Хранение информации о перехваченных объектах, результатах их анализа и применения политик, а также предоставление возможности для просмотра хранящейся информации посредством запросов из консоли управления. Представляет собой базу данных. Состоит из следующих модулей: Модуль взаимодействия с удаленной БД, Модуль загрузки объектов в БД, Модуль хранения настроек системы, Модуль хранения объектов. **Примечание:** перед сохранением объектов в БД, они преобразуются из одного внутреннего формата (XML) в другой внутренний формат.
Подсистема мониторинга	Возможность удаленного мониторинга состояния серверов, на которых установлены компоненты InfoWatch Traffic Monitor, и работающих на них служб. Также выполнение общих действий по управлению сервером. Работа с подсистемой осуществляется администратором через веб-интерфейс.
Подсистема аудита	Возможность настраивать поисковые фильтры (по персоне, действию, объекту, датам), получать краткую наглядную информацию по событиям и нарушениям

Подсистема InfoWatch Traffic Monitor	Назначение подсистемы
	согласно установленным фильтрам,а также устанавливать период хранения событий в Системе. Работа с подсистемой осуществляется администратором через веб-интерфейс.
Подсистема «Консоль управления»	Обеспечение работы графического пользовательского интерфейса, с помощью которого производится администрирование, настройка и использование Traffic Monitor. Состоит из следующих модулей: • Модуль мониторинга; • Модуль контроля; • Модуль настройки.

2.3 Лицензирование

После установки Системы необходимо также установить лицензию.
Для этого запросите файл лицензионного ключа (см. документ "InfoWatch Traffic Monitor. Руководство пользователя", раздел "Запрос лицензии").
Лицензия определяет срок действия, количество пользователей, набор модулей перехвата и модулей анализа, а также возможность взаимодействия со сторонними системами.

Лицензионный ключ представляет собой файл формата LIC.
При установке и использовании лицензионного ключа нужно учитывать следующее:

- Если период действия лицензии истек, работа перехватчиков будет остановлена. Для возобновления работы Системы установите новую лицензию.
- Если требуется изменить настройки передачи трафика согласно новой схеме развертывания, замените лицензионный ключ с учетом новых перехватчиков.

При полной переустановке операционной системы или системы InfoWatch Traffic Monitor вам потребуется заново установить лицензию. Поэтому рекомендуется сохранить файл лицензионного ключа на каком-либо носителе информации.

Важно!

О проверке валидности лицензии и об управлении лицензиями см. документ "InfoWatch Traffic Monitor. Руководство пользователя", раздел "Управление лицензиями".

Ниже приведены списки модулей, которые используются в продукте.

Модули перехвата, разработанные компанией InfoWatch		
Подсистема	Тип событий	Протокол
TM	Электронная почта	POP3 IMAP

Модули перехвата, разработанные компанией InfoWatch		
Подсистема	Тип событий	Протокол
		SMTP NRPC
	Web-сообщение	HTTP HTTPS
	Web-почта	HTTP HTTPS
	ICQ	OSCAR
	Data Discovery	-
DM	Электронная почта	POP3 IMAP Outlook SMTP
	Web-сообщение	HTTP HTTPS
	Web-почта	HTTP HTTPS
	Печать	-
	Skype	SKYPE
	XMPP	XMPP
	Mail.Ru Агент	MMP
	FTP	FTP
	Буфер обмена	-
	Снимки экрана	-
	Съемные устройства	-

Модули перехвата, разработанные компанией InfoWatch		
Подсистема	Тип событий	Протокол
	Сетевые ресурсы	-
	Терминальная сессия	-
	Облачное хранилище	HTTPS
	Telegram	-
	Ввод с клавиатуры	-
	Vkontakte	HTTPS
	Facebook	HTTPS
	WhatsApp	-
Adapters	MS Lync	SIP
	Электронная почта (Lotus)	NRPC
	Съемные устройства (Lumension Device Control)	-
	Печать (Lumension Device Control)	-
	Web-сообщение (ICAP)	ICAP
	Web-почта (ICAP)	ICAP

Модули перехвата сторонних разработчиков	
Тип событий	Протокол
Электронная почта	POP3 MAPI SMTP/ESMTP IMAP NRPC
Web-почта	HTTP HTTPS ICAP

Модули перехвата сторонних разработчиков	
Тип событий	Протокол
ICQ	OSCAR
Mail.ru Агент	MMP
Skype	Skype
XMPP	XMPP
MS Lync	SIP
Web-сообщение	HTTP HTTPS ICAP
FTP	FTP
Съемные устройства	-
Печать	-
SmartLogger	XMPP
Cisco UCM	XMPP

Модули анализа:

1. Лингвистический анализ;
2. Детектор эталонных документов;
3. Детектор векторных изображений;
4. Детектор текстовых объектов;
5. Детектор выгрузок из баз данных;
6. Детектор форм;
7. Графический анализ;
8. Детектор печатей;
9. Автолингвист.

Дополнительные возможности:

1. Модули автообновления эталонных выгрузок;
2. Экспорт в сторонние системы анализа.

Ограничение:

Технология блокирования каналов утечки на рабочих станциях по результатам анализа не лицензируется.

3 Типы установки Системы

Развертывание Системы осуществляется следующими способами, исходя из расчетной нагрузки на аппаратные средства и цели внедрения (подробнее см. документ "*InfoWatch Traffic Monitor. Руководство по установке*", статья "Схемы развертывания Системы и выбор типа установки"):

- *"Все-в-одном" Enterprise* - тип установки Системы с расширенными возможностями, включая: использование СУБД PostgreSQL, настройку масштабируемости, тонкий контроль за рабочими станциями. Подробнее см. документ "*InfoWatch Traffic Monitor. Руководство по установке*", статья "Установка TM Enterprise и TM Standard в режиме "Все-в-одном";
- *Распределенная установка TM Enterprise (*компоненты Traffic Monitor и СУБД устанавливаются на разные машины*)* - тип установки Системы для функционирования под большой нагрузкой и работой с большим объемом данных. Подробнее см. документ "*InfoWatch Traffic Monitor. Руководство по установке*", статья "Распределенная установка TM Enterprise".

Важно!

В случае распределенной установки Системы на разные серверы (или при создании кластера серверов) вводится ряд дополнительных ограничений и настроек:

- необходимо настроить сетевые параметры поисковика Sphinx (подробнее см. "Конфигурирование работы Sphinx при распределенной установке")
- некоторые процессы серверной части (**iw_adlibitum, iw_bookworm, iw_deliver, iw_licensed, iw_indexer, iw_is, iw_image_autoling, iw_text_autoling**) и пользовательской консоли (**iw_kicker, iw_configerator**) должны быть запущены в единственном экземпляре на кластере (подробнее см. "Список процессов серверной части Traffic Monitor")

Каждый из типов установки, в зависимости от приобретаемой лицензии, может включать установку перехватчиков Системы:

- InfoWatch Device Monitor - предназначен для настройки схем безопасности, системы мониторинга компьютеров, контроля доступа к компьютерам компании и др. (подробнее см. документ "*InfoWatch Traffic Monitor. Руководство по установке*", статья "Установка InfoWatch Device Monitor").
- Adapters - модули перехвата для интеграции со сторонними системами.

4 Настройка Системы после установки

После установки Системы выполняются следующие настройки, необходимые для штатного функционирования Системы:

- Настройка синхронизации времени - о включении автоматической синхронизации времени на серверах;
- Конфигурирование работы Sphinx при распределенной установке - о настройках для распределенной установки;
- Настройка перехвата трафика - особенности настроек для различных типов перехватываемого трафика;
- Автозапуск процессов - перечень системных процессов и описание необходимости и порядка включения и отключения их автозапуска;
- Настройка OCR-экстракторов - порядок установки пакетов, необходимых для распознавания текста в перехваченных событиях;
- Настройка отправки уведомлений пользователям и сотрудникам - обязательные настройки для поддержки почтовых уведомлений, отправляемых в результате срабатывания тех или иных правил в политиках (подробнее см. документ «*InfoWatch Traffic Monitor. Руководство пользователя*»);
- Ограничение количества найденных событий - изменение максимального количества событий, выводимых в Консоли управления;
- Настройка Сервера InfoWatch Device Monitor - настройки отдельных модулей Сервера Device Monitor и изменение настроек протоколирования (подробнее см. документ *"InfoWatch Device Monitor. Руководство по установке, конфигурированию и администрированию"*);
- Настройка сбора данных в филиальной сети - настройки конфигурационных файлов служб, осуществляющих сбор данных в филиальной сети;
- Настройка межсервисного взаимодействия (служба Consul) - настройки конфигурационного файла и службы, осуществляющей управление процессами Системы;
- Обнаружение рабочих станций для установки агентов - сканирование сети на наличие рабочих станций вне Active Directory (подробнее см. документ *"InfoWatch Device Monitor. Руководство пользователя"*).

Рекомендуется:

- изменить предустановленные в Системе пароли, следуя требованиям информационной безопасности (подробнее в статье Изменение предустановленных паролей в Системе);
- настроить защиту от подбора паролей при авторизации в консолях Traffic Monitor и Device Monitor (подробнее в статье Настройка защиты от подбора паролей);
- настроить сетевой экран, чтобы предотвратить несанкционированный доступ к компонентам Системы (подробнее в статье Рекомендации по настройке безопасного окружения).

4.1 Изменение предустановленного пароля

Для учетных записей в Системе предустановлен стандартный пароль – xxXX1234 (подробнее в статье "Предустановленные серверные параметры" в *"InfoWatch Traffic Monitor. Руководство по установке"*). В процессе эксплуатации Системы его необходимо заменить, следуя требованиям информационной безопасности. Стабильная и безопасная работа Системы требует хранить пароли в надежном, недоступном для других месте.

Стандартный пароль изменяется в конфигурационных файлах Системы. Чтобы заменить предустановленный пароль на новый:

1. Остановите процессы Traffic Monitor:
   `iwtm stop`
2. Перейдите в директорию /opt/iw/tm5:
   `cd /opt/iw/tm5`
3. Выполните команду:
   `egrep -lir --include=\*.{cfg,conf} 'xxXX1234' | xargs -l sed -i -e 's/xxXX1234/<new_pass>/g'` где `<new_pass>` - новый пароль.
4. Измените предустановленные стандартные пароли используемой СУБД. Подробнее см. в статье "Изменение предустановленных паролей (PostgreSQL)".
5. Запустите процессы Traffic Monitor:
   `iwtm start`

При замене пароля следуйте рекомендациям, приведенным в статье "Приложение А. Рекомендации по составлению имен и паролей".

4.2 Настройка защиты от подбора паролей

Для защиты Системы реализована блокировка авторизации пользователя при повторяющихся неудачных попытках авторизации. Защита от подбора паролей настраивается в базе данных.

По умолчанию защита отключена.

Настройка выполняется отдельно для InfoWatch Traffic Monitor и для InfoWatch Device Monitor.

Чтобы включить защиту для сервера Traffic Monitor:

1. Подключитесь напрямую к базе данных, которую использует сервер Traffic Monitor.
2. Найдите таблицу `SETTING`.
3. Найдите 3 параметра, которые отвечают за настройку защиты:
   - **`web_logon_attempts_count`** - количество допустимых неудачных попыток авторизации;
   - **`web_logon_blocking_period`** - время, на которое будет заблокирована возможность авторизации для пользователя, в секундах;
   - **`web_logon_attempts_period`** - период неудачных попыток авторизации, в секундах.
4. Задайте значения всех трех параметров в соответствии с требованиями безопасности.

   > **Важно!**
   >
   > Если любому из параметров задано некорректное значение или 0, защита будет отключена.

5. Для применения настроек достаточно сохранить изменения в базе данных. Перезагрузка сервера не требуется, поддерживается динамическое изменение настроек защиты.

При блокировании авторизации разделе **Управление - Аудит** консоли управления Traffic Monitor будет создано соответствующее событие, в котором будет указано время разблокировки.
При очередной попытке авторизации заблокированный пользователь будет уведомлен о превышении числа неудачных попыток входа.

Пример настроек защиты для Traffic Monitor:

- `web_logon_attempts_count - 5`
- `web_logon_blocking_period - 300`
- `web_logon_attempts_period - 120`

При данных настройках, если в течение 2 минут (120 секунд) от имени пользователя будут выполнены 5 неудачных попыток авторизации, для данного пользователя будет заблокирована возможность авторизации на 5 минут (300 секунд). Отсчет периода неудачных попыток начинается с первой. Если за 120 секунд пользователь совершит только 4 неудачные попытки, следующая попытка будет первой в новом периоде.

4.3 Предварительные настройки

После установки системы необходимо выполнить следующие настройки:

- Настройка синхронизации времени
- Конфигурирование работы Sphinx при распределенной установке.

4.3.1 Настройка синхронизации времени

1. Установите системное время с помощью команды `date`. Например, для установки 11 сентября 2013 13:30 запустите команду со следующими параметрами:
   `date 09111330`
2. Скопируйте системное время для настройки аппаратных часов с помощью команды:
   `hwclock --systohc`
3. Проверьте, что системное и аппаратное время настроены корректно:
   `hwclock ; date`
4. Время должно быть одинаковым, допустимы небольшие отклонения.Остановите службу ntp:
   `systemctl stop ntp`
5. Определите сервер синхронизации времени. Вы можете использовать любую службу точного времени, работающую по протоколу ntp и доступную из вашей сети: как сетевое оборудование, так и контроллеры домена Windows (сервер Active Directory).
   Чтобы проверить, поддерживает ли сервер NTP, воспользуйтесь командой ntpdate -q <IP> (где IP - адрес проверяемого сервера), например:
   ```
   root@atl-iw:~# ntpdate -q 10.10.0.98
   server 10.10.0.98, stratum 3, offset 9.196765, delay 0.04437
   11 Sep 13:09:02 ntpdate[13819]: step time server 10.10.0.98 offset
   9.196765 sec
   root@atl-iw:~#
   ```
6. Настройте синхронизацию, указав сервер NTP-синхронизации в файле **/etc/ntp.conf**. Для этого добавьте запись вида (укажите IP-адрес вашего NTP-сервера):
   `server 10.10.0.98`
7. Запустите службу ntp:
   `systemctl start ntp`
8. Проверьте текущее состояние службы ntp с помощью команды:
   `systemctl status ntp`
9. Включите автоматическую синхронизацию времени:
   `chkconfig --level 345 ntp on`

4.3.2 Конфигурирование работы Sphinx при распределенной установке

При полнотекстовом поиске по запросам в Traffic Monitor используется механизм Sphinx. Служба **sphinx** устанавливается в режиме All-in-one или при установке шаблона Индексер**.** Сведения о режимах установки Системы приведены в документе «*InfoWatch Traffic Monitor. Руководство по установке*», статья "Схемы развертывания Системы и выбор типа установки".

Распределенной считается установка, при которой компоненты InfoWatch Traffic Monitor установлены на разных серверах.

> **Примечание:**
>
> При наличии нескольких серверов укажите IP-адрес сервера, на котором запущена служба **sphinx** (сервер с шаблоном Индексер). IP-адрес указывается в параметре `hostname` секции `search` конфигурационного файла **web.conf** на сервере с шаблоном Веб-консоль.

4.4 Настройка перехвата трафика

В Системе доступен перехват данных, передаваемых по протоколу SMTP.

SMTP (Simple Mail Transfer Protocol) - почтовый протокол, используемый почтовым клиентом для отправки исходящих сообщений электронной почты на сервер.

В ОС Astra Linux Special Edition "Смоленск" реализовано мандатное управление доступом.
Для этого используется мандатная метка, которая определяется:

1. мандатным уровнем конфиденциальности (от 0 до 255);
2. мандатным уровнем целостности;
3. мандатной категорией.

> **Примечание:**
>
> Более детальное описание мандатного управления доступом и контроля целостности см. в документации к операционной системе Astra Linux.

Traffic Monitor может учитывать мандатные метки при перехвате SMTP-трафика.

В зависимости от типа перехватываемого трафика и способа перехвата Система настраивается следующими способами:

- Настройка перехвата SMTP-трафика;
- Прием объектов, перехваченных InfoWatch Device Monitor.

> **Важно!**
>
> Предполагается, что Система уже установлена до начала настройки.
> Сведения об установке для каждой из схем развертывания приведены в документе "InfoWatch Traffic Monitor. Руководство по установке".

Важно!

После настройки перехвата трафика необходимо настроить параметры анализа объектов (см. документ "InfoWatch Traffic Monitor. Руководство пользователя").

Важно!

На Агентах Device Monitor, установленных на компьютерах по управлением ОС Astra Linux или РЕД ОС, перехват почтового трафика осуществляется только по каналам SMTP, POP3 и IMAP.

4.4.1 Настройка перехвата SMTP-трафика

Раздел содержит информацию о том, как настроить перехват SMTP-трафика:

- Прием копий с почтового сервера;
- Настройки почтовых серверов для перехвата SMTP-трафика.

Подробное описание схемы перехвата SMTP-трафика см. в статье "Схема перехвата SMTP-трафика".

Прием копий с почтового сервера

На корпоративном почтовом сервере требуется настроить правило, отправляющее скрытую копию (BCC) для каждого отправленного письма. Копия должна отравляться на несуществующий почтовый адрес почтового домена, IP-адрес которого соответствует серверу Traffic Monitor. Данная функция поддерживается большинством почтовых серверов.

Интеграция сервера Traffic Monitor с почтовым сервером Exim4 реализуется следующим образом. Почтовый сервер Exim4, встроенный в Систему Traffic Monitor, получает копии писем, отправленных по SMTP-протоколу. Копии писем поступают на **25**-й порт и передаются процессу **iw_smtpd** на порт **2025**. В Системе создается событие для каждого из писем, информация о которых затем помещается в базу данных.

Чтобы настроить прием копий с почтового сервера, выполните следующие действия:

1. Убедитесь, что включен автозапуск процесса **iw_smtpd** (см. "Автозапуск процессов")
2. Если необходимо, чтобы Traffic Monitor перехватывал письма с мандатным уровнем конфиденциальности выше 0, настройте работу SMTPD:
   a. Установите `exim4`, учитывающий мандатные метки. Для этого выполните команду:
   `apt-get install exim4-daemon-heavy`
   b. Убедитесь, что в конфигурационном файле **smtpd.conf**, расположенном в директории `/opt/iw/tm5/etc/` указано:
   `"EnablePrivSock": true,`
   c. Установите права пользователю **iwtm**, выполнив команду:
   `usercaps -m 0x100 iwtm`
   d. Выйдите из системы и пройдите повторную авторизацию.
   e. Выполните команду:
   `/etc/init.d/iwtm restart smtpd`
   f. Выполните команду:
   `/etc/init.d/exim4 restart`

3. На почтовом сервере настройте пересылку скрытых копий SMTP-сообщений (BCC - Blind Carbon Copy) на сервер Traffic Monitor. Настройка Exim4 для отправки скрытых копий приведена в статье "Настройка пересылки скрытых копий Exim4".
4. Настройте встроенный в Систему Exim4 (см. "Настройка сервера Exim4 в системе Traffic Monitor").
5. Убедитесь, что включен автозапуск для процессов, участвующих в перехвате и обработке почты:
exim4, iw_smtpd, iw_messed, iw_warpd,iw_luaengined, iw_cas, iw_pas, iw_x2x, iw_x2db, iw_tech-tools (см. "Автозапуск процессов").
6. Убедитесь, что в `/opt/iw/tm5/etc/luaengined.conf` в параметре `"TransportModes" / " Sm tp"` установлен режим `Copy` или пустое значение.

Все письма, полученные по правилу пересылки скрытой копии или журналирования, будут содержать постоянного получателя, например, `"iwtm-mon@local"`. Для его удаления настройте параметр `SmtpTmBccList` в файле **luaengined.conf**.

Важно!

Если в MS Exchange используется правило журналирования, для корректной обработки этих писем включите соответствующую опцию в Traffic Monitor. Для этого в конфигурационном файле `messed.conf` укажите:

```
"ExchangeJournalReports": true,
```

Чтобы опция стала активной, повторно загрузите конфигурацию или перезапустите службу.

Настройки почтовых серверов для перехвата SMTP-трафика

Раздел содержит информацию о настройке почтовых серверов для перехвата сервером Traffic Monitor:

- Настройка пересылки скрытых копий Exim4;
- Настройка сервера Exim4 в системе Traffic Monitor.

Настройка сервера Exim4 в системе Traffic Monitor.

Чтобы настроить встроенный в Систему сервер Exim4 для пересылки принятых им скрытых копии в Traffic Monitor, выполните следующие шаги:

1. Создайте файл `etc/exim4/exim4.conf`
2. Внесите в него следующую информацию и сохраните изменения:

```
primary_hostname = <Ваш хостнейм>
recipients_max=100
message_size_limit=50M
hostlist relay_from_hosts = MAIN_RELAY_NETS
MAIN_RELAY_NETS= 127.0.0.0/8 ; 10.0.0.0/8 ; 192.168.0.0/16 ; 172.16.0.0/12
daemon_smtp_ports = 25
#no policy checks
acl_smtp_rcpt = accept
```

```
#Routers: standard DNS routing and local users
begin routers
forward_route:
driver = manualroute
domains = *
self = send
transport = remote_smtp
route_list = * 127.0.0.1::2025
no_more
#Transports: SMTP and local mailboxes
begin transports
remote_smtp:
driver = smtp
command_timeout = 10s
```

3. Перезапустите Exim:
   `sudo service exim4 restart`
4. Чтобы корректно принимать почту с различными мандатными метками:
   a. Директории `db`, `input` и `msglog`, расположенные в `/var/spool/exim4`, должны иметь следующие атрибуты:
   `drwxrwxrwxm-- 2 Debian-exim Debian-exim Уровень_3:Низкий:Нет:ccnr,ehole db`
   `drwxrwxrwxm-- 2 Debian-exim Debian-exim Уровень_3:Низкий:Нет:ccnr,ehole input`
   `drwxrwxrwxm-- 2 Debian-exim Debian-exim Уровень_3:Низкий:Нет:ccnr,ehole msglog`
   Для этого выполните команду:
   `sudo /usr/sbin/pdpl-file 3:::ccnr,ehole <имя директории>`
   b. В директории `db` создайте файлы со следующими привилегиями (или измените атрибуты существующих файлов):
   `-rw-r-----m-- 1 Debian-exim Debian-exim Уровень_3:Low:Нет:ehole retry.lockfile`
   `-rw-r-----m-- 1 Debian-exim Debian-exim Уровень_3:Low:Нет:ehole wait-remote_smtp`
   `-rw-r-----m-- 1 Debian-exim Debian-exim Уровень_3:Low:Нет:ehole wait-remote_smtp.lockfile`
   Для этого выполните команду:
   `sudo /usr/sbin/pdpl-file 3:::ehole <имя файла>`

Настройка пересылки скрытых копий Exim4

Для настройки пересылки скрытых копий создайте дополнительные файлы конфигурации:

1. Создайте файл конфигурации **`/etc/exim4/conf.d/main/30_exim4-config_system_filter`** со следующим содержимым:
```
# System wide filter:
system_filter = /etc/exim4/conf.d/system.filter
system_filter_user = Debian-exim
system_filter_group = Debian-exim
system_filter_reply_transport = bcc_transport
# System wide filter end.
```

2. Создайте файл фильтра **`/etc/exim4/conf.d/system.filter`**. В нем укажите домен почтового сервера, с которого необходимо создавать скрытые копии, а также вспомогательный адрес e-mail с вспомогательным доменом:
```
# Exim filter
if first_delivery
 and ("$h_to:, $h_cc:, $h_bcc" contains "<почтовый_домен>")
 or ("$h_from:" contains "<почтовый_домен>")
then
 unseen deliver "bcc@<вспомогательный_домен>"
endif
```
3. Создайте маршрут для вспомогательного домена **`/etc/exim4/conf.d/router/120_exim4-config_iwtm_route`** со следующим содержимым:
```
iwtm_route:
driver = manualroute
domains = <вспомогательный_домен>
transport = bcc_transport
route_list = * <IP-адрес_сервера_Traffic_Monitor>::<порт>
```

> **Примечание:**
>
> Укажите значение порта **`25`** для отправки копий на сервер Exim4 в системе Traffic Monitor.
> Укажите значение порта **`2025`** для отправки копий в Traffic Monitor через SMTPD.

4. Создайте описание транспорта bcc_transport, используемого для доставки сообщений. Необходимо создать файл **`/etc/exim4/conf.d/transport/30_exim4-config_bcc_transport`** со следующим содержимым:
```
bcc_transport:
driver = smtp
command_timeout = 10s
```
5. Для обновления и перезапуска Exim4 выполните команды:
```
sudo update-exim4.conf
sudo /etc/init.d/exim4 restart
```
6. Чтобы корректно пересылать почту с различными мандатными метками:
   a. Директории `db`, `input` и `msglog`, расположенные в `/var/spool/exim4`, должны иметь следующие атрибуты:
   ```
   drwxrwxrwxm-- 2 Debian-exim Debian-exim
   Уровень_3:Низкий:Нет:ccnr,ehole db
   drwxrwxrwxm-- 2 Debian-exim Debian-exim
   Уровень_3:Низкий:Нет:ccnr,ehole input
   drwxrwxrwxm-- 2 Debian-exim Debian-exim
   Уровень_3:Низкий:Нет:ccnr,ehole msglog
   ```
   Для этого выполните команду:
   ```
   sudo /usr/sbin/pdpl-file 3:::ccnr,ehole <имя директории>
   ```
   b. В директории `db` создайте файлы со следующими привилегиями (или измените атрибуты существующих файлов):
   ```
   -rw-r-----m-- 1 Debian-exim Debian-exim Уровень_3:Низкий:Нет:ehole
   retry
   -rw-r-----m-- 1 Debian-exim Debian-exim Уровень_3:Низкий:Нет:ehole
   retry.lockfile
   ```

```
-rw-r-----m-- 1 Debian-exim Debian-exim Уровень_3:Низкий:Нет:ehole
wait-bcc_transport
-rw-r-----m-- 1 Debian-exim Debian-exim Уровень_3:Низкий:Нет:ehole
wait-bcc_transport.lockfile
```

Для этого выполните команду:

```
sudo /usr/sbin/pdpl-file 3:::ehole <имя файла>
```

4.4.2 Прием объектов, перехваченных InfoWatch Device Monitor

1. Установите и настройте серверную часть Traffic Monitor (см. документ "*InfoWatch Traffic Monitor. Руководство по установке*", статья "Установка InfoWatch Traffic Monitor").
2. Настройте передачу событий из Device Monitor на сервер Traffic Monitor (см. в базе знаний статью "Интеграция Device Monitor с различными версиями Traffic Monitor").
3. Настройте параметры анализа объектов (см. документ "*InfoWatch Traffic Monitor. Руководство пользователя*", статья "Технологии").

4.5 Автозапуск процессов

В этом разделе описаны особенности автозапуска процессов Системы, а именно:

- Проверка автозапуска процессов;
- Включение и выключение автозапуска процессов.

Общая информация о работе с процессами изложена в разделе "Процессы серверной части Traffic Monitor Server".

4.5.1 Проверка автозапуска процессов

> **Важно!**
>
> Отключите автозапуск нелицензированных процессов, отвечающих за перехват трафика. Это позволит уменьшить количество сообщений в журнале протоколирования.

При логическом выделении основного сервера, сервера перехвата и сервера с веб-консолью, убедитесь, что автозапуск процессов настроен в соответствии с указаниями в таблице.

Автозапуск должен быть включен только для процессов, которые необходимы данному экземпляру сервера Traffic Monitor для корректной работы:

Процесс	По умолчанию автозапуск включен
iw_adlibitum	Да **Примечание:** Процесс должен работать только на основном сервере
iw_agent	Да
iw_analysis	Да

Процесс	По умолчанию автозапуск включен
iw_kicker	Да **Примечание:** Процесс должен работать только на сервере с веб-консолью
iw_blackboard	Да
iw_bookworm	Да **Примечание:** Процесс должен работать только на основном сервере
iw_capstack	Нет
iw_cas	Да
iw_configerator	Да **Примечание:** Процесс должен работать только на сервере с веб-консолью
iw_deliver	Да **Примечание:** Процесс должен работать только на основном сервере
iw_icap	Да
iw_is	Да **Примечание:** Процесс должен работать только на основном сервере
iw_indexer	Да **Примечание:** Процесс должен работать только на основном сервере
iw_image_autoling	Да **Примечание:** Процесс должен работать только на сервере с веб-консолью
iw_text_autoling	Да **Примечание:** Процесс должен работать только на сервере с веб-консолью
iw_licensed	Да **Примечание:** Процесс должен работать только на основном сервере
iw_luaengined	Да

Процесс	По умолчанию автозапуск включен
iw_messed	Да
iw_proxy_http iw_proxy_icq iw_proxy_smtp	Нет
iw_qmover_client	Нет
iw_qmover_server	Нет
iw_sample_compiler	Да
consul	Да
iw_smtpd	Да
iw_sniffer	Нет
iw_system_check	Да
iw_tech_tools	Да
iw_updater	Да
iw_warpd	Да
iw_x2x	Да
iw_x2db	Да
iw_xapi_xapi iw_xapi_puppy	Да

Примечание:

Чтобы удалить с сервера все **iw_kicker** процессы, удалите пакет **iwtm-web-meta**:

```
rpm -e iwtm-web-meta
```

Подробнее о включении автозапуска процессов см. "Включение и выключение автозапуска процессов".

4.5.2 Включение и выключение автозапуска процессов

Чтобы включить/отключить автозапуск процесса (пример приведен для iw_proxy):

Служба **iw_proxy** состоит из трех компонентов: **iw_proxy_http**, **iw_proxy_icq**, **iw_proxy_smtp**. Поэтому:

1. Для возможности автозапуска процесса (например, **iw_proxy_http**) необходимо установить ему статус `enable`, выполнив команду:
   `iwtm enable iw_proxy_http`
   Чтобы отключить возможность автозапуска процесса, установите статус `disable:`
   `iwtm disable iw_proxy_http`
2. Перезапустите службу **iw_proxy**:
   `iwtm restart iw_proxy`

Все процессы Traffic Monitor со статусом `enable` подлежат автозапуску, а процессы со статусом `disable` могут быть запущены только вручную.

Чтобы настроить автозапуск процесса:

1. Директория хранения unit-файлов: **/usr/lib/systemd/system.** Внесите изменения в unit-файл нужного процесса, (см. "Описание конфигурационных и unit-файлов демонов Traffic Monitor") и сохраните его.
2. Выполните перезагрузку конфигурации:
   `systemctl daemon-reload`
3. Добавьте настроенный процесс в автозапуск:
   `iwtm enable <имя_демона>`
4. Запустите процесс:
   `iwtm start <имя_демона>`
5. Проверить, запустился ли сервис, можно любой из команд:
   `systemctl status <имя_демона>`
   `iwtm status`

Чтобы включить автозапуск сразу нескольких процессов, введите:

i `wtm enable <имя_демона1> <имя_демона2> ... <имя_демонаN>`

Чтобы выключить автозапуск сразу нескольких процессов, введите:

i `wtm disable <имя_демона1> <имя_демона2> ... <имя_демонаN>`

Чтобы включить автозапуск всех процессов в Системе, введите:

`iwtm enable all`

Чтобы выключить автозапуск всех процессов в Системе, введите:
`iwtm disable all`

4.6 Модуль взаимодействия с удаленной базой данных

Если схема развертывания Системы выбрана таким образом, что база данных, в которую передаются перехваченные объекты, находится на удаленном сервере, то необходимо установить модуль взаимодействия с удаленной базой данных.

Удаленной считается база данных, установленная на отдельностоящем сервере при помощи ключа установки **TME DB Server** или **TME All-in-one** (подробно о ключах установки см. "InfoWatch Traffic Monitor. Р уководство по установке", статья "Установка из дистрибутива TME").

Стандартным процессом передачи данных в базу является **iw_x2db** (см. "Список модулей Traffic Monitor Server"). Использование модуля взаимодействия с удаленной базой данных дает дополнительную возможность регулировать:

- скорость передачи файлов в файловую очередь
- расписание проходимости канала передачи файлов

Настройка модуля взаимодействия с отдельностоящей базой данных включает следующие задачи:

- Настройка клиентской части модуля взаимодействия с удаленной БД
- Настройка серверной части модуля взаимодействия с удаленной БД

4.6.1 Настройка сбора данных в филиальной сети

Если Система установлена в сети филиалов, то за отправку данных о перехваченных событиях и их получение на сервере Traffic Monitor отвечают службы:

- **iw_qmover_client** - пересылает данные из филиала;
- **iw_qmover_server** - принимает данные в головном отделении.

По умолчанию данные службы отключены и запускаются Офицером Безопасности вручную из Консоли управления Traffic Monitor:

- **iw_qmover_server** - в первую очередь;
- **iw_qmover_client** - во вторую очередь.

Их конфигурация настраивается в соответствующих конфигурационных файлах директории `/opt/iw/tm5/etc` : **qmover_client.conf** и **qmover_server.conf** (подробнее см. документ "Справочник по конфигурационным файлам", статьи "qmover_client.conf" и "qmover_server.conf").

4.6.2 Настройка клиентской части модуля взаимодействия с удаленной БД

Настройка общих параметров

1. Настройте следующие параметры клиента в конфигурационном файле **qmover_client.conf**, расположенном в директории `/opt/iw/tm5/etc` :

Параметр	Описание
`ListenerIP`	IP-адрес сервера (служба **qmover_server**)
`Port`	Порт сервера (центральный офис)
`NookDir`	Рабочий каталог службы **qmover_server**
`ChannelWidth`	Ширина полосы пропускания канала, скорость закачки данных в Traffic Monitor (Кбит/с). Может быть неявно ограничена параметром `WindowSize`

2. Перезапустите Traffic Monitor:
`iwtm restart`

Настройка автозапуска процессов

- Включите автозапуск для процесса **iw_qmover_client**.
- Отключите автозапуск для процессов **iw_x2db**, **iw_x2x**, **iw_deliverd** и **iw_adlibitum**.

Подробнее о включении и выключении автозапуска процессов см. "Включение и выключение автозапуска процессов".

Изменение ширины полосы пропускания для канала передачи данных

По умолчанию полоса пропускания имеет ширину 256 Кбит/с. Но Вы можете настроить автоматическое изменение полосы пропускания в различные периоды времени. Для этого используется утилита **iw_qmover_channel_width_setter**.
Допускается изменение ширины полосы пропускания. Минимальная ширина – 10 Кбит/с.
Максимальная ширина не установлена, но рекомендуемое максимальное значение – 2 Мбит/с.

Чтобы настроить ограничения на ширину полосы пропускания,
Запустите утилиту с обязательными параметрами:
`iw_qmover_channel_width_setter <unix_socket_name> <полоса_пропускания>`
где

- `unix_socket_name` – имя сокета (автоматически создается при запуске на время выполнения службы; при завершении службы удаляется автоматически); значение по умолчанию - `/opt/iw/tm5/run/.channel_socket` ;
- `полоса_пропускания` – ширина полосы пропускания, в кбит/с, которую нужно установить.

Пример
Имеется канал с полосой пропускания 256 кбит/с. Необходимо, чтобы в период с 9.00 до 18.00 канал был занят на 50%. А с 18.00 до 9.00 на 100%.
Рассчитайте ширину полосы пропускания, исходя из загрузки вашего канала. В этом примере ширина полосы пропускания для разных интервалов времени составляет:

Интервал	**Ширина полосы пропускания**	
	Процент от величины канала	В пересчете на кбит/с
9.00 до 18.00	50%	128
18.00 до 9.00	100%	256

Добавьте в /**etc/crontab** команды:
```
00 9 * * * iwtm /opt/iw/tm5/bin/iw_qmover_channel_width_setter /opt/iw/tm5/run/.channel_socket 128
 00 18 * * * iwtm /opt/iw/tm5/bin/iw_qmover_channel_width_setter /opt/iw/tm5/run/.channel_socket 256
```

4.6.3 Настройка серверной части модуля взаимодействия с удаленной БД

Настройка конфигурации

Настройте следующие параметры клиента в конфигурационном файле **/opt/iw/tm5/etc/qmover_server.conf**. Директория, в которой хранится очередь объектов – `/opt/iw/tm5/queue/db` .
Для этого:

1. Укажите IP-адрес обслуживаемого агента (филиал) в параметре `IP`.
2. Укажите порт, на котором сервер прослушивает объекты, поступающие от агентов в параметре `Port`.

> **Важно!**
>
> При изменении параметров филиалов (изменение количества филиалов или их IP-адресов) следует внести изменения в файл **qmover_server.conf**.

Настройка автозапуска

Включите автозапуск для процесса **iw_qmover_server** (подробнее о включении и выключении автозапуска процессов см. "Включение и выключение автозапуска процессов".

4.7 Настройка OCR-экстракторов

Настройка OCR для различных перехватчиков производится в следующих конфигурационных файлах:

- **warpd.conf** – чтобы включить OCR для анализа перехваченных изображений, в секции `Warp` укажите параметру `EnableOCR` значение `true`;
- **sample_compiler.conf** – чтобы включить OCR для анализа изображений, загружаемых в качестве эталонных документов, укажите параметру `EnableOCR` значение `true`.

В файле **/opt/iw/tm5/etc/config-perm/bookworm/ocr_custom.xml** вы можете настроить OCR для каналов перехвата:

- на уровне сервиса.
- на уровне типа события для конкретного сервиса.
- на уровне протокола для конкретного типа события.

Подробнее о настройках см. "Настройка использования OCR"

> **Примечание.**
>
> В случае распределенной установки Traffic Monitor на несколько серверов вы должны задать настройки включения OCR для каждого сервера отдельно.

> **Важно!**
>
> Настройка на уровне протокола имеет более высокий приоритет, чем настройка на уровне типа события. Настройка на уровне типа события имеет более высокий приоритет, чем настройка на уровне сервиса.

> **Примечание.**

Если с помощью SDK был зарегистрирован новый тип событий, то к нему применяются настройки для сервиса, к которому относится данный тип события. Если зарегистрирован новый протокол, для него действуют настройки типа события, к которому относится данный протокол.

После того, как вы внесли изменения в файлах, выполните команду:

```
iwtm restart
```

В таблице ниже перечислены каналы перехвата, для анализа событий из которых в Traffic Monitor может использоваться OCR.

Сервис	Тип события	Протокол
Почта	Email	POP3 SMTP IMAP
	Web-почта	HTTP HTTPS
Интернет-активность	Web-сообщение	HTTP HTTPS
Хранение	Data Discovery	-

При установке с помощью инсталлятора на серверную часть Системы устанавливаются оба OCR-экстрактора. По умолчанию на работу настроен ABBYY FineReader Engine 11. Распознавание текста из извлеченных изображений производится с использованием одного из двух OCR-экстракторов: ABBYY FineReader Engine 11 и Tesseract 4.1.1.

Функциональные ограничения экстракторов приведены в таблице ниже:

Функциональность	ABBYY FineReader Engine 11	Tesseract 4.1.1
Распознавание углов поворота изображения	0(+/-20) , 90(+/-20), 180(+/-20) и 270(+ /-20) градусов	-
Распознавание цветного изображения	+	-
Коррекция изображения	+	-
Рекомендуемое разрешение изображения	300dpi для текста с размером шрифта от 10pt 400-600dpi для текста с размером шрифта 9pt и меньше	300dpi для текста с размером шрифта от 10pt 400-600dpi для текста с размером шрифта от 9pt и меньше

Примечание:

Чтобы избежать больших отклонений от рекомендуемого разрешения изображения, при работе экстрактора ABBYY FineReader Engine 11 используется параметр `AutoOverwriteResolution` (со значением `true` по умолчанию), позволяющий автоматически определять разрешение изображения.

Чтобы настроить на работу экстрактор ABBYY FineReader Engine 11:

1. Удалите символьную ссылку **/opt/iw/tm5/bin/iw_image2text**.
2. Создайте символьную ссылку:
```
ln -s /opt/iw/tm5/bin/iw_image2text_fre /opt/iw/tm5/bin/iw_image2text
```

Чтобы заменить используемый экстрактор ABBYY FineReader Engine 11 на Tesseract 4.1.1:

1. Удалите символьную ссылку **/opt/iw/tm5/bin/iw_image2text**.
2. Создайте символьную ссылку:
```
ln -s /opt/iw/tm5/bin/iw_image2text_ts /opt/iw/tm5/bin/iw_image2text
```

Чтобы изменить ограничение на размер пересылаемого изображения:

1. Перейдите в директорию **/opt/iw/tm5/etc/image2text_fre.conf** (для FineReader Engine 11) или /**opt/iw/tm5/etc/image2text_ts.conf** (для Tesseract).
2. Отредактируйте параметры `MaxSizeInKb` (верхняя граница) и `MinSizeInKb` (нижняя граница). По умолчанию установлено 1536 КБ и 200 КБ соответственно.

Чтобы включить OCR только для событий облачного хранилища:

1. Откройте справочник `/opt/iw/tm5/etc/config/bookworm/services.xml` и найдите соответсвующие для облачного хранилища `mnemo` и `key`.
   Например,
```
object_type mnemo="cloud_storage"
key="AA9DFB259F0DFEE040BADC95815E13A200000000"
```
2. Скопируйте данные key и `mnemo` и вставьте в `/opt/iw/tm5/etc/config-perm/bookworm/ocr_custom.xml`

Если в качестве OCR-экстрактора используется ABBYY FineReader Engine 11, необходимо настроить лицензию ABBYY. Для этого:

1. Введите полученные серийный номер и пароль в конфигурационный файл **image2text_fre.conf**, расположенный в директории **/opt/iw/tm5/etc**:
   - в поле **SerNum** введите серийный номер;
   - в поле **Pwd** введите пароль.
2. Скопируйте полученный файл с лицензией (формат .LocalLicense) в директорию /**var/lib/ABBYY/SDK/11/Licenses**.
3. Убедитесь, что у пользователя **iwtm** есть права на доступ к скачанному файлу.
4. Перезапустите сервис **iw_warpd**.

Экстрактор ABBYY FineReader Engine 11 может работать в двух режимах:

1. сбалансированный – предусматривает максимально качественную обработку изображений. Этот режим включен по умолчанию. Настройки его пользовательского профиля находятся в файле **etc/FRProfile.ini**;
2. быстрый – позволяет максимально быстро обработать изображения, при синтезе документа одновременно загружается 64 страницы. При необходимости этот режим можно настроить вручную.
   В быстром режиме включены:
   - быстрый режим определения ориентации страниц (качество снижено);
   - стандартный анализ страницы, без подбора шаблонов;
   - игнорирование теней;
   - обнаружение таблиц во время анализа;
   - быстрое извлечение объектов (качество снижено).

   В быстром режиме выключены:
   - балансировочный режим;
   - обработка изображений, как фото;
   - сжатие изображений;
   - использование бинаризации изображений;
   - детектирование параметров шрифта;
   - удаление мусора с изображения;
   - удаление шумов с временного изображения;
   - детектирование "пористого" текста;
   - распознавание всего текста на изображении, включая текст в самих изображениях;
   - извлечение из изображения всего возможного текста;
   - детектирование заголовков;
   - детектирование сносок;
   - детектирование оглавления;
   - детектирование колонтитулов;
   - детектирование страниц документа.

Чтобы активировать быстрый режим обработки документов, в конфигурационном файле **image2text_fre.conf**, расположенном в директории **/opt/iw/tm5/etc**, укажите путь до файла с настройками пользовательского профиля в параметре `"ABBYYProfile":"etc/FRProfile_fast.ini"` и сохраните изменения.

> **Важно!**
>
> Оптимизация скорости обработки документов недоступна для скан-копий схем, карт местности и монохромных изображений.

4.8 Настройка отправки уведомлений пользователям и сотрудникам

Пользователь Консоли управления имеет возможность настроить отправку уведомлений из Системы пользователю или сотруднику. Чтобы Система имела возможность отправлять уведомления, требуется:

- указать Системе электронный адрес, с которого будут отправляться уведомления – электронные сообщения пользователям или сотрудникам

- настроить отправку писем через почтовый сервер Exim.

Уведомления отправляются в результате срабатывания тех или иных правил в политиках (подробнее см. документ "InfoWatch Traffic Monitor. Руководство пользователя").

Чтобы проверить, отправляются ли письма через почтовый сервер,
Выполните команду:
`sendmail -v <employee@company.com> < <sample.log>`
где:

- `<employee@company.com>` - email-адрес пользователя, которому будет отправлено уведомление
- `<sample.log>` - текстовый файл, содержимое которого будет служить текстом письма (для проверки рекомендуется использовать простой текстовый файл размером до 1 МБ).

4.9 Ограничение количества найденных событий

Вы можете указать ограничение для количества событий, которые будут выведены в Консоли управления в результате применения фильтра. Для этого в таблице *SETTING* Базы данных укажите требуемое значение для атрибута *query_stop_count*. По умолчанию задано ограничение 10 000.

4.10 Настройка межсервисного взаимодействия (служба Consul)

Для регистрации сервисов, мониторинга доступности и обнаружения компонентов Traffic Monitor используется децентрализованный отказоустойчивый discovery-сервис Consul (Консул). Агент Консула:

- устанавливается на каждый хост и является полноправным участником кластера
- обнаруживает сервисы, собирает данные об их состоянии, реализуют интерфейсы DNS, API HTTP и RPC CLI.
- может быть запущен в одном из двух режимов: клиентском или серверном.

Consul используется во всех типах установки. Сведения о режимах установки Системы приведены в Руководстве по установке, статья "Схемы развертывания Системы и выбор типа установки").

При схеме установки Системы All-in-one конфигурирование параметров подключения к службе Консул осуществляется автоматически при установке. Исключение составляет распределенная установка, когда компоненты Traffic Monitor и База данных установлены на разных серверах (Система установлена на более чем одну ноду и имеет более одного сетевого интерфейса). Информацию по настройке см. в статье Конфигурирование Consul и создание кластера.

Настройка работы службы осуществляется в конфигурационном файле по пути `/opt/iw/tm5/etc/consul/consul.json`. Полное описание параметров можно посмотреть на странице "Конфигурационный файл consul.json и unit-файл iwtm-consul.service".

4.10.1 Запуск и остановка службы

Для запуска службы Консул используются любая из команд:

```
service iwtm-consul start
systemctl start iwtm-consul.service
```

Для остановки службы Консул используются команды:

```
service iwtm-consul stop
systemctl stop iwtm-consul.service
```

Ручной запуск агента Консул при необходимости может быть осуществлен следующим способом `:`

```
consul agent -data-dir=<path> -bind=<bind_addr> -bootstrap -server -ui
```

где:

- `<path>` - путь до директории со служебными данными (например: `/opt/iw/tm5/var/consul`),
- `<bind>` - адрес сетевого интерфейса.

Если в Системе больше одного сетевого интерфейса, то нужно указать один параметр (на выбор):

- `-bind=<ip-адрес>`.
- `-config-file` или `-config-dir -` путь к конфигурационному файлу (например: `/opt/iw/tm5/etc/consul`).

При загрузке нескольких конфигурационных файлов их опции будут объединены.

4.10.2 Регистрация сервисов в Consul

Сервис можно зарегистрировать в Consul двумя способами:

- использовать HTTP API или конфигурационный файл агента, в случае если сервис может общаться с Consul самостоятельно;
- зарегистрировать сервис как внешний компонент.

Сервисы с обязательной регистрацией в Consul	Сервисы, установленные на нодах с присутствием Consul Server или Consul Client
iw_adlibitum	iw_adlibitum
iw_agent	iw_agent
iw_analysis	iw_analysis
iw_blackboard	iw_bookworm
iw_bookworm	iw_capstack
iw_capstack	iw_cas
iw_cas	iw_configerator
iw_icap	iw_icap
iw_deliver	iw_indexer
iw_indexer	iw_licensed
iw_licensed	iw_kicker
iw_luaengined	iw_luaengined
iw_messed	iw_messed
iw_pas	iw_pas
iw_proxy_http	iw_proxy_http
iw_proxy_icq	iw_proxy_icq
iw_proxy_smtp	iw_proxy_smtp
	iw_qmover_server

Сервисы с обязательной регистрацией в Consul	Сервисы, установленные на нодах с присутствием Consul Server или Consul Client
iw_qmover_server	iw_sample_compiler
iw_sample_compiler	iw_system_check
iw_system_check	iw_smpd
iw_smtpd	iw_tech_tools
iw_tech_tools	iw_updater
iw_updater	iw_warpd
iw_warpd	iw_xapi_xapi
iw_xapi_xapi	iw_xapi_puppy
iw_xapi_puppy	iw_x2x
iw_x2x	iw_x2db
iw_x2db	Web GUI

После регистрации взаимодействие между сервисами Traffic Monitor и Consul осуществляется по сценарию:

- регистрация при помощи клиента Consul;
- установление TCP-соединения или возврат HTTP-кода;
- дерегистрация.

Клиент Consul получает от сервера список доступных для подключения компонентов Traffic Monitor.

4.10.3 Распределенная установка

Важно!

Для использования Consul необходима реализация full-mesh топологии сети (соединение "каждый с каждым") всех агентов внутри кластера, а также серверов при объединении их в WAN.

Важно!

В случае распределенной установки ТМ не должно быть запрета использования TCP и UDP между разными сегментами сети (в том числе нодами) в брендмауэре Windows.

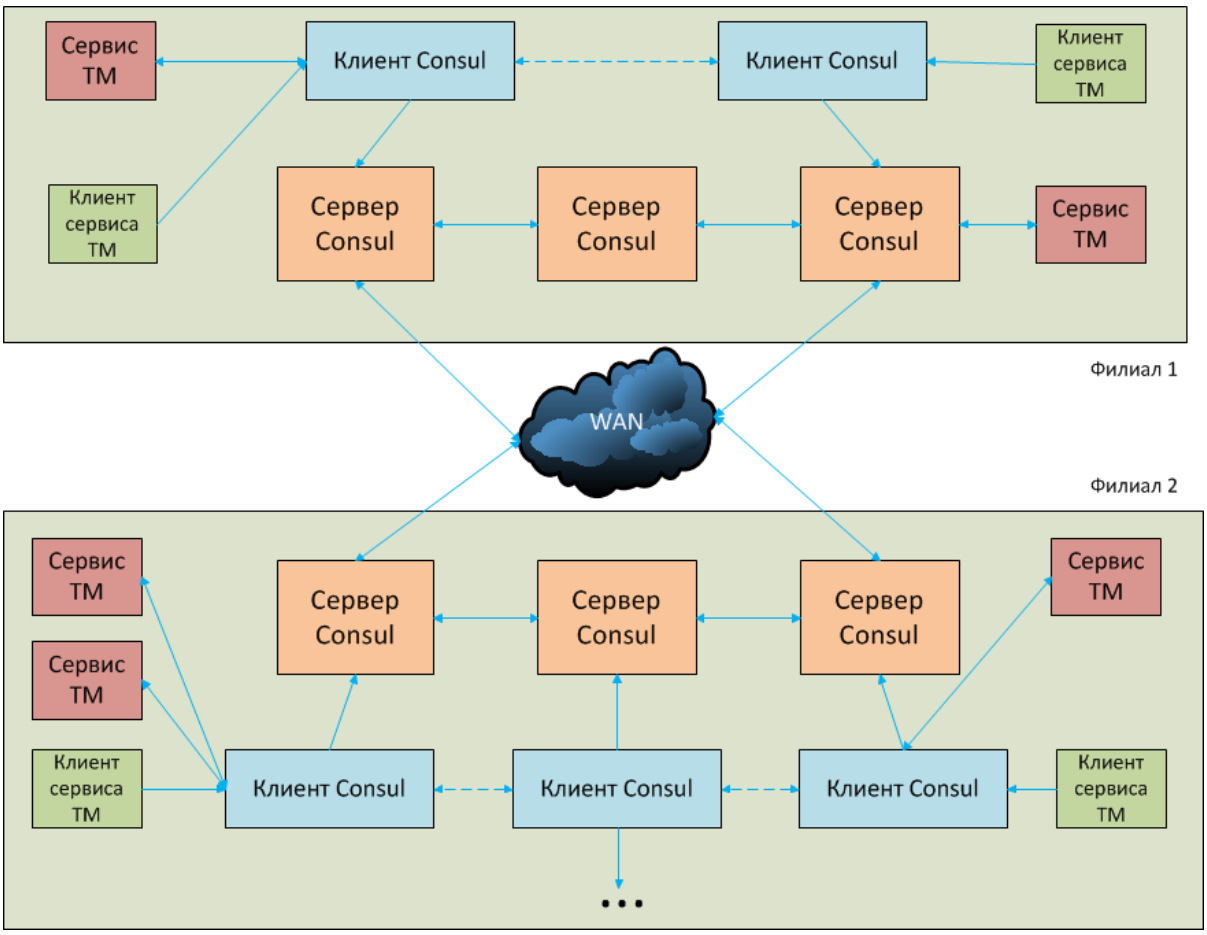
Кластеризация

Агент собирает данные об узле и сервисе и отсылает их серверу. Компоненты системы в поиске сервисов обращаются с запросом к агенту, запущенному на локальной машине, который пересылает его доступному серверу. Если сервер не в состоянии ответить на запрос, он может направить его в другой дата-центр (филиал) и вернуть полученный ответ.

Серверы образуют кластер и самостоятельно назначают сервера-Лидера, отвечающего за координацию элементов в кластере. Первый/единственный сервер обычно запускается в bootstrap-режиме (назначается лидером вручную). При старте агента для присоединения к кластеру достаточно

указать один сервер этого кластера. При его конфигурировании в опции `retry_join` рекомендуется указывать все серверы кластера. Кворум для проведения операций и обеспечения согласованности требуется в каждом дата-центре. При наличии нескольких дата-центров в каждом создается отдельный кластер. Дата-центры не изолированы друг от друга в рамках задачи обнаружения сервисов. Агент в одном дата-центре может получить информацию из другого дата-центра.

Сеть Consul может использовать один сервер, но рекомендуется, чтобы избежать потери данных, использовать от трех до семи серверов в дата-центре.



Пример:

Кластер из трех узлов (серверов) сохраняет свою работоспособность при выходе из строя одного сервера.

Примеры использования интерфейса командной строки:

Команда	Описание
`consul members`	вывод списка членов кластера
`consul catalog services -node <имя_ноды>`	вывод списка сервисов кластера на конкретной машине

Команда	Описание
`consul operator raft list-peers -stale=true`	вывод списка всех серверов даже с случае развала кластера

Присоединение к кластеру

Регистрация дополнительных нод на кластере возможна любым из способов:

- Ввести команду: `consul join <ip-адрес>`. При этом нужно убедиться, что у всех серверов и клиентов Consul совпадают параметры `datacenter` и `encrypt` в конфигурационном файле consul.json (см. Конфигурационный файл consul.json и unit-файл iwtm-consul.service)
- Указать список всех серверов, к которым надо присоединиться в параметре `retry_join` в конфигурационном файле `consul.json`

4.10.4 Настройка сетевых правил доступа в Consul

Для установки сетевого обмена необходимо, чтобы следующие порты между всеми агентами Консул (серверами и клиентами) были открыты.

Порт по умолчанию	Интерфейс	Поддерживаемые протоколы	Описание
8300	Server RPC	TCP	Используется серверами для обработки входящих запросов от других агентов
8301	Serf LAN	TCP, UDP	Используется агентами для обработки потоков данных в локальной сети
8302	Serf WAN	TCP, UDP	Используется серверами для обмена данными по WAN с другими серверами
8400	CLI RPC	TCP	Используется всеми агентами для обработки RPC из CLI
8500	HTTP API	TCP	Используется клиентами для взаимодействия с интерфейсом HTTP API
8600	DNS	TCP, UDP	Используется для разрешения DNS-запросов

4.10.5 Конфигурационный файл consul.json и unit-файл iwtm-consul.service

Содержимое	Описание
{ 	
"bootstrap_expect": 1, 	Количество ожидаемых серверов в кластере (только для режима сервера)
"server": true, 	Запуск агента в режиме сервера (false – клиента)
"datacenter": "iwtm", 	Название дата-центра
"data_dir": "/opt/iw/tm5/var/consul", 	Директория для служебных данных Consul
"encrypt": "4RTZ5ttYY6RwIYX28XWNPw==", 	Секретный ключ, разделяемый между агентами кластера
"log_level": "WARN", 	Уровень логирования
"enable_syslog": true, 	Разрешить логирование в syslog
"disable_update_check": true, 	Запрет проверки на наличие обновлений Consul
"leave_on_terminate": false, 	Если true, то при получении SIGTERM, рассылает прощальное сообщение и покидает кластер в штатном порядке. По умолчанию: для сервера – false, для клиента – true
"skip_leave_on_interrupt": true, 	Если false, то при получении сигнала прерывания (например, SIGINT) покидает кластер в штатном порядке. По умолчанию: для сервера – true, для клиента – false
"rejoin_after_leave": true, 	Если true, то присоединяется к кластеру при старте

Содержимое	Описание
`"retry_join": "0.0.0.0",`	В случае распределенной установки содержит список серверов, к которым надо присоединиться
`"ui": false,`	Доступ к веб-интерфейсу (по умолчанию запрещен)
`"client_addr": "0.0.0.0"`	IP клиента, к которому открыт доступ по веб-интерфейсу
`}`	

Unit-файл **iwtm-consul.service**

Код	Описание
`[Unit]`	Раздел для определения метаданных демона и настройки его взаимодействия с другими демонами
`Description=Consul Service discovery agent`	Название демона
`Requires=network-online.target`	Сервисы не могут работать без network-online.target
`After=network-online.service`	Демоны, которые будут запущены до запуска текущего демона
`[Service]`	Раздел настройки запуска демона
`StandartOutput=syslog`	Перенаправление стандартного вывода в syslog
`StandartError=syslog`	Перенаправление стандартного вывода ошибок в syslog
`Type=simple`	Тип запуска демона. `simple` - демон запускается и переходит в режим ожидания на консоли
`User=iwtm`	Имя пользователя, от которого осуществляется запуск демона
`Group=iwtm`	Имя группы пользователя

Код	Описание
`SyslogIdentifier=iwtm-consul`	Устанавливает имя процесса для префиксных строк журнала, отправленных в систему ведения журнала или в буфер журнала ядра.
`PermissionsStartOnly=true`	Запуск команд из ExecStartPre под пользователем **root**
`ExecStartPre=/opt/iw/tm5/bin/consul validate /opt/iw/tm5/etc/consul`	Полный путь и аргументы команды, которые должны быть выполнены до запуска основного процесса
`ExecStart=/opt/iw/tm5/bin/consul agent $OPTIONS -config -dir= /opt/iw/tm5/etc/consul`	Полный путь и аргументы команды, которая должна быть выполнена для запуска демона. Запускается от пользователя, указанного в `User`
`ExecReload=/bin/kill -s SIGHUP $MAINPID`	Команда для перезагрузки конфигурации демона, если она доступна. Запускается от пользователя, указанного в `User`
`KillSignal=SIGINT`	Команда, по которой можно принудительно завершить работу демона
`TimeoutStopSec=5`	Время, в течение которого система будет ждать остановки демона, прежде чем пометить его недоступный или завершенный принудительно
`[Install]`	Определение поведения демона, если он включен или отключен
`WantedBy=multi-user.target`	Запускать этот демон, когда система грузится в multi-user режиме

4.11 Рекомендации по настройке безопасного окружения

Компоненты Traffic Monitor используют различные сетевые порты на серверах Системы.

> **Примечание:**
>
> Подробнее о наборах компонентов и шаблонах на серверах Системы см. "InfoWatch Traffic Monitor. Руководство по установке", "Схемы развертывания Системы и выбор типа установки".
>
> О процессах на серверах Traffic Monitor см. "Процессы серверной части Traffic Monitor Server".

Чтобы предотвратить несанкционированный доступ к компонентам, рекомендуется настроить сетевой экран. Должны быть разрешены только те соединения, которые необходимы для работы компонентов.

4.11.1 Порты компонентов

Компонент	Порты	Описание	Рекомендация
consul	TCP/UDP 8301	Используется для регистрации, мониторинга и обнаружения сервисов Traffic Monitor	Настройте сетевой экран для каждого сервера Traffic Monitor. Внешние системы не должны отправлять трафик на порт компонента
nginx	TCP 443, 80	Используется для: • Веб-консоли Traffic Monitor. • Интеграций с продуктами Платформы, которые получают данные из Traffic Monitor. • Взаимодействия с адаптерами, которые используют: DataExport API, API для Выгрузок из БД, API для данных аудита.	Настройте сетевой экран для сервера с шаблоном Веб-консоль. Отправлять трафик на порты компонента должны только: • Офицеры безопасности и администраторы Системы. • Продукты Платформы. • Адаптеры.
iw_smtpd	TCP 2025	Компонент получает трафик от почтового сервера Postfix/ Exim	Настройте сетевой экран для каждого сервера с шаблоном Перехватчики. Только почтовый сервер должен отправлять трафик на порт компонента
iw_xapi	TCP 9100, 9101	Используется для: • Получения данных от Device Monitor. • Получения данных от Data Discovery. • Взаимодействия с адаптерами, которые отправляют данные в Traffic Monitor с помощью pushAPI.	Настройте сетевой экран для каждого сервера с шаблоном Перехватчики. Отправлять трафик на порты компонента должны только: • Device Monitor. • Data Discovery. • Адаптеры.

4.12 Настройка сервера лицензирования (в случае кластера)

Для корректного получения данных от перехватчиков в полном объеме необходимо настроить сервер лицензирования. Он должен быть включен на основном сервере Traffic Monitor и выключен на остальных нодах кластера.
Для этого на ноде:

1. Остановите службу **iw_licensed**:
   `iwtm stop licensed`
2. Выключите автозапуск службы **iw_licensed**:
   `iwtm disable iw_licensed`

На основном сервере Traffic Monitor:

1. Включите автозапуск службы **iw_licensed:**
   `iwtm enable iw_licensed`
2. Запустите службу **iw_licensed**:
   `iwtm start licensed`

Для установки, удаления, проверки валидности или запроса лицензии ознакомьтесь с разделом "Управление лицензиями" Руководства пользователя Traffic Monitor.

5 Мониторинг

Подсистема мониторинга выполняет следующие функции:

- мониторинг работы всех серверов, входящих в состав решения InfoWatch Traffic Monitor (как физических, так и виртуальных);
- мониторинг всех программных компонентов, входящих в состав решения (ОС, СУБД, БД, процессы и т.д.);
- контроль значений индикаторов для каждого сервера и компонента. Под контролем подразумевается периодическое получение значения индикатора и сравнение значения индикатора с пороговым значением;
- возможность включения и отключения мониторинга отдельных индикаторов и отдельных серверов;
- отправка на почту уведомлений о выходе значений индикаторов из нормальных пределов.

Подсистема мониторинга автоматически устанавливается при установке серверных компонентов системы InfoWatch Traffic Monitor с помощью инсталлятора. О порядке установки см. документ «InfoWatch Traffic Monitor. Руководство по установке».

Если Система развернута на нескольких серверах с использованием инсталлятора и шаблонов установки, диагностика серверов будет настроена автоматически, а состав датчиков будет соответствовать шаблонам.

Также доступна Ручная настройка индикаторов.

5.1 Настройки подсистемы мониторинга

> **Важно!**
>
> Если Система развернута на нескольких серверах с использованием инсталлятора и шаблонов установки, диагностика серверов будет настроена автоматически, а состав датчиков будет соответствовать шаблонам.

Настройка подсистемы мониторинга включает следующие задачи:

- Настройка подключения Device Monitor;
- Ручная настройка индикаторов;
- Настройка адреса сервера синхронизации времени для подсистемы мониторинга;

5.1.1 Настройка подключения Device Monitor

Чтобы настроить мониторинг для сервера Device Monitor:

1. Отредактируйте файл `/opt/iw/tm5/etc/nagios/iwmon/iwmon-hosts-dm.cfg`:

- Раскомментируйте секции `host` и `hostgroup`.
- В параметре `address` секции `host` определите имя хоста или IP-адрес сервера Device Monitor.
- В файле / `opt/iw/tm5` `/etc/nagios/iwmon/iwmon-services-dm.cfg` раскомментируйте секцию `service`.

- В файле `/opt/iw/tm5/etc/nagios/iwmon/iwmon-commands.cfg` раскомментируйте секцию `Check Dm server`.
- Перезапустите процесс nagios:
  `systemctl restart iwtm-nagios`

5.1.2 Ручная настройка индикаторов

Включение и выключение индикаторов производится в конфигурационных файлах директории `/opt/iw/tm5/etc/nagios/iwmon`:

- iwmon-services-db-psql.cfg – настройка индикаторов для базы данных Postgre SQL;
- iwmon-services-dm.cfg – настройка индикаторов для Device Monitor;
- iwmon-services-loadavg.cfg – настройка параметров нагрузки на серверы (индикатор Общая нагрузка системы);
- iwmon-services.cfg – настройка основных индикаторов;
- iwmon-services-queue.cfg – настройка индикаторов очередей.

Параметр `service_description` отображает назначение каждого индикатора.

Чтобы включить индикатор, измените значение параметра `register на 1.`

Чтобы выключить индикатор, измените значение параметра `register` на `0`.

Чтобы применить изменения, перезапустите процесс nagios:

`systemctl restart iwtm-nagios`

5.1.3 Настройка адреса сервера синхронизации времени для подсистемы мониторинга

Для корректной синхронизации времени, на всех серверах системы необходимо указать IP-адрес NTP-сервера. Вы можете использовать любую службу точного времени, работающую по протоколу NTP и доступную из вашей сети: как сетевое оборудование, так и контроллеры домена Windows.
Чтобы настроить синхронизацию времени на сервере:

1. Откройте на редактирование файл `/opt/iw/tm5/etc/nagios/iwmon/iwmon-services-ntp.cfg`.
2. В значении параметра `check command` замените IP-адрес, заданный по умолчанию, на актуальный IP-адрес NTP-сервера (сервера синхронизации времени).

6 Администрирование базы данных

Этот раздел содержит информацию по администрированию:

- PostgreSQL.

Информацию по сбору статистики БД можно посмотреть в статье базы знаний "Сбор статистики БД".

6.1 PostgreSQL

Чтобы подключиться к серверу БД из терминала сервера Traffic Monitor, используйте следующие команды:

```
su - iwtm
psql -p 5433 -h <IP-адрес_сервера_БД> postgres iwtm
psql -p 5433 -h 127.0.0.1 postgres iwtm
```
- для подключения к локальной БД

Для подключения к БД с рабочих станций под управлением Windows, используйте программу pgAdmin.

Информация, необходимая для подключения содержится в конфигурационном файле **/opt/iw/tm5/csw/postgres/database.conf**.

Раздел содержит инструкции по администрированию PostgreSQL:

- Изменение предустановленных паролей;
- Табличные пространства в базе данных InfoWatch Traffic Monitor;
- Управление ежедневными табличными пространствами;
- Резервное копирование базы данных;
- Проведение регламентных работ на сервере базы данных;
- Логирование базы данных.

6.1.1 Изменение предустановленных паролей

Чтобы заменить предустановленные пароли пользователей БД:

1. Остановите процессы Traffic Monitor:
   `iwtm stop`
2. Подключитесь к БД PostgreSQL:
   `psql -p 5433 -h 127.0.0.1 postgres iwtm`
3. Получите список пользователей:
   `\du`
4. Измените пароли пользователей:
   `alter user 'iw_user' with password 'new_password';`
   где `'iw_user'` - выбранный пользователь БД, `'new_password'` - новый пароль для этого пользователя.
5. Выйдите из БД PostgreSQL:
   `\q`
6. Запустите процессы Traffic Monitor:
   `iwtm start`

Чтобы изменить предустановленный пароль Traffic Monitor, смотрите "Изменение предустановленного пароля Traffic Monitor".

6.1.2 Табличные пространства в базе данных InfoWatch Traffic Monitor

В базе данных InfoWatch Traffic Monitor Enterprise имеются два типа табличных пространств (ТП):

Тип табличного пространства	Назначение
Основное	Хранение настроек, которые нужны для анализа и обработки объектов (конфигурация, теги, цвета и пр.). Управление системой через Консоль управления (роли, учетные записи пользователей и др.)
Ежедневное	Хранение объектов, перехваченных в течение одних суток. Хранение информации о результатах анализа и обработки объектов (разобранный объект, категории, термины и др.). Состоит из трех табличных пространств: для хранения объектов со статусами *Нарушение*, *Нет нарушений, Остальное (пространство для хранения снимков экрана)*. Таким образом, достигается возможность раздельного архивирования, восстановления и удаления.

Все данные об объектах, перехваченных в определенный день, находятся в одном ежедневном ТП. Ежедневные ТП создаются каждые сутки с таким расчетом, чтобы в базе данных всегда были ТП для работы в ближайшие шесть суток, не включая текущие. Всем ежедневным ТП автоматически присваивается имя *IWTM_X,* где *X* – номер ТП (tbs_id из таблицы tbs_list).

Параметры, предназначенные для управления сегментами данных, задаются при настройке схемы базы данных, но могут быть переопределены после создания схемы (см. "Управление ежедневными табличными пространствами").
При использовании типа установки **TM Standard**, события хранятся в едином табличном пространстве. Автоматический расчет свободного места для будущих событий с освобождением пространства происходит еженедельно.

6.1.3 Управление ежедневными табличными пространствами

В этом разделе:

- Настройка размещения файлов на файловой системе;
- Настройка режимов хранения файлов табличного пространства;
- Архивирование ежедневных табличных пространств;
- Восстановление ежедневных табличных пространств;
- Удаление ежедневных табличных пространств.

Архивирование ежедневных табличных пространств

Чтобы освободить пространство на жестком диске, Вы можете периодически архивировать устаревшие данные. Архив с данными рекомендуется размещать на внешних носителях информации. При необходимости эти данные могут быть восстановлены.

Важно!

Если в качестве интервала времени используются месяцы или годы, то рекомендуется в расчете использовать максимальное значение интервала. Для месяца – 31 день. Для года – 366 дней. Например, чтобы архивировать ежедневные ТП старше 4-х лет, в задании iwtm_iwtm_archive_tablespaces укажите интервал 366*4=1464 дня.

Архивирование может выполняться:

- Автоматически после истечения указанного периода (см. "Автоматическое архивирование ежедневных табличных пространств");
- Вручную – для архивирования выбранного ЕТП (см. "Архивирование ежедневных табличных пространств вручную").

Автоматическое архивирование ежедневных табличных пространств

Перед тем, как включить автоматическое архивирование, необходимо проверить, что каталог архивирования задан верно:

```
select value
from setting
where setting = 'archive_path';
```

Пользователь `postgres` должен являться владельцем каталога.

Если каталог установлен неправильно, установите его, выполнив следующую команду:

```
begin;
select sp_setting_set('archive_path', '/test/archive/');
commit;
```

Для автоматического архивирования табличных пространств (по умолчанию функция выключена) вы можете использовать сценарии (запускаются от имени владельца схемы данных):

- Для ежедневного табличного пространства, хранящего объекты со статусом *Нарушение*:
  ```
  begin;
  select sp_setting_set('violation_archive_enabled', '1');
  select sp_setting_set('violation_archive_period', 'D');
  commit;
  ```
- Для ежедневного табличного пространства, хранящего объекты со статусом *Нет нарушений*:
  ```
  begin;
  select sp_setting_set('noviolation_archive_enabled', '1');
  select sp_setting_set('noviolation_archive_period', 'D');
  commit;
  ```
- Для ежедневного табличного пространства, хранящего *снимки экрана (скриншоты)*:
  ```
  begin;
  select sp_setting_set('other_archive_enabled', '1');
  select sp_setting_set('other_archive_period', 'D');
  commit;
  ```

где:
1 – показатель того, что автоматическое архивирование включено (чтобы выключить, вместо 1 используйте значение 0);
D – количество дней, по истечении которого ежедневное табличное пространство будет архивировано Системой. Это число должно быть меньше устанавливаемого количества дней до удаления ежедневного ТП (см. "Удаление ежедневных табличных пространств").

Архивирование ежедневных табличных пространств вручную

Архивирование ЕТП вручную производится в порядке, представленном ниже.

Выборка ежедневных табличных пространств и файлов данных

Список отключаемых табличных пространств можно получить, выполнив запросы к базе данных от имени владельца схемы базы данных. Запросы составляются в соответствии со стратегией архивирования, принятой в вашей организации.

Пример. Для выборки ежедневных ТП с указанием основных атрибутов (название, идентификатор, путь, тип, дата создания, размер) можно применить следующий запрос:

```
SELECT t.tbs_name, t.tbs_id, t.tbs_type, t.part_date,
pg_tablespace_location(oid),
pg_size_pretty(pg_tablespace_size(spcname)), t.status, t.note
 FROM  pg_tablespace, iwtm.tbs_list t
 WHERE t.tbs_name = pg_tablespace.spcname AND t.status IN (0, 3);
```

```
tbs_name | tbs_id | tbs_type | part_date  | pg_tablespace_location |
pg_size_pretty | status |        note
----------+--------+----------+------------+-----------------------
+----------------+--------+--------------------
 iwtm_2   |      2 |        1 | 2017-03-30 | /u02/pgdata/iwtm_2     | 692
kB          |      0 | Partitions created
 iwtm_3   |      3 |        2 | 2017-03-30 | /u02/pgdata/iwtm_3     | 692
kB          |      0 | Partitions created
 iwtm_4   |      4 |        0 | 2017-03-31 | /u02/pgdata/iwtm_4     | 25
MB          |      0 | Partitions created
 iwtm_5   |      5 |        1 | 2017-03-31 | /u02/pgdata/iwtm_5     | 692
kB          |      0 | Partitions created
 iwtm_6   |      6 |        2 | 2017-03-31 | /u02/pgdata/iwtm_6     | 692
kB          |      0 | Partitions created
 iwtm_7   |      7 |        0 | 2017-04-01 | /u02/pgdata/iwtm_7     | 692
kB          |      0 | Partitions created
 iwtm_8   |      8 |        1 | 2017-04-01 | /u02/pgdata/iwtm_8     | 692
kB          |      0 | Partitions created
 iwtm_9   |      9 |        2 | 2017-04-01 | /u02/pgdata/iwtm_9     | 692
kB          |      0 | Partitions created
```

Важно!

Настоятельно рекомендуется для получения списка файлов использовать запрос, пример которого описывается далее в этом разделе. Это связано с тем, что список файлов данных, полученный другими способами, может оказаться неполным.

Отключение ежедневных табличных пространств от базы данных

Важно!

Не отключайте ежедневные ТП во время работы заданий iwtm_iwtm_add_parts, iwtm_iwtm_delete_tablespaces, iwtm_iwtm_archive_tablespaces так как это может привести к повреждению данных.

1. От имени владельца схемы базы данных вызовите процедуру:
   `select pkg_part_archive_tablespace(N);`
   где `N` – это значение атрибута `tbs_id` целевого ТП из таблицы `tbs_list`.
   После выполнения этого сценария статус ежедневного ТП изменится на *Отключено от базы данных*.
2. Повторите вызов процедуры для каждого ежедневного ТП, которое нужно отключить.

Перенос файлов данных

Перенесите каталоги с заархивированными табличными пространствами, принадлежащие отключенным ежедневным ТП, на другой носитель информации.

Для получения списка отключенных табличных пространств используйте команду:
`select * from tbs_list where status = 10;`
Файлы данных хранятся в каталоге:
`select value from setting where setting = 'archive_path';`

Восстановление ежедневных табличных пространств

Важно!

Табличное пространство можно восстанавливать только в той схеме базы данных, в которой оно было отключено (даже если эта схема была обновлена). Восстановить табличное пространство после полной переустановки схемы базы данных невозможно.

Перемещение файлов данных

Для восстановления ежедневного ТП необходимо переместить файлы данных этого табличного пространства с внешнего носителя в каталог, путь к которому можно получить, выполнив запрос:

```
select value from setting where setting = 'archive_path';
```

Важно!

Убедитесь, что пользователь *postgres* имеет права на чтение и запись в том каталоге, куда будут перемещаться файлы данных.

Подключение ежедневного табличного пространства

Важно!

Не подключайте ежедневное ТП во время работы заданий iwtm_iwtm_add_parts, iwtm_iwtm_delete_tablespaces и iwtm_iwtm_archive_tablespaces. Это может привести к повреждению данных.

1. От имени владельца схемы базы данных вызовите процедуру:
   `select pkg_part_restore_tablespace(N);`
   где `N` – ID табличного пространства (указывается в tbs_list).
   После выполнения процедуры статус ежедневного ТП изменится на *Восстановлено*.
2. Повторите вызов процедуры для каждого ежедневного ТП, которое нужно подключить.

Настройка размещения файлов в файловой системе

Ежедневные табличные пространства могут храниться либо в одном каталоге, либо распределенно, в разных каталогах на разных дисках.

При установке Системы с помощью поставляемого инсталлятора (см. документ "InfoWatch Traffic Monitor. Руководство по установке") задается использование одной директории ежедневных табличных пространств, расположенной в `/u02/pgdata1/`.

Примечание.

Для архивирования табличных пространств используется директория `/u02/arch`.

Однако при больших нагрузках рекомендуется размещать ежедневные табличные пространства распределенно, на разных файловых системах (LUN-ах, физических дисках) для поочередного их использования. Например, если задано 3 файловых системы, то данные будут размещаться следующим образом:

Первый день – ежедневное табличное пространство создается в файловой системе 1.
Второй день – ежедневное табличное пространство создается в файловой системе 2.
Третий день – ежедневное табличное пространство создается в файловой системе 3.
Четвертый день – ежедневное табличное пространство создается в файловой системе 1.
…

Распределение файлов ежедневных ТП (количество и расположение) можно изменять с помощью следующих сценариев.

Пример сценария, изменяющего количество отдельных мест хранения ежедневных ТП:

```
begin;
select pkg_part_set_df_path_cnt('4');
commit;
```

Важно!

Изменив количество мест хранения ежедневных ТП, обязательно откорректируйте (добавьте/удалите) пути их расположения.

Пример сценария, изменяющего пути для расположения ежедневных ТП:

```
begin;
```

```
select pkg_part_set_df_path('/test1/', 1);
select pkg_part_set_df_path('/test2/', 2);
select pkg_part_set_df_path('/test3/', 3);
select pkg_part_set_df_path('/test4/', 4);
commit;
```

> **Примечание:**
>
> Рекомендуется при указании пути в конце указывать символ «/».

Пример сценария просмотра содержимого ежедневных ТП:

```
select a.d, a.code, power(10, trunc(log(10, a.binary_size + a.text_size))) ||
'-' ||
power(10, trunc(log(10, a.binary_size + a.text_size)) + 1) size_range,
count(1) cnt, sum(a.binary_size) binary_size, sum(a.text_size) text_size
from
(
select date_trunc('day', o.capture_date) d, s.display_name code, o.object_id,
coalesce(length(os.source),0)+coalesce(length(os.context),0)+
coalesce(length(o.gui_xml),0)+coalesce(length(o.preview_data),0) binary_size,
coalesce(length(o.text), 0) text_size
from object o
inner join service s on o.service_code = s.service_id and s.language = 'eng'
inner join object_source os on os.object_id = o.object_id
where o.capture_date between to_date('05.05.2014', 'dd.mm.yyyy') and
to_date('14.05.2014', 'dd.mm.yyyy')
) a
group by a.d, a.code, power(10, trunc(log(10, a.binary_size + a.text_size))) ||
'-' ||
  power(10, trunc(log(10, a.binary_size + a.text_size)) + 1)
order by 1, 2, 3;
```

> **Примечание:**
>
> Чтобы получить результат анализа размеров перехваченных объектов в табличных пространствах вместе со служебной информацией БД, рекомендуется использовать pgAdmin.

Настройка режимов хранения файлов табличного пространства

Если во время установки Система была настроена на режим переноса данных **Normal** (обычный), в процессе эксплуатации режим хранения может быть изменен.

Для того, чтобы настроить режим хранения **Fast/slow** (быстрые и медленные диски), при котором свежие данные сохраняются в быстром разделе и через указанное количество дней перемещаются на медленные разделы, необходимо выполнить следующие действия:

1. Отредактируйте файл `/etc/fstab`, чтобы новый раздел автоматически монтировался в `/u03`
   `/dev/sdd1 /u03/ ext4 defaults 1 2`
2. Создайте папку "pgdata":
   `mkdir /u03/pgdata`
3. Смените пользователя:
   `chown -R postgres:postgres /u03`
4. Зайдите в базу данных:
   `su - iwtm`
   `psql -p 5433 -h <IP-адрес_сервера_БД> postgres iwtm`
   `psql -p 5433 -h 127.0.0.1 postgres iwtm` - для подключения к локальной БД
5. Укажите период хранения на быстрых дисках:
   `select pkg_part_set_fast_days(7);`
6. Укажите путь к быстрому диску
   `select pkg_part_set_fast_path('/u03/pgdata/',1);`
7. Укажите путь к медленному диску:
   `select pkg_part_set_df_path('/u02/pgdata1/',1);`
8. При необходимости можно использовать несколько медленных дисков. Данные будут записываться на следующий медленный диск при заполнении предыдущего. Для добавления дополнительных медленных дисков укажите новый путь:
   `select pkg_part_set_df_path('/u02/pgdata2/',2);`
9. Укажите общее количество медленных дисков:
   `select pkg_part_set_df_path_cnt('2');`
10. Поменяйте режим на медленные-быстрые диски:
    `select pkg_part_set_filesys_type('fast/slow');`
    Данные будут перенесены с быстрых дисков на медленные с помощью ночного задания (джоба).
11. Чтобы запустить перенос файлов вручную, выполните:
    `select iwtm.pkg_part_move_fast_tablespaces_to_slow();`

Для того, чтобы настроить режим хранения **Rotate** (ежедневное переключение), при котором переход к следующему разделу происходит ежедневно и при переполнении предыдущего, необходимо выполнить следующие действия:

1. Монтируйте раздел и убедитесь, что он способен монтироваться автоматически при перезагрузке
2. Создайте папку "pgdata":
   `mkdir /u03/pgdata`
3. Смените пользователя:
   `chown -R postgres:postgres /u03`
4. Зайдите в базу данных:
   `su - iwtm`
   `psql -p 5433 -h <IP-адрес_сервера_БД> postgres iwtm`
   `psql -p 5433 -h 127.0.0.1 postgres iwtm` - для подключения к локальной БД

5. Добавьте путь второго раздела:
   `select pkg_part_set_df_path('/u03/pgdata/',2);`
6. Укажите новое количество путей:
   `select pkg_part_set_df_path_cnt(2);`
7. Переключитесь на режим rotate:
   `select pkg_part_set_filesys_type('rotate');`

```
postgres=# select * from setting where setting like 'df_%';
setting | value | editable
-------------+---------------+----------
df_path1 | /u02/pgdata1/ | 1
df_path2 | /u03/pgdata/ | 1
df_path_cnt | 2 | 1
df_filesys_type | rotate | 1
```

Примечание:

Особенности режимов хранения данных (**normal**, **fast/slow** и **rotate**) описаны в статье базы знаний "Настройка режима хранения данных в ТП. Хранение данных на разных дисках".

Удаление ежедневных табличных пространств

Если дальнейшее хранение данных не требуется, то Вы можете воспользоваться процедурой удаления ежедневных ТП (iwtm_iwtm_delete_tablespaces). Данная процедура позволяет автоматически удалить все данные, хранящиеся в табличном пространстве, сегменты, табличное пространство и файлы данных.

Важно!

1. Удаление табличных пространств – необратимая операция. После выполнения этой операции Вы не сможете восстановить удаленные данные.
2. Процедура iwtm_iwtm_delete_tablespaces не работает с теми табличными пространствами, которые были отключены/подключены вручную.

В результате выполнения этой процедуры удаляются все ежедневные ТП (в т.ч. информация об архивированных ежедневных ТП), которые удовлетворяют следующему условию:

Дата создания табличного пространства меньше или равна разнице между текущей датой и заданным интервалом времени для удаления табличного пространства.

Примечание:

Если архивированное ежедневное ТП не подлежит восстановлению (т.к. информация о нем была удалена из базы данных), вы можете удалить файлы данных этого ТП из архива.

Важно!

Во время удаления табличных пространств доступ к соответствующим таблицам закрыт. По этой причине в лог-файлах процессов **iw_deliver**, **iw_x2db**, **iw_updater** могут отображаться ошибки доступа к базе данных. После удаления ТП эти процессы восстанавливают доступ к БД автоматически.

Рекомендуется запускать задание на удаление данных ежедневно. В противном случае количество удаляемых данных увеличится, что приведет к большим временным затратам на выполнение данной процедуры и, как следствие, к увеличению времени простоя сервера Traffic Monitor.

Важно!

Если в качестве интервала времени используются месяцы или годы, то рекомендуется в расчете использовать максимальное значение интервала. Для месяца – 31 день. Для года – 366 дней. Например, чтобы удалять ежедневные ТП старше 4-х лет, в задании iwtm _iwtm_delete_tablespaces укажите интервал 366*4=1464 дня.

Определение интервала времени для отключения ежедневных ТП
Для автоматического отключения табличных пространств (по умолчанию функция выключена) вы можете использовать различные сценарии (запускаются от имени владельца схемы данных). Установите срок хранения информации об архивных ТП:

- Для ЕТП, хранящего объекты со статусом *Нарушение*:
```
begin;
select sp_setting_set('violation_delete_enabled', '1');
select sp_setting_set('violation_delete_period', 'D');
commit;
```
- Для ЕТП, хранящего объекты со статусом *Нет нарушений*:
```
begin;
select sp_setting_set('noviolation_delete_enabled', '1');
select sp_setting_set('noviolation_delete_period', 'D');
commit;
```
- Для ЕТП, хранящего *снимки экрана (скриншоты)*:
```
begin;
select sp_setting_set('other_delete_enabled', '1');
select sp_setting_set('other_delete_period', 'D');
commit;
```
где:
`1` – показатель того, что автоматическое удаление включено (чтобы выключить, используйте значение 0);
`D` – количество дней, по прошествии которых ежедневное табличное пространство будет отключено в БД. Это число должно быть больше устанавливаемого количества дней до архивирования ежедневного ТП (см. "Автоматическое архивирование ежедневных табличных пространств").
После отключения архивных ТП от БД (статус *Offline*) они становятся недоступны Системе.

Удаление архивированных ТП

При отключении в БД архивированных ежедневных ТП в Системе остаются файлы данных (по умолчанию в директории / `u02/arch)` . Чтобы освободить пространство на жестком диске, можно (на выбор):

- удалить их вручную;
- добавить новые задачи, запускаемые по расписанию в файле `/etc/cron.d/iwtm_cleanup` или в другом файле с задачами в этой директории.

Пример

Чтобы удалить все архивированные ежедневные ТП старше 150 суток:

1. Откройте на редактирование файл `/etc/cron.d/iwtm_cleanup` или другой файл с задачами в этой директории.
2. Добавьте строки:
```
35 3 * * * root find /u02/arch/ -type f -mtime +150 -delete > /dev/null 2>&1 &
40 3 * * * root find /u02/arch/ -type d -ctime +10 -empty -delete > /dev/null 2>&1 &
```
3. Сохраните изменения.
4. Примените новые настройки сервиса `cron` :
```
systemctl reload crond
```

Удаление ежедневных ТП вручную

Если вам требуется немедленно удалить ежедневные ТП (например, из-за нехватки места в файловой системе был изменен интервал удаления, но следующий запуск задания произойдет нескоро), от имени владельца схемы базы данных выполните сценарий:

```
select pkg_part_delete_tablespaces();
```

Если требуется удалить одно ЭТП, выполните сценарий, указав его номер (`tbs_id`):

```
select pkg_part_delete_tablespace(<tbs_id>);
```

Удаление ежедневных ТП с помощью скрипта

Если вам требуется настроить автоматическое удаления ежедневных ТП, используйте скрипт **/opt/iw/tm5/bin/dbtools/dbconf-iwdrop-postgres.sh** или **/opt/iw/tm5/bin/dbtools/dbconf-jobs-postgres.sh.** Возможны команды:

Цель	Команда
Задать период для удаления	`/opt/iw/tm5/bin/dbtools/dbconf-iwdrop-postgres.sh set violation|noviolation|other <days> [-v]`
Включить автоудаление	`/opt/iw/tm5/bin/dbtools/dbconf-iwdrop-postgres.sh enable violation|noviolation|other [-v]`

Цель	Команда
Выключить автоудаление	`/opt/iw/tm5/bin/dbtools/dbconf-iwdrop-postgres.sh disable violation|noviolation|other [-v]`
Просмотреть статус настроек	`/opt/iw/tm5/bin/dbtools/dbconf-iwdrop-postgres.sh show [-v]`
Удалить все ЕТП	`/opt/iw/tm5/bin/dbtools/dbconf-jobs-postgres.sh start iwtm_delete_tablespace`

Пример

Чтобы удалить все объекты со статусом *"Нет нарушения"*, старше 15 суток, выполните команды:

```
/opt/iw/tm5/bin/dbtools/dbconf-iwdrop-postgres.sh set noviolation 15 [-v]
/opt/iw/tm5/bin/dbtools/dbconf-iwdrop-postgres.sh enable noviolation [-v]
```

6.1.4 Резервное копирование базы данных

Для снижения рисков потери данных рекомендуется ежемесячно выполнять создание резервной копии (бэкапа) базы данных. Для хранения бэкапов рекомендуется использовать специально выделенные системы хранения.

Для специалистов, не являющихся администраторами PostgreSQL, то есть не располагающих стандартными методами создания резервной копии БД, рекомендуется описанная ниже процедура «холодного бэкапа», выполняемая на остановленной БД. Данная процедура также может применяться для переноса базы данных с одного сервера на другой.

Далее в разделе:

- Создание резервной копии базы данных;
- Восстановление базы данных из резервной копии.

Создание резервной копии базы данных

Процедура создания резервной копии (выполнения холодного бэкапа) осуществляется в следующем порядке:

- Определение размера резервной копии;
- Проверка хранилища резервной копии;
- Создание каталогов для резервной копии;
- Остановка системы;

- Остановка PostgreSQL;
- Копирование файлов БД в хранилище резервных копий.

Определение размера резервной копии

Чтобы рассчитать размер будущего архива, необходимо узнать суммарный размер каталога с базой PostgreSQL, каталога основного табличного пространства и ежедневных табличных пространств:

1. Войдите в систему от имени пользователя **root**;
2. Получите суммарный размер каталогов:
   ```
   du -sx -BM /u01/postgres/ /u02/pgdata  /u02/pgdata1  /u02/arch
   ```

Если используется другой режим хранения, то данные могут находиться в других директориях (подробнее см. "Настройка режимов хранения файлов табличного пространства").

Проверка хранилища резервной копии

Для проверки приемлемости выбранного хранилища резервной копии:

1. Примонтируйте внешнее хранилище к серверу БД. Оно должно соответствовать следующим требованиям:
   - Размещаться на компьютере, отличном от того, где работает база данных.
   - Быть доступно с компьютеров, где работают сервера и базы данных, подлежащие резервному копированию.
   - Иметь больше свободного места на жестком диске, чем размер резервной копии.
2. Определите количество свободного места на жестком диске:
   a. Выполните следующую команду от имени **root**:
      ```
      OS>df -h
      ```
   b. Убедитесь, что свободного места в примонтированном разделе больше, чем размер резервной копии.

Создание каталогов для резервной копии

> **Внимание!**
>
> Директории файлов резервной копии должны находиться на компьютере, отличном от того, где расположена БД.

Чтобы создать структуру каталогов для бэкапа:

1. Войдите в систему от имени пользователя **root**;
2. Создайте директорию для хранения файлов резервной копии:
   ```
   mkdir /opt/IWTM_Backup_Files
   ```
3. Создайте следующие поддиректории:
   ```
   mkdir /opt/IWTM_Backup_Files/postgres
   mkdir /opt/IWTM_Backup_Files/pgdata
   mkdir /opt/IWTM_Backup_Files/arch
   ```

Остановка системы

Остановка системы является необязательным шагом, но рекомендуется для выполнения, если на сервере мало свободного места.

Чтобы остановить систему:

1. На компьютере, где запущены процессы серверной части Trafic Monitor, зайдите в систему от имени пользователя **root**.
2. Остановите все запущенные процессы:
   `iwtm stop`
   По окончании процедуры резервного копирования сервисы можно запустить с помощью следующих команд:
   `iwtm start`

Остановка PostgreSQL

> **Важно!**
>
> Перед началом резервного копирования файлов базы данных обязательно нужно остановить БД PostgreSQL.

Чтобы остановить БД PostgreSQL:

1. На компьютере, где работает база данных, зайдите в систему от имени пользователя **root**.
2. В командной строке введите:
   ```
   service pgagent-9.6 stop
   service postgresql-9.6 stop
   ```
   По окончании процедуры резервного копирования сервисы можно запустить с помощью следующей команды:
   ```
   service pgagent-9.6 start
   service postgresql-9.6 start
   ```

Копирование файлов БД в хранилище резервных копий

1. Остановите все сервисы PostgreSQL (см. "Остановка PostgreSQL"). Чтобы убедиться в этом, выполните команду:
   `ps aux | grep postgre`
   Если сервисы Postgre SQL не остановлены, файлы резервной копии могут быть повреждены и оказаться непригодными для восстановления.
2. На компьютере, где установлена БД, скопируйте директории (со всем содержимым) с помощью ранее созданного списка директорий.
   Если система была установлена с помощью инсталлятора и были оставлены параметры по умолчанию, то:
   - скопируйте содержимое директории **/u01/postgres/** в директорию **/opt/IWTM_Backup_Files/postgres**
   - скопируйте содержимое директории **/u02/pgdata/** в директорию **/opt/IWTM_Backup_Files/pgdata**

- скопируйте содержимое директории **/u02/pgdata1/** в директорию **/opt/IWTM_Backup_Files/pgdata1**
- скопируйте содержимое директории **/u02/arch/** в директорию **/opt/IWTM_Backup_Files/arch**

> **Примечание:**
>
> В качестве хранилища файлов необходимо использовать только:
>
> - внешний диск с файловыми системами Ext4, XFS;
> - удаленное блочное устройство, подключенное по протоколам iSCSI, NFS;
> - локально подключенное блочное устройство с файловыми системами Ext4, XFS.

Восстановление базы данных из резервной копии

С учетом причины падения вашей базы данных, выберите подходящую процедуру восстановления БД:

- Если БД была повреждена вследствие сбоя системы или ошибки пользователя, восстановите старую БД. Например, если случайно был удален важный файл, вы можете восстановить БД до состояния, когда этот файл еще существовал (см. "Восстановление на той же базе данных").
- Если старая БД не может больше использоваться, создайте новую и восстановите данные на ней (см. "Восстановление на новой базе данных").

> **Важно!**
>
> Не допускается обновлять Traffic Monitor на другую версию до или в процессе восстановления Базы данных.

Восстановление на той же базе данных

Ниже описана процедура восстановления на БД, имеющей ту же структуру каталогов, что и та, с которой была создана резервная копия.

Чтобы восстановить базу данных с помощью создания новой базы данных:

1. Убедитесь в работоспособности БД. Проверьте существующую схему БД, сервер БД, где размещена эта схема, и компьютер, на котором работает сервер БД;
2. Остановите сервисы PostgreSQL:
   ```
   service postgresql-9.6 stop
   service pgagent-9.6 stop
   ```
3. Установите БД PostgreSQL согласно инструкции, приведенной в документе « InfoWatch Traffic Monitor. Руководство по установке ».
4. Выполните следующие шаги :

a. Удалите все содержимое каталогов **/u01/postgres/**, **/u02/pgdata/**, **/u02/pgdata1/**, **/u02/arch/**
b. Скопируйте содержимое директории **/opt/IWTM_Backup_Files/postgres** в директорию **/u01/postgres/**
c. Скопируйте содержимое директории **/opt/IWTM_Backup_Files/pgdata/** в директорию **/u02/pgdata/**
d. Скопируйте содержимое директории **/opt/IWTM_Backup_Files/pgdata1/** в директорию **/u02/pgdata1/**
e. Скопируйте содержимое директории **/opt/IWTM_Backup_Files/arch/** в директорию **/u02/arch/**

5. Проверьте права. При необходимости, измените их:
```
chown postgres /u01/postgres/ -R
chown postgres /u02/pgdata/ -R
chown postgres /u02/pgdata1/ -R
chown postgres /u02/arch/ -R
```

6. Запустите базу данных:
```
service postgresql-9.6 start
service pgagent-9.6 start
```

Восстановление на новой базе данных

При восстановлении PostgreSQL, необходимо копировать файлы БД в те же каталоги, в которых они были сохранены.

Чтобы восстановить БД, скопируйте каталоги, проверьте права и запустите сервисы БД (см. "Восстановление на той же базе данных").

6.1.5 Проведение регламентных работ на сервере базы данных

Важно!

Категорически не рекомендуется выключать сервер БД кнопкой питания. В некоторых случаях это может привести к повреждению БД.

При выполнении регламентных работ на сервере базы данных придерживайтесь такого порядка:

1. Закройте все окна браузера, отображающие Консоль управления. Убедитесь, что отсутствуют соединения со схемой БД Traffic Monitor из других программ. Если такие соединения есть, отключите их.
2. На сервере Traffic Monitor остановите процессы TM:
```
iwtm stop
service iwtm-php-fpm stop
service nginx stop
```
3. На сервере базы данных получите список заданий, запускающихся по расписанию. Для этого в psql, из-под учетной записи владельца схемы выполните запрос:
```
select jobname,  case when jobagentid is null then 'scheduled' else 'running' end state
```

```
from pgagent.pga_job
where jobenabled= true;
```

4. Выключите задания, выполнив сценарий:
```
select iwtm.pkg_utility_disable_job('JOB_1);
```
...
```
select iwtm.pkg.utility_disable_job('JOB_N');
commit;
```
где `JOB_1… JOB_N` – имена выключаемых заданий.
5. Убедитесь, что ни одно задание не выполняется. Для этого выполните запрос:
```
select pga_job.jobname
from pgagent.pga_job
where pga_job.jobagentid is not null;
```
текущее задание невозможно остановить из PostgreSQL, это следует осуществить посредством остановки агента из Linux с помощью команды:
`service pgagent-9.6 stop` (остановка всех задач при помощи единой команды)
6. Выполните необходимые работы с базой данных.
7. По окончании необходимых работ с сервера Traffic Monitor проверьте соединение с сервером базы данных:
```
psql -p 5433 -h server_name postgres postgres
```
где `server_name` - имя или ip адрес базы данных
Если проверка пройдена успешно, тогда в ответ на выполнение команды будет выведено следующее приглашение psql:
```
postgre=#
```
8. Запустите процессы Traffic Monitor Server:
```
iwtm start
service iwtm-php-fpm start
service nginx start
```
9. Проверьте системный журнал на наличие ошибок. Путь к файлу журнала:
```
/var/log/messages
```
10. Если в системном журнале содержится информация об ошибках, то обратитесь в службу технической поддержки.
11. Включите выполнение ранее отключенных заданий:
```
BEGIN
```
select iwtm.pkg_utility_enable_job('JOB_1');
...
select iwtm.pkg_utility_enable_job('JOB_N');
commit;
end;
/
где `JOB_1… JOB_N` – имена ранее остановленных заданий.

6.1.6 Логирование базы данных

Логированием работы СУБД PostgreSQL в Traffic Monitor управляет скрипт, который позволяет устанавливать уровень логирования и выгружать данные лога в CSV-файл. При работе скрипта используются настройки текущего конфигурационного файла `/opt/iw/tm5/csw/database.conf`. Запускать скрипт следует из директории его местонахождения: **/** `opt/iw/tm5/csw/postgres/scripts/sys_log/sys_log.sh`.

Скрипт поддерживает следующие команды:

- чтобы установить уровень логирования: `debug, info, warning, error –` выполните команду:
  `level`
  при этом выводится текущий уровень логирования при запуске без второго параметра;
- чтобы выгрузить тело лога (представление V_SYS_LOG) в CSV-файл, выполните команду:
  `export`
  при этом, если требуется, можно задать количество выгружаемых строк:
  `export <кол-во строк>`

О хранении логов на диске см. "Ротация логов базы данных".

Ротация логов базы данных

Ротация логов базы данных - это процесс архивирования, хранения и удаление старых данных в журналах, нужный для экономии и освобождения свободного дискового пространства. В Traffic Monitor ротация настроена на лог БД в текстовом формате: `/u01/postrges/pg_log/*.log.` Он ротируется службой **logrotate** по 100 МБ и удаляется после превышения количества в 1000 файлов.

Настройки службы **logrotate** находятся в `/etc/logrotate.d/iwtm-pg-conf.`

7 Администрирование серверной части InfoWatch Traffic Monitor

В этой главе описаны компоненты серверной части Traffic Monitor и методы их использования:

- Процессы серверной части Traffic Monitor Server;
- Настройка конфигурационных файлов серверной части Traffic Monitor;
- Настройка параметров обработки архивов вложений;
- Архивирование каталога очереди сообщений;
- Логирование работы Системы;
- Файловые очереди;
- Восстановление работоспособности системы в аварийных ситуациях.
- Настройка передачи информации в SIEM
- Удаление временных файлов

7.1 Процессы серверной части Traffic Monitor Server

В этом разделе:

- Список процессов серверной части Traffic Monitor;
- Настройка конфигурационных файлов серверной части Traffic Monitor;
- Работа с процессами серверной части Traffic Monitor.

7.1.1 Список процессов серверной части Traffic Monitor

Работа Системы осуществляется посредством процессов. Один и тот же процесс может быть запущен единовременно в нескольких экземплярах.

Назначение	Имя процесса	Описание процесса	Конфигурационный файл
Сбор данных	iw_icap	Обрабатывает HTTP-трафик. Принимает HTTP-запросы от ICAP-клиента. Извлекает данные из HTTP-запросов. Затем извлеченные данные добавляются в XML-контекст. Готовый XML-контекст передается подсистеме анализа и принятия решения для проверки. По окончании анализа передает ICAP-клиенту ответ с разрешением/запрещением на доставку HTTP-запроса. Также передает HTTP-запрос процессу **iw_x2db** для сохранения в базу данных	/opt/iw/tm5/etc/icap.conf

Назначение	Имя процесса	Описание процесса	Конфигурационный файл
	iw_sniffer	Перехватывает трафик, который передается по протоколам SMTP, HTTP, ICQ, POP3, IMAP, NRPC	/opt/iw/tm5/etc/sniffer.conf
	iw_proxy	Принимает копию трафика и передает ее модулю iw_messed, разбив на http-, icq- и smtp-трафик. Включает следующие процессы: • iw_proxy_http - процесс, принимающий копию HTTP-трафика. Принимает HTTP-запрос, формирует XML-контекст из полученного объекта. Затем XML-контекст передается подсистеме анализа и принятия решения **iw_analysis**. • iw_proxy_icq - процесс, принимающий копию ICQ-трафика. Принимает ICQ-сообщение, формирует XML-контекст из полученного объекта. Затем XML-контекст передается подсистеме анализа и принятия решения **iw_analysis**. • iw_proxy_smtp - процесс, принимающий копию SMTP-трафика. Принимает SMTP-письмо, формирует XML-контекст из полученного объекта. Затем XML-контекст передается	/opt/iw/tm5/etc/proxy.conf

Назначение	Имя процесса	Описание процесса	Конфигурационный файл
		подсистеме анализа и принятия решения. По окончании анализа копия объекта передается процессу **iw_x2db** для укладки в базу данных	
	iw_smtpd	Принимает SMTP-письма. В случае интеграции с почтовым сервером Postfix принимает входящие сообщения от Postfix. Если интеграция с Postfix отсутствует, сообщения принимаются от корпоративного почтового сервера или от почтового клиента (в зависимости от настроек Вашей почтовой системы). Принимает входящие сообщения в формате SMTP, преобразует в XML-контекст данные SMTP-конверта. Затем передает SMTP-письмо и XML-контекст процессу **iw_messed**	/opt/iw/tm5/etc/smtpd.conf
	iw_capstack	Выполняет обработку трафика, передаваемого по протоколам POP3, IMAP, NRPC	/opt/iw/tm5/etc/capstack.conf
	iw_messed	Обрабатывает SMTP-писема. Извлекает данные из SMTP-, POP3- и IMAP-объектов. Затем извлеченные данные добавляются в полученный XML-контекст. Готовый XML-контекст, содержащий данные конверта и письма, передается подсистеме анализа и принятия решения для проверки. По окончании анализа **iw_messed** передает SMTP-письма, доставка которых разрешена, компоненту	/opt/iw/tm5/etc/messed.conf

Назначение	Имя процесса	Описание процесса	Конфигурационный файл
		почтовой системы, ответственному за доставку почты (только в нормальном и прозрачном транспортном режиме). В случае интеграции с почтовым сервером Postfix, таким компонентом является Postfix. Если интеграция с Postfix отсутствует, то, в зависимости от настроек вашей почтовой системы, таким компонентом может быть корпоративный почтовый сервер или почтовый клиент. Кроме того, копия проверенного SMTP-письма передается процессу **iw_x2db** для сохранения в базу данных	
	iw_xapi	Включает в себя две службы: **iw_xapi_xapi** и **iw_xapi_puppy**. Получает объекты от Infowatch Device Monitor и внешних систем (через адаптеры) по thrift-интерфейсу, складывает в файловую очередь. Далее отправляет объекты процессу **iw_analysis** . При получении eml-объектов передает их процессу **iw_messed**	/opt/iw/tm5/etc/xapi.conf
	iw_analysis	Забирает объекты из файловой очереди, в которую их кладет **iw_xapi_xapi/ iw_xapi_puppy, iw_proxy_icq, iw_proxy_http**. Посредством файловой очереди отправляет на обработку процессам **iw_warpd**, **iw_cas**, **iw_pas**, **iw_luaengined**. Далее объекты складываются в файловую очередь для отправки в базу данных процессом **iw_x2db**	/opt/iw/tm5/etc/analysis.conf

Назначение	Имя процесса	Описание процесса	Конфигурационный файл
Обработка данных	iw_warpd	Управляет процессами извлечения данных из контейнеров, вложенных в перехваченные объекты	/opt/iw/tm5/etc/warpd.conf
	iw_image2text_fre	Осуществляет распознавание текста в изображениях при помощи OCR ABBYY FineReader	/opt/iw/tm5/etc/image2text_fre.conf
	iw_image2text_ts	Осуществляет распознавание текста в изображениях при помощи OCR Tesseract	/opt/iw/tm5/etc/image2text_ts.conf
	iw_cas	Выполняет роль сервера контентного анализа. Процесс **iw_cas** принимает от подсистем перехвата текстовые запросы в формате plain-text для проведения контентного анализа. По окончании контентного анализа возвращает результат запроса подсистеме анализа и принятия решения	/opt/iw/tm5/etc/cas.conf
	iw_image_autoling	Выполняет автоматическую классификацию графических объектов	/opt/iw/tm5/etc/iw_image_autoling.conf
	iw_text_autoling	Выполняет автоматическую классификацию текстовых объектов	/opt/iw/tm5/etc/iw_text_autoling.conf
	iw_pas	Получает результаты анализа **iw_cas**. Определяет наличие объекта защиты и добавляет объекту соответствующие атрибуты	/opt/iw/tm5/etc/pas.conf
	iw_luaengined	Обеспечивает выполнение LUA-скрипта согласно действующей политике	/opt/iw/tm5/etc/luaengined.conf

Назначение	Имя процесса	Описание процесса	Конфигурационный файл
	iw_x2x	Получает данные с xml+dat-файлами посредством файловой очереди. Полученные файлы модифицируются и отправляются процессу **iw_x2db**	/opt/iw/tm5/etc/x2x.conf
	iw_tech_tools	Верифицирует условия для выгрузок из БД и регулярные выражения, позволяет нормализовать текст в соответствии с регулярными выражениями	/opt/iw/tm5/etc/tech_tools.conf
Загрузка в БД	iw_qmover_client	Работает на Traffic Monitor Server, установленном в филиале. Отправляет перехваченные объекты в базу данных центрального офиса	/opt/iw/tm5/etc/qmover_client.conf
	iw_qmover_server	Работает на Traffic Monitor Server, установленном в центральном офисе. Принимает объекты, полученные от Агента, установленного в филиале. Передает через **iw_x2x** объекты процессу **iw_x2db** для сохранения в базу данных	/opt/iw/tm5/etc/qmover_server.conf
	iw_x2db	Загружает в БД объекты, проверенные подсистемой анализа и принятия решения, из выходной файловой очереди процесса **iw_x2x**	/opt/iw/tm5/etc/x2db.conf
Инфраструктура	iw_adlibitum	Управляет процессами получения актуальных данных с сервера Active Directory, Domino Directory, Samba DC, Astra Linux Directory, Astra Linux Directory Pro и FreeIPA	/opt/iw/tm5/etc/adlibitum.conf

Назначение	Имя процесса	Описание процесса	Конфигурационный файл
	iw_agent	Требуется для управления конфигурацией Системы	/opt/iw/tm5/etc/agent.conf
	iw_blackboard	Осуществляет взаимодействие применяемых политик и БД	/opt/iw/tm5/etc/blackboard.conf
	iw_bookworm	Выполняет роль справочника в Системе	/opt/iw/tm5/etc/bookworm.conf
	iw_cas_config_compiler	Переводит конфигурационный файл сервера контентного анализа в бинарный вид для возможности использования конфигурации в контентном анализе	/opt/iw/tm5/etc/cas_config_compiler.conf
	iw_configerator	Формирует конфигурацию, которая отправляется в Device Monitor	/opt/iw/tm5/etc/configerator.conf
	iw_deliver	Выполняет доставку писем. Отправляет SMTP-письма получателям в случае, если доставка письма разрешена из Management Console (только в нормальном и прозрачном транспортных режимах). Также этот процесс доставляет SMTP-письма, которые по той или иной причине (например, ввиду отсутствия связи с почтовым relay-сервером или почтовым клиентом) не смог доставить процесс **iw_messed**	/opt/iw/tm5/etc/deliver.conf
	iw_metainfo_fetcher	Осуществляет выборку из БД данных с метаинформацией о всех контактах, в том числе о контактах, участвующих в событиях, для индексации. Отправляет их процессу **iw_is**	-

Назначение	Имя процесса	Описание процесса	Конфигурационный файл
	iw_audit_fetcher	Осуществляет выборку из БД данных событий расширенного аудита для индексации. Отправляет их процессу **iw_is**	-
	iw_is	Управляет процессом индексирования данных и работой поисковых серверов для полнотекстового поиска. Подготавливает файлы с описанием данных, которые через файловые очереди доходят до индексатора, и запускает его	/opt/iw/tm5/etc/is.conf
	iw_indexer	Индексирует текст в событиях из БД. Получает доступ к базе данных для ее индексации и складывает индексы в файловое хранилище. При выполнении поискового запроса **sphinx** получает id объектов из файлового хранилища индексов	/opt/iw/tm5/etc/indexer.conf
	iw_kicker	Служит для корректной работы WebGUI, осуществляет запуск сервисов: agent, blackboard, export, import, report, notifier, selection, systemcheck, xapisamplecompiler, samplecompiler, querytracker, reporttracker	/opt/iw/tm5/etc/kicker.conf
	iw_licensed	Подсистема лицензирования. Производит мониторинг и обработку установленных в Системе лицензий	/opt/iw/tm5/etc/licensed.conf
	iw_rammer	Выполняет досылку писем с ошибками обработки при работе "в разрыв"	/opt/iw/tm5/etc/rammer.conf

Назначение	Имя процесса	Описание процесса	Конфигурационный файл
	iw_sample_compiler	Создает цифровые отпечатки из загруженных эталонных файлов	/opt/iw/tm5/etc/sample_compiler.conf
	consul	Производит обнаружение, регистрацию и мониторинг доступности сервисов, взаимодействующих с интерфейсом пользователя	/opt/iw/tm5/etc/consul/consul.json
	iw_system_check	Собирает данные от службы Nagios и предоставляет их для отображения в Консоли управления	/opt/iw/tm5/etc/system_check.conf
	iw_updater	Загружает конфигурацию из базы данных на Traffic Monitor Server	/opt/iw/tm5/etc/updater.conf

Внимание!

В данной версии ТМ перехват для Astra Linux осуществляется на стороне DM и перехватчиком **iw_smtpd** на стороне ТМ. Поэтому следующие сервисы ТМ должны быть установлены, но отключены по умолчанию до принудительного запуска:

- iw_icap_buf
- iw_icap
- iw_sniffer
- iw_proxy
- iw_capstack

7.1.2 Настройка конфигурационных файлов серверной части Traffic Monitor

Полный перечень конфигурационных файлов Системы и описание настроек, которые вы можете изменять для отражения специфики работы Системы в вашей инфраструктуре, находятся в документе "*Справочник по конфигурационным файлам*". Файлы, не описанные в данном документе, изменять не рекомендуется.

Расположение конфигурационных файлов: `/opt/iw/tm5/etc`.

Конфигурационные файлы Системы имеют формат JSON. Названия конфигурационных файлов соответствуют названиям тех компонентов, для настройки которых они используются. Например, для конфигурирования компонента **iw_x2x** используется конфигурационный файл **x2x.conf**.

Различают общие настраиваемые секции параметров (см. документ "*Справочник по конфигурационным файлам*", статьи: *"Общая секция Bookworm", "Общая секция Discovery", "Общая секция Logging", "Общая секция Statistics", "Общая секция Statistic", "Общая секция ThriftServers"*) и параметры, специфичные для каждого из конфигурационных файлов (см. описание отдельных файлов в документе "*Справочник по конфигурационным файлам*").

7.1.3 Работа с процессами серверной части Traffic Monitor

Список процессов Traffic Monitor представлен в главе "Список процессов серверной части Traffic Monitor".

Все процессы Traffic Monitor Server запускаются от имени пользователя **iwtm**, который создается в процессе установки Traffic Monitor Server автоматически (подробнее о учетных записях см. документ «InfoWatch Traffic Monitor. Руководство по установке»).

> **Примечание:**
>
> Имя пользователя прописывается в unit-файлах процессов в параметре `User`. Для стабильной работы Системы рекомендуется оставить значение этого параметра без изменений.

Во время работы состояние процессов Traffic Monitor Server проверяется сценарием **pguard**, который создается в директории `/opt/iw/tm5/bin` в ходе установки. Если по какой-либо причине один или несколько процессов Traffic Monitor Server были остановлены, сценарий **pguard** перезапускает эти процессы.
Управление работой процессов Traffic Monitor Server выполняется при помощи bash-скрипта **/opt/iw/tm5/bin/iwtm**, который является сценарием автозапуска для уровней запуска (runlevel) 2, 3, 4, 5.

> **Примечание:**
>
> 1. Для перехвата копии трафика через Sniffer (ICQ- HTTP-, SMTP-трафик) запускаются отдельные процессы **iw_proxy**.
> 2. Процессы **iw_qmover_server, iw_qmover_client** доступны, но для них по умолчанию выключен автозапуск. Каждый процесс запускается на соответствующей стороне (в филиалах **iw_qmover_client,** в центральном офисе **iw_qmover_server**).
> 3. Процесс **iw_rammer** не установлен по умолчанию. После ручной установки он становится доступен в списке процессов.

Ключи запуска для сценария **iwtm**:

Ключ запуска	Назначение
`start`	запуск процессов
`stop`	безопасная остановка процессов

Ключ запуска	Назначение
`status`	вывод на экран информации о состоянии процессов (запущен/не запущен/доступен/не доступен)
`restart`	перезапуск процессов (последовательное выполнение start+stop)
`reload`	перезагрузка конфигурации процесса
`kill`	передача процессу сигнала для его немедленного завершения (SIGKILL)
`test`	вывод на экран всех доступных демонов iwtm
`enable`	назначение демону статуса "доступен" - автозапуск демона разрешен
`disable`	назначение демону статуса "недоступен" - автозапуск демона запрещен
`services_state`	вывод на экран всех доступных демонов iwtm с побитовыми масками

Примеры команд

Команда	Описание
`iwtm status`	Проверить состояние процессов Traffic Monitor Server

Любая команда на выбор:
`service iw_cas start` (если требуется просмотр логов)
`iwtm start cas`
`iwtm start iw_cas`
`iwtm start iw_cas.service`
`systemctl start iw_cas` (без возможности просмотра логов)
`systemctl start iw_cas.service`
	Запустить сервер контентного анализа (CAS)

Важно!

В каталоге `/opt/iw/tm5/run` хранятся PID-файлы. При аварийном завершении работы необходимо перед запуском процессов стереть все PID-файлы.

Изменение параметров автозапуска процессов

Информация о необходимости включения и отключения автозапуска процессов изложена в статье "Проверка автозапуска процессов".

Для каждого процесса, управляемого bash-скриптом **iwtm,** можно задать возможность автозапуска в командной строке. Эта настройка описана в статье "Включение и выключение автозапуска процессов". Подробнее о командах systemd в статье "Команды для работы с systemd".

7.2 Настройка использования OCR

Технология OCR предназначена для перевода изображений рукописного, машинописного или печатного текста в текстовые данные. По умолчанию технология отключена для всех сервисов, типов событий и связанных с ними протоколов. OCR используется для:

- анализа перехваченных изображений,
- анализа изображений, загруженных как эталонные документы.

В первом случае извлечением данных из контейнеров, вложенных в перехваченные объекты, занимается процесс **iw_warpd**. Во втором случае процесс **iw_sample_compiler** создает цифровые отпечатки из загруженных эталонных файлов. Включение и настройка технологии OCR осуществляется в их конфигурационных файлах. Подробнее см. Настройка OCR-экстракторов.

При необходимости, использование OCR для анализа перехваченных изображений можно настроить следующими способами:

Настройка перехвата на уровне:	Настройка применяется к содержимому событий,	Приоритет выполнения настройки
Сервиса	Полученных по протоколам типов событий этого сервиса (если для данных типов событий или протоколов не задана своя настройка OCR)	Низкий
Типа события для конкретного сервиса	Полученных по протоколам этого типа события (если для этих протоколов не задана своя настройка OCR)	Средний
Протокола для конкретного типа события	Полученных по выбранному протоколу для указанного типа события	Высокий

Настройки OCR задаются в справочнике демона **iw_bookworm** в каталоге `/opt/iw/tm5/etc/config-perm/bookworm:`

- `ocr.xml` - предустановленные настройки, используемые по умолчанию;
- `ocr_custom.xml` - пользовательские настройки для конкретного типа внедрения.

Для включения использования технологии OCR при внедрении необходимо внести изменения в файле `ocr_custom.xml` (подробнее см. "Конфигурационный файл ocr_custom.xml"), а именно :

1. Выбрать конкретные объекты: сервисы, протоколы и типы событий.

   > **Примечание:**
   >
   > Новые типы событий, зарегистрированные через плагин, автоматически добавляются в `/opt/iw/tm5/etc/config/bookworm/services.xml` и становятся доступными для обработки и переноса в файл `ocr_custom.xml`.

2. Задать идентификаторы объектов справочника сервисов и типов событий (файлы с описанием находятся в каталоге `/opt/iw/tm5/etc/config/bookworm/`).
3. Включить использование OCR (`ocr_enabled="true"`).

При необходимости настройки OCR для нескольких нод (распределенная установка) необходимо задать правила для каждой из нод. Для этого нужно:

1. Создать новый xml-файл для описания ноды в каталоге `/opt/iw/tm5/etc/config-perm/bookworm`.
2. Указать созданный xml-файл в секции `CustomNodeXMLPath` конфигурационного файла `/opt/iw/tm5/etc/bookworm.conf`
3. Указать в параметре `<ocr_option node>` название ноды из файла `/opt/iw/tm5/node_name`.
4. Описать переопределяемые правила для ноды, как описано выше.

> **Примечание:**
>
> В этом случае будет установлен следующий приоритет настроек (по убыванию): справочник с настройками для ноды -> справочник с пользовательскими настройками -> справочник с предустановленными настройками.

> **Важно!**
>
> При обновлении Системы на текущую версию, параметры OCR будут сброшены на предустановленные. В случае добавления или изменения функциональных возможностей OCR новые настройки будут доступны в файле `ocr.xml`. Пользовательские настройки останутся в файле `ocr_custom.xml` без изменений.

7.2.1 Конфигурационный файл ocr_custom.xml

Примеры правил OCR в выключенном состоянии представлены ниже, где:

- `object_type` – тип события;
- `service` – сервисы;
- `protocol` – протоколы;
- `key` - ключи;
- `mnemo` – идентификаторы объектов справочника типов событий, сервисов и протоколов.

При включении правил имеет смысл указывать не все, а только необходимые из них:

```
<ocr_options node="*" ocr_enabled="false">
  <service key="CE6D8E0E27DA11E2962FC1DB6088709B00000000" mnemo="email" ocr_enabled="false">
    <object_type key="D2B5132E27DA11E28444C2DB6088709B00000000" mnemo="email" ocr_enabled="false">
      <protocol key="501A9868F3460E2E5AAF0C6AB21F8D6F63C09109" mnemo="pop3" ocr_enabled="false"/>
      <protocol key="2ECB2A264E31677895BC5D8845D33BC26F637CDF" mnemo="imap" ocr_enabled="false"/>
      <protocol key="A324CCB227DA11E2A8D99FDB6088709B00000000" mnemo="smtp" ocr_enabled="false"/>
      <protocol key="5F7E60DB3E86EFF90FDA87E72209EA1B0E017F16" mnemo="mapi" ocr_enabled="false"/>
      <protocol key="93B79EB44EE1A45522C08319EC47B499294776D7" mnemo="nrpc" ocr_enabled="false"/>
    </object_type>
    <object_type key="D8333C9027DA11E29E34CADB6088709B00000000" mnemo="email_web" ocr_enabled="false">
      <protocol key="A779DB3627DA11E2ADDCA0DB6088709B00000000" mnemo="http" ocr_enabled="false"/>
      <protocol key="7ED97C84BDBDD99C1C21AC0A6D6191F6A891C440" mnemo="https" ocr_enabled="false"/>
    </object_type>
  </service>
  <service key="DD6ECB5227DA11E2B507CBDB6088709B00000000" mnemo="im" ocr_enabled="false">
    <object_type key="E266719627DA11E2A83ECCDB6088709B00000000" mnemo="im_icq" ocr_enabled="false">
      <protocol key="ABA5D02027DA11E2B78FA1DB6088709B00000000" mnemo="oscar" ocr_enabled="false"/>
    </object_type>
    <object_type key="EEC2D87627DA11E29EA6D2DB6088709B00000000" mnemo="im_mail_ru" ocr_enabled="false">
      <protocol key="B8C7FC6A27DA11E2A080B7DB6088709B00000000" mnemo="mmp" ocr_enabled="false"/>
    </object_type>
    <object_type key="F249A80827DA11E29BA2D3DB6088709B00000000" mnemo="im_skype" ocr_enabled="false">
      <protocol key="DD3D593857521DE2E9618C26FD3E1B914F8A16F9" mnemo="skype" ocr_enabled="false"/>
    </object_type>
    <object_type key="E7C3A26C27DA11E296D3CDDB6088709B00000000" mnemo="im_xmpp" ocr_enabled="false">
      <protocol key="AF69EA3427DA11E2BE09A2DB6088709B00000000" mnemo="xmpp" ocr_enabled="false"/>
    </object_type>
  </service>
  <service key="62E7C68AD1354D118282FAFF07DA59ED00000000" mnemo="multimedia" ocr_enabled="false">
    <object_type key="55E10E81F5C94B3FB742CB61CA9476F800000000" mnemo="multimedia_photo" ocr_enabled="false
"/>
  </service>
  <service key="0794BBB227DB11E2BD81E9DB6088709B00000000" mnemo="web" ocr_enabled="false">
    <object_type key="0CAFCC9027DB11E2AD27EDDB6088709B00000000" mnemo="web_common" ocr_enabled="false">
      <protocol key="A779DB3627DA11E2ADDCA0DB6088709B00000000" mnemo="http" ocr_enabled="false"/>
      <protocol key="7ED97C84BDBDD99C1C21AC0A6D6191F6A891C440" mnemo="https" ocr_enabled="false"/>
    </object_type>
  </service>
  <service key="111CBA0427DB11E2AB82EEDB6088709B00000000" mnemo="file" ocr_enabled="false">
    <object_type key="1515A7B027DB11E2BBB2EFDB6088709B00000000" mnemo="file_exchange" ocr_enabled="false">
      <protocol key="BEFB1A7C27DA11E2AB10B8DB6088709B00000000" mnemo="ftp" ocr_enabled="false"/>
    </object_type>
    <object_type key="190D004827DB11E287B1F0DB6088709B00000000" mnemo="file_copy_out" ocr_enabled="false"/>
  </service>
  <service key="1DA8A90427DB11E289E8F5DB6088709B00000000" mnemo="print" ocr_enabled="false">
    <object_type key="225BD8F427DB11E2926DF9DB6088709B00000000" mnemo="print_common" ocr_enabled="false"/>
  </service>
  <service key="7843FC5BEA024E9B274E26DB43B7E680D8BC9356" mnemo="placement" ocr_enabled="false">
    <object_type key="602A224D9335579214E3188D1D2745DB9F85D500" mnemo="crawler" ocr_enabled="false"/>
  </service>
</ocr_options>

<ocr_options node="*" ocr_enabled="false">
</ocr_options>
```

Примечание:

Для включения OCR на всех уровнях сразу (на уровне сервиса, типа события и на уровне протокола) необходимо в нижней строке файла `ocr_custom.xml` заменить

```
<ocr_options node="*" ocr_enabled="false">
```

на

```
<ocr_options node="*" ocr_enabled="true">
```

7.3 Настройка параметров обработки архивов вложений

Обработка вложений настраивается в двух конфигурационных файлах:

- файл `/opt/iw/tm5/etc/extractors.conf` (см. документ "*Справочник по конфигурационным файлам*", статья "extractors.conf");
- файл `/opt/iw/tm5/etc/config-perm/bookworm/extractors.xml` (см. "Конфигурационный файл extractors.xml").

Важно!

Если в MS Exchange используется правило журналирования, для корректной обработки этих писем включите соответствующую опцию в Traffic Monitor. Для этого в конфигурационном файле `messed.conf` укажите:

```
"ExchangeJournalReports": true,
```

Чтобы опция стала активной, повторно загрузите конфигурацию или перезапустите службу.

7.3.1 Конфигурационный файл extractors.xml

Файл **/opt/iw/tm5/etc/config-perm/bookworm/extractors.xml** содержит настройки справочника и базу сигнатур, с помощью которых определяется тип файлов при распаковке архивов и вложений.

Параметр	Описание и примеры настройки
`FilenameCharsets`	Если кодировка имени файла отличается от стандартной (ANSI, UTF), то файл будет обрабатываться как соответствующий кодировкам, указанным в данном параметре. Несколько значений перечисляются через запятую; Система будет последовательно пытаться обработать файл в перечисленных кодировках, пока не найдет похожую. Значения по умолчанию – `UTF-8,cp1251,cp866`
`MinFileSizeInBytes`	Минимальный размер файла (в байтах), подлежащего распаковке. Это ограничение позволяет увеличить производительность Системы, за счет отказа от распаковки небольших файлов. Значение по умолчанию – `10` **Примечание:** Обработка файлов размером меньше 10 байт не поддерживается.

	Данное ограничение также распространяется на тип событий Буфер обмена.
`MaxTextPlainSizeInKb`	Верхняя граница размера файла, сигнатура которого не определена. При превышении порогового значения файл автоматически считается бинарным и не отправляется на анализ в **iw_cas**. Значение по умолчанию – `25600`
`SpeedInKBs`	Предположительная скорость работы экстрактора в расчете на 1 ГГц частоты процессора (в Кб/с). Может принимать целочисленные значения от 1 и выше. Значение по умолчанию – `100`
`TimeoutInSec`	Время ожидания до завершения обработки файла (в секундах). Значение по умолчанию – `1800`
`UseLog`	Логирование экстрактора. Значение по умолчанию – `false`

Если общие значения параметров `TimeoutInSec` и `MinFileSizeInBytes` не подходят для каких-либо типов файлов, включите эти параметры в секции для нужных типов файлов с другими значениями

Пример

Время работы экстрактора можно вычислить по формуле:

Время работы экстрактора = Размер файла в Кб / 1024 / (`SpeedInKBs` * Тактовая_частота процессора в МГц / 1000)

Остальные секции содержат сигнатуры, при помощи которых детектируются типы файлов, находящиеся во вложениях и архивах. Секции имеют следующий набор параметров:

Параметр	Описание и примеры настройки
`Extension`	Расширение файла архива. Необязательный параметр, необходим для корректной работы некоторых архиваторов, которые требуют наличия правильного расширения входных файлов **Пример** `Ext = arj`
`Name`	Имя формата. Используется для точного определения имени формата при распаковке файла. Для определения/уточнения формата может включать также список имен: `"defaultname, name1(code1), name2(code2)"` В этом случае формат определяется на основании кода (*codeN*), присвоенного файлу при распаковке. Например, если распаковка завершается с кодом *code1*, то считается, что файл имеет формат *name1*. Список имен может также включать имя по умолчанию (*defaultname*), которое будет присваиваться в случаях,

	когда код не присвоен:
	Пример 1 (одно имя формата)
	`text/xml`
	
	Пример 2 (список имен формата)
	`"application/arj, UNKNOWN(9), text/encrypted(4)"`
	По умолчанию формат файла определяется как application/arj. Если формат архива не удается определить, ему присваивается имя `UNKNOWN.` Для зашифрованного архива используется имя формата `text/encrypted`

`Signature`	Сигнатура формата, которая представляет собой «образцовую» последовательность значений (байтов) с начальным смещением и масками. Эта последовательность необходима для однозначного определения формата файлов по их содержимому
	Сигнатура может включать до 16 триад вида:
	`[OFFSET]:BYTE_VALUE/[MASK]`
	Допускается также и строковая сигнатура, например, "BZh", для BZIP2. Эта сигнатура означает указанную последовательность байтов от смещения 0. Поиск строковых сигнатур выполняется без учета регистра символов.
	Параметр OFFSET является необязательным. Для первого байта этот параметр по умолчанию равен 0, для всех последующих – увеличивается на единицу. Смещение можно задать с символом «?» – поиск образца в первых N байтах файла. Например, для поиска образца в первых 500 байтах файла, нужно задать смещение '`500`?' .
	Параметр MASK, также является необязательным и равен `0xFF` по умолчанию.
	Пример (сигнатура заголовка WAV-формата):
	`Sign = "'RIFF', 0x8: 'WAVEfmt'"`
	Начиная от смещения 0, расположена последовательность 'R', 'I', 'F', 'F', далее от смещения 8 должно следовать 'W', 'A', 'V', 'E', 'f', 'm', 't'

`Comment`	Комментарий с пояснениями. Например, указание используемой версии архиватора: `Comment = "bzip2 v. 1.0.x"`
`Command`	Строка команды для извлечения файлов. Например, `Extract = "/usr/bin/bzip2 -f -dc ${SRC} > ${OUTDIR}/${SRC}"` где `${OUTDIR}` – каталог, в который будет распакован архив, а `${SRC}` – имя файла, который нужно распаковать
`ExcludeFromContext`	Если параметр включен (`On`), то распаковка объекта выполняется, но информация об архиваторе не добавляется в XML-контекст объекта. Это позволяет не отображать в контексте объекта фиктивные контейнеры типа MS-TNEF. По умолчанию параметр отключен (`Off`)
`CaseInsensitive`	Учет регистра символов при обработке файла. Значение по умолчанию – `false`

`CreateTags`	Извлечение метаинформации из файла. Значение по умолчанию – `true`
`Archive`	Регламентирует вид извлеченной информации. При значении `true` - файлы. При значении `false` - данные.
`PIRegexp`	Содержит описание сигнатуры PIRE для детектирования формата файла. Может быть указан несколько раз в пределах одного Extractor. Имеет атрибуты: • assurance - достоверность определения формата файла этой сигнатурой, • offset - смещение, на котором нужно начинать поиск этой сигнатуры, • max_length - длина проверяемого участка, • case_sensitive - включение значимости регистра букв в описании сигнатуры.
`DetectableByRegexp`	Может использоваться для отключения детектирования формата с помощью PIRE (для него будут использоваться сигнатуры из Signature).
`OpenFifoRetryTimeout`	Таймаут между попытками открыть fifo очередь
`OpenFifoRetryCount`	Количество попыток открыть fifo очередь
`ResultCharset`	Указывает кодировку извлеченного экстрактором текста
`NeedProcess`	Нужно ли файлы с указанным mime-типом отправлять на анализ.

7.4 Архивирование каталога очереди сообщений

При ошибках в очереди входящих SMTP-писем:

1. Заархивируйте каталог очереди SMTP-писем **queue/smtp** и сохраните его для последующего анализа.
2. Удалите каталог.

7.5 Логирование работы Системы

Для удобства отслеживания работы процессов, подсистема протоколирования имеет шесть уровней:

Название уровня	Описание
`fatal`	Показывает ошибки, которые препятствуют дальнейшей работе процесса

Название уровня	Описание
`error`	Показывает сообщения об ошибках, которые не являются критическими для работоспособности процесса
`warning`	Выводит сведения о потенциально опасных ситуациях
`info`	Общая полезная информация о процессе (старт/стоп, применение конфигураций и т.д.)
`debug`	Выводит информацию, которая чаще всего используется для диагностики работы сервиса (IT, системные администраторы и т.д.)
`trace`	Чаще всего используется для отслеживания кода разработчиками

Настройка уровней протоколирования

> **Важно!**
>
> Следует учитывать, что изменение уровня протоколирования на более подробный может значительно снизить производительность Системы.

Конфигурационные файлы хранятся в директории `/opt/iw/tm5/etc`. По умолчанию все процессы имеют уровень протоколирования `warning`.

Чтобы изменить уровень протоколирования для процессов Traffic Monitor Server:

1. В конфигурационном файле нужного процесса, в секции `Logging` установите для параметра `GlobalLevel` необходимое значение.
2. Сохраните измененный файл.
3. После внесения изменений в системный журнал, перезапустите процесс:
   `iwtm restart <имя_службы>`

Все логи процессов по умолчанию хранятся в каталоге `/var/log/infowatch`.

Для экстракторов, у которых нет отдельного конфигурационного файла, настройте уровень протоколирования в конфигурационном файле **/opt/iw/tm5/etc/config-perm/bookworm/extractors.xml**:

1. Откройте конфигурационный файл;
2. Для параметра `text-extractor` добавьте значение `-l <level>,` где <level> - уровень протоколирования. Например:
   `< Command> bin/text-extractor -l error -t ooxml -i ${OUTDIR}" < Command>`
3. После внесения изменений, перезапустите процессы:
   `iwtm restart bookworm`
   `iwtm restart warpd`

> **Примечание:**

Данный тип настройки протоколирования доступен для экстракторов следующих форматов: docx, html, msoffice_xml, msole, odp, odt, pptx, xlsx, xml.

7.6 Файловые очереди

Чтобы обеспечить загрузку имеющихся очередей объектов:

1. Остановите службы iw_icap, iw_deliver, iw_luaengined, iw_is, iw_smtpd, iw_sniffer, iw_xapi_xapi, iw_xapi_puppy, iw_analysis, iw_messed, iw_proxy_http, iw_proxy_icq, iw_proxy_smtp, iw_capstack.
2. Переместите объекты из очереди ошибок в нужную очередь.
3. Дождитесь, пока будут обработаны все объекты в файловой очереди.
4. Запустите службы iw_icap, iw_deliver, iw_luaengined, iw_is, iw_smtpd, iw_sniffer, iw_xapi_xapi, iw_xapi_puppy, iw_analysis, iw_messed, iw_proxy_http, iw_proxy_icq, iw_proxy_smtp, iw_capstack.

Чтобы запустить все службы, выполните команду: `iwtm start.` По окончании процесса убедитесь, что службы запущены: `iwtm status`
Варианты команд для запуска/остановки отдельных служб:

Запуск службы	Остановка службы
`iwtm start icap`	`iwtm stop icap`
`iwtm start deliver`	`iwtm stop deliver`
`iwtm start luaengined`	`iwtm stop luaengined`
`iwtm start is`	`iwtm stop is`
`iwtm start smtpd`	`iwtm stop smtpd`
`iwtm start sniffer`	`iwtm stop sniffer`
`iwtm start xapi_xapi`	`iwtm stop xapi_xapi`
`iwtm start xapi_puppy`	`iwtm stop xapi_puppy`
`iwtm start analysis`	`iwtm stop analysis`
`iwtm start messed`	`iwtm stop messed`
`iwtm start proxy_http`	`iwtm stop proxy_http`
`iwtm start proxy_icq`	`iwtm stop proxy_icq`
`iwtm start proxy_smtp`	`iwtm stop proxy_smtp`
`iwtm start capstack`	`iwtm stop capstack`

Объекты в Системе перемещаются посредством файловых очередей:

Путь к директории очереди	Формат файлов в очереди
queue/analysis	.xml & .dat
queue/smtp	.xml & .dat
queue/db	.xml & .dat
queue/blackboard	.dat
queue/blackboard_errors	.dat
queue/is	.xml & .dat
queue/x2x	.xml & .dat
queue/errors	.xml & .dat
queue/x2x-errors	.xml & .dat
queue/x2db-errors	.xml & .dat
queue/final-errors	.xml & .dat

Каждая из очередей содержит дополнительные технические очереди для отслеживания процесса обработки объектов: `.db, .in, .out.`

Пример:

Для очереди `queue/smtp` процесс обработки разделен на следующие этапы:

1. `queue/smtp/.in` – файл формируется в данной очереди в процессе получения объектов от Postfix службой **iw_smtpd**. Если объект по какой-то причине задерживается в этой очереди, необходимо проверить службы **iw_smtpd** или **iw_proxy_smtp**.
2. `queue/smtp/.db` – по окончании обработки файл перемещается из очереди `.in` в очередь `.db`. В эту очередь попадают почтовые eml-объекты от **iw_xapi**
3. `queue/smtp/.out` – в эту очередь объект перемещается следующей службой в цепочке - **iw_messed**.

Особенность:

Очереди, обслуживающие службу **iw_is**, дополнительно разделяются на:

- `queue/is/cmd/` – файловая очередь команд **iw_is**;

- `queue/is/fetching/` – очередь команд на загрузку;
- `queue/is/indexing/` – очередь команд на индексацию;
- `queue/is/errors/` – очередь ошибок.

Очередь	Службы, которые помещают события в очередь	Служба, которая забирает события из очереди
`queue/analysis`	iw_xapi_xapi, iw_xapi_puppy	iw_analysis
`queue/smtp`	iw_xapi_xapi, iw_xapi_puppy, iw_smtpd, iw_proxy_smtp, iw_capstack	iw_messed
`queue/db`	iw_messed, iw_analysis, iw_icap, iw_proxy_icq, iw_proxy_http	iw_x2x
`queue/blackboard`	iw_x2db, iw_deliver	iw_blackboard
`queue/blackboard_errors`	ошибка при обработке iw_blackboard	-
`queue/x2x`	iw_x2x	iw_x2db
`queue/is`	iw_metainfo_fetcher	iw_is
`queue/errors`	все службы, если произошла ошибка обработки	iw_rammer
`queue/x2x-errors`	ошибки обработки iw_x2x	
`queue/x2db-errors`	ошибки обработки iw_x2db	
`queue/final-errors`	iw_rammer	-

Сбор статистики по файловым очередям включается в конфигурационном файле службы в директории `/opt/iw/tm5/etc`. Установите значение `true` в параметре `Enabled` секции `Statistic.`

7.7 Восстановление работоспособности системы в аварийных ситуациях

При серьезных сбоях в работе серверов или сети восстановить работоспособность Системы можно, выполнив следующие действия:

1. Остановите процессы Traffic Monitor, выполнив команду:
   `iwtm stop`
2. Переместите объекты из очередей ошибок в обычные. Процедура выполняется при помощи утилиты iw_qtool (см. статью базы знаний "Как переместить объекты между очередями при помощи утилиты iw_qtool").
3. Если Traffic Monitor Server работает с установленной СУБД Postgre, то для проверки соединения выполните команды:
   `su - iwtm`
   `psql -p 5433 -h <IP-адрес_сервера_БД> postgres iwtm`
   `psql -p 5433 -h 127.0.0.1 postgres iwtm` - для подключения к локальной БД
   Если проверка пройдена успешно, то в ответ на выполнение указанной команды будет выведено приглашение psql:
   `postgres=#`
4. Запустите процессы Traffic Monitor, выполнив команду:
   `iwtm start`

После запуска Traffic Monitor, проверьте системный журнал на наличие ошибок. Путь к файлу журнала: `/var/log/messages.` Если в системном журнале содержится информация об ошибках, обратитесь в службу технической поддержки.

7.8 Управление языками с поддержкой морфологии

Языки с поддержкой морфологии в Traffic Monitor настраиваются посредством изменения конфигурационного файла и установки пакета со словарем языка.

В зависимости от настроек, заданных при установке, в Системе могут быть установлены два и более языков (см. "InfoWatch Traffic Monitor. Руководство по установке"). Обязательно устанавливаются русский и английский словари морфологии - даже в том случае, если это явно не указывалось при установке.

Сведения по доступным действиям приведены в статьях:

- Добавление нового языка для поиска событий. Морфология и добавление терминов.;
- Обновление установленного языка;
- Удаление языка для поиска и терминов.

7.8.1 Добавление нового языка для поиска событий. Морфология и добавление терминов.

Чтобы добавить в Систему новый язык для поиска событий:

1. Остановите процессы **iw_indexer** и **iw_is**:
   `iwtm stop indexer`
   `iwtm stop is`

2. Перейдите в локальный репозиторий, где лежат все пакеты, загруженные при установке:
   `cd /opt/iw/distr`
3. Перейдите в директорию, соответствующую текущей версии Traffic Monitor
4. Выберите необходимый пакет и выполните его установку с помощью команды:
   `rpm -i <название пакета>`
   Например, для установки пакета со словарем французского языка необходимо ввести команду:
   `rpm -i iwtm-sphinx-dict-fra-7.0.0-792.x86-64.rpm`
5. Запустите процессы:
   `iwtm start indexer`
   `iwtm start is`

> **Примечание:**
>
> При установке нового словаря происходит полная переиндексация БД, что занимает некоторое время.

Чтобы включить морфологию языка для поиска событий:

1. Остановите процессы **iw_indexer** и **iw_is**:
   `iwtm stop indexer`

   `iwtm stop is`
2. Откройте файл sphinx_options.ini с помощью команды:
   `mcedit /opt/iw/tm5/etc/sphinx_options.ini`
   и добавьте ключ необходимого языка в параметр `sphinx_language`.
   Например, чтобы включить морфологию русского и турецкого языков в Системе, необходимо указать следующие значения параметра:
   ```
   sphinx_languages = 'rus tur'
   ```

> **Примечание:**
>
> Чтобы выставить приоритет языка, установите параметр с языком последним в списке значений `sphinx_languages` файла **sphinx_options.ini**.

3. Для добавления терминов откройте файл **database.conf** используемой СУБД PostgreSQL:
   `mcedit /opt/iw/tm5/csw/postgres/database.conf`
4. Добавьте в данный файл ключ языка с поддержкой морфологии из параметра `cfdb_language`, например:
   ```
   \set cfdb_language 'rus'
   ```

> **Примечание:**
>
> Полный список ключей для морфологии разных языков приведен в статье "Список языков с поддержкой морфологии".

5. Запустите процессы **iw_indexer** и **iw_is**:
   ```
   iwtm start indexer
   iwtm start is
   ```

7.8.2 Обновление установленного языка

Чтобы обновить существующий в Системе словарь:

1. Перейдите в каталог с пакетами словарей:
   ```
   cd /opt/iw/disrt/7.0.0/
   ```
2. Выполните установку пакета:
   ```
   rpm -i <название пакета>
   ```
   Например:
   ```
   rpm -i iwtm-sphinx_dict-deu-7.0.0.792.x86_64.rpm
   ```

> **Примечание:**
>
> При обновлении словаря повторная индексация БД производиться не будет: новые языки с поддержкой морфологии будут использоваться только для новых событий.

7.8.3 Удаление языка для поиска и терминов

Чтобы удалить словарь из Системы:

1. Остановите процессы **iw_indexer** и **iw_is**:
   ```
   iwtm stop indexer
   iwtm stop is
   ```
2. Откройте файл **database.conf** используемой СУБД PostgreSQL:
   ```
   mcedit /opt/iw/tm5/csw/postgresql/database.conf
   ```
3. Удалите код языка с поддержкой морфологии из параметра `define cfdb_language`.
4. Удалите файлы индексации из директории `/var/lib/sphinx/`
5. Запустите процессы:
   ```
   iwtm start indexer
   iwtm stop is
   ```

После удаления словаря будет произведена полная переиндексация БД.

> **Примечание:**
>
> Чтобы сохранить язык БКФ, добавьте код языка с поддержкой морфологии в параметр `cfdb_language` конфигурационного файла **database.conf** непосредственно перед обновлением Системы. При этом язык с поддержкой морфологии также будет восстановлен.
>
> Чтобы удалить язык с поддержкой морфологии после обновления Системы, повторно пройдите шаги, указанные в данном разделе.

7.9 Настройка передачи информации в SIEM

Traffic Monitor может интегрироваться с SIEM-системами (ArcSight, Tivoli и др.). Под интеграцией подразумевается поступление в SIEM-систему консолидированной информации со всех установленных в компании компонентов системы TM.

Для интеграции используется скрипт **/opt/iw/tm5/bin/config/iwtm-siem.conf.py**, доступный на сервере Traffic Monitor.

Информация из TM доступна для SIEM системы:

1. Посредством табличного представления:

- События, зарегистрированные в TM;
- Аудит сессий пользователей консоли TM;

- Посредством rsyslog:

- Вход/выход пользователей Linux-сервера TM;
- События, зафиксированные в системном журнале Linux-сервера TM, включая информацию о входе/выходе пользователей Базы Данных PostgreSQL;
- Состояние служб Linux-сервера TM или группы серверов.

В этом разделе:

- Настройки на стороне SIEM
- Настройки на стороне TM
- Типы логов, передаваемых в SIEM

7.9.1 Настройки на стороне SIEM

Чтобы подготовить SIEM к получению данных от TM:

1. В SIEM укажите таблицы, из которых необходимо забирать информацию:

- `IWTM.ARC_VIEW_OBJECTS2` - события TM;
- `IWTM.ARC_VIEW_AUDIT_LOG –` аудит пользователей TM
- Создайте учетную запись SIEM для доступа к таблицам БД.
- Настройте обработку данных, извлеченных из БД TM.

Информацию о табличных представлениях (иногда требуется для анализа) см. в статьях:

- "Табличное представление событий TM";
- "Табличное представление аудита пользователей".

> **Важно!**
>
> При настройке интеграции могут понадобиться дополнительные модули от производителя SIEM системы, позволяющие SIEM системе работать с табличными представлениями. Например, для интеграции с HP ArcSight необходим модуль HP Flex Conector.

Табличное представление событий TM

При подключении к БД TM используйте созданного пользователя **siem** (см. "Создание пользователя siem"). Искомые данные содержатся в таблице `IWTM.ARC_VIEW_OBJECTS2`.

> **Примечание:**
>
> Если для импорта в SIEM требуется отфильтровать данные, вы можете выполнить фильтрацию по полю `capture_date` или `insert_date`. Мы рекомендуем выполнять фильтрацию по полю `insert_date`. Фильтрация по полю `object_id` не поддерживается.

Наименование атрибута	Тип данных	Описание	Возможные значения	Пример заполнения
`object_id`	number(20)	Атрибут события *Идентификатор события*	ID события в БД TM. Всегда присутствует.	110
`monitorcode`	varchar2(4000)	Атрибут события *Тип события*	Тип события определяет способ передачи данных. Может принимать одно из следующих значений: • Съемные устройства; • Печать; • ICQ; • Skype; • XMPP; • Mail.Ru Агент; • Telegram; • Vkontakte; • Facebook; • WhatsApp; • MS Lync; • FTP; • Email; • Web-почта; • Web-сообщение; • Буфер обмена; • Облачные хранилища;	Email

Наименование атрибута	Тип данных	Описание	Возможные значения	Пример заполнения
			• Ввод с клавиатуры; • Терминальная сессия; • Сетевые ресурсы; • Чтение файла; • Запись файла.	
protocol		Атрибут события *Протокол*	Протокол может принимать одно из следующих значений: • OSCAR • Skype • XMPP • MMP • SIP • FTP • POP3 • IMAP • MAPI • SMTP • HTTP • HTTPS • NRPC • WhatsApp • Отсутствует	
verdict	varchar2(12)	Атрибут события *Вердикт*	Возможны значения: • *Quarantined* (*Карантин*) • *Forbidden* (*Заблокировано*) • *Allowed* (*Пропущено*)	Forbidden
date_of_capture	varchar2(50)	Атрибут события *Дата перехвата* в		2014-06-19T18:56:26+04:00

Наименование атрибута	Тип данных	Описание	Возможные значения	Пример заполнения
		ISO 8601 с указанием часового пояса даты перехвата		
`capture_data`	timestamp	Дата перехвата события (в UTC, без указания часового пояса) в формате `timestamp`. Используется для быстрой фильтрации данных		timestamp
`insert_date`	timestamp	Дата вставки события (в UTC, без указания часового пояса) в формате `timestamp`		timestamp
`device`	varchar2(256)	Атрибут события *Имя устройства*	Наименование устройства, с которого или на которое происходило копирование. Либо наименование принтера, на который был отправлен на печать документ из перехваченного события. Может быть не заполнено.	Kingston USB Drive 5.0

Наименование атрибута	Тип данных	Описание	Возможные значения	Пример заполнения
`websource`	varchar2(256)	Атрибут события *Ресурс*	Адрес посещенного веб-ресурса или адрес облачного хранилища. Может быть не заполнено.	Yahoo.com
`recipientscontacts`	clob	Атрибут события *Получатели*	Список контактов получателей через запятую. Контакты указанны в формате `<тип контакта>:<значение контакта>`. Может быть не заполнено.	email:ivanov@infowatch.ru, email:petr.petrov@infowatch.ru
`recipientsfullname`	clob	Атрибут события *Получатели*	Список полных имен получателей через запятую. Указаны в том же порядке, что и в `recipients`. Указаны только в том случае, если получателей удалось проидентифицировать. Поле может быть не заполнено.	Ivanov Ivan Ivanovich, Petrov Petr Petrovich
`recipientsdomainaccountname`	clob	Атрибут события *Получатели*	Список ключей идентификации типа `auth` всех получателей через запятую. Указаны в том же порядке, что и в `recipients`. Может быть не заполнено.	Ivanov@iw, Petrov@iw

Наименование атрибута	Тип данных	Описание	Возможные значения	Пример заполнения
`senderdomainaccountname`	clob	Атрибут события *Отправители*	Ключ идентификации типа `auth` отправителя. Поле может быть не заполнено.	Petrov@iw
`senderfullname`	clob	Атрибут события *Отправители*	Полное имя отправителя. Поле может быть не заполнено.	Petrov Petr Petrovich
`sendercontacts`	clob	Атрибут события *Отправители*	Контакт отправителя, указанный в формате `<тип контакта>:<значение контакта>`. Может быть не заполнено.	email:petr.petrov@infowatch.ru
`sendermachinedomainname`	clob	Атрибут события *Рабочая станция*	Доменное имя `dnshostname` рабочей станции отправителя. Может быть не заполнено.	XP-PETROV
`sendermachineip`	clob	Атрибут события *Рабочая станция*	IP-адрес рабочей станции отправителя. Может быть не заполнено.	10.60.20.184
`perimetersin`	clob	Наименование периметра, в который вошло событие	Список наименований периметров, в которые вошло событие. Периметры в списке указаны через запятую. Может быть не заполнено.	Периметр компании

Наименование атрибута	Тип данных	Описание	Возможные значения	Пример заполнения
`perimetersout`	clob	Наименование периметра, из которого ушло событие	Список наименований периметров, из которых ушло событие. Периметры в списке указаны через запятую. Может быть не заполнено.	Периметр компании
`tags`	clob	Атрибут события *Тег*	Список тегов через запятую. Может быть не заполнено.	New
`categories`	clob	Атрибут события *Категория*	Список категорий через запятую. Может быть не заполнено.	Confidentially
`text_objects`	clob	Атрибут события *Текстовый объект*	Список текстовых объектов, обнаруженных в перехваченном событии. Текстовые объекты в списке указаны через запятую. Может быть не заполнено.	special control
`fingerprints`	clob	Атрибуты события *Бланк*, *Эталонный документ*, *Печать*, *Выгрузка из БД*	Список названий форм, эталонных документов , печатей и выгрузок из БД, обнаруженных в перехваченном объекте. Элементы в списке указаны через запятую. Может быть не заполнено.	Бланк заявки.doc

Наименование атрибута	Тип данных	Описание	Возможные значения	Пример заполнения
`protecteddocuments`	clob	Атрибут события *Объект защиты*	Список объектов защиты, обнаруженных в перехваченном объекте. Объекты защиты в списке указаны через запятую. Может быть не заполнено.	Договор аренды
`policies`	clob	Атрибут события *Политика*	Список названий политик, сработавших на перехваченном объекте. Политики в списке указаны через запятую. Может быть не заполнено.	Политика контроля новых сотрудников
`violationtype`	varchar2(40)	Атрибут события *Группа правил*	Может принимать следующие значения: ▪ *Copy* (*Нарушение копирования*) ▪ *Placement* (*Нарушение хранения*) ▪ *Transfer* (*Нарушение передачи*) Может быть не заполнено.	Copy
`violation_level`	Список	Атрибут события *Уровень нарушения*	Может принимать следующие значения: ▪ *High* (*Высокий*) ▪ *Medium* (*Средний*) ▪ *Low* (*Низкий*) ▪ *No violation* (	Medium

Наименование атрибута	Тип данных	Описание	Возможные значения	Пример заполнения
			Отсутствует) Может быть не заполнено.	
`userdecision`	varchar2(27)	Атрибут события *Решение пользователя*	Решение пользователя консоли. Может принимать следующие значения: ▪ *Violation* (*Нарушение*) ▪ *NoViolation* (*Нет нарушения*) ▪ *NotProcessed* (*Решение не принято*) ▪ *AdditionalProcessingNeeded* (*Требует дополнительной обработки*) Может быть не заполнено.	Violation
`filepath`	clob	Атрибут события *Путь к файлу*	Список путей к файлам. Пути в списке указаны через запятую. Может быть не заполнено.	\\xp-petrov\C$\666.txt
`attachments`	clob	Атрибут события *Имя файла вложения*	Имена файлов вложений, указанные через запятую.	666.txt, Petrov.doc, 3.jpg
`url`	varchar2(4000)	Атрибут события *URL*	Полный адрес, с которого осуществлялась передача данных.	10.60.21.34/3.jpg

Наименование атрибута	Тип данных	Описание	Возможные значения	Пример заполнения
			Может быть не заполнено.	
`capture_server_ip`	varchar2(256)	Атрибут события *Сервер перехвата IP*	IP-адрес сервера, на котором были перехвачены данные. Может быть не заполнено.	10.60.21.34
`capture_server_hostname`	varchar2(256)	Атрибут события *Имя сервера перехвата*	Имя сервера, на котором были перехвачены данные. Может быть не заполнено.	iwtm.infowatch.ru

Важно!

Количество выводимых элементов в одной ячейке одной записи ограничено 1000 штук (Например: максимум 1000 получателей или извлеченных файлов в событии).

Табличное представление аудита пользователей

При подключении к БД TM используйте созданного пользователя **siem** (см. "Создание пользователя siem"). Искомые данные содержатся в таблице `IWTM.ARC_VIEW_AUDIT_LOG`.

Атрибут	Тип данных	Описание	Пример заполнения
`audit_log_id`	Number(20)	ID записи в аудите сессий пользователей консоли TM.	12345
`change_date`	Varchar2(50)	Дата и время зарегистрированного действия пользователя	05.08.2015 09:35:10.641446000
`user_login`	varchar2(256)	Логин пользователя, осуществившего действие.	Admin
`user_fullname`	varchar2(256)	Полное имя пользователя,	Petrov Petr Petrovich

Атрибут	Тип данных	Описание	Пример заполнения
		совершившего действие.	
`user_email`	varchar2(256)	E-mail пользователя, совершившего действие.	petr.petrov@infowatch.ru
`operation`	varchar2(40)	Тип действия, которое было произведено пользователем. Возможны значения: • restart (перезапуск) • delete_hash (удаление хэша) • sync (синхронизировать) • add_tag (добавить тег) • remove_tag (удалить тег) • run (выполнение запроса) • start (запуск) • stop (остановка) • view (просмотр) • create (создание) • update (редактирование) • delete (удаление) • login_failure (неуспешная попытка входа) • login (успешный вход) • logout (выход) • change_password (изменение пароля) • decision_update (изменение пользовательского решения)	Edit

Атрибут	Тип данных	Описание	Пример заполнения
		• remove_tag (изменение тегов события) • delete_ref (удаление ссылки) • copy (копирование) • move (перемещение) • commit (применение изменений в конфигурации системы) • rollback (откат изменений в конфигурации системы) • draft (сохранение изменений в конфигурации системы) • add (добавление) • import (импорт) • export (экспорт)	
`entity_type`	varchar2(40)	Тип объекта, над которым осуществлялось действие. Возможны значения: • AgentJob (диагностические данные) • Adlibitum (адлибитум) • Agent (служба) • Classifier (классификатор) • NetworkSettings (сетевые параметры) • NotificationSettings (состояние системы)	Dashboard

Атрибут	Тип данных	Описание	Пример заполнения
		• ObjectReport (выгрузка событий) • Query (запуск поиска событий) • QueryReportRun (агрегация отчета) • Setting (настройки) • UpdateSystem (обновление системы) • Category (категория) • Dashboard (дашборд) • DashboardWidget (виджет) • EtForm (эталонные формы) • EtStamp (эталонные печати) • EtTable (эталонные выгрузки) • Fingerprint (эталонные документы) • LdapContact (LDAP контакт) • LdapGroup (LDAP группа) • LdapPerson (LDAP персона) • LdapStatus (LDAP статус персоны) • LdapWorkstation (LDAP рабочая станция) • Perimeter (периметр) • Policy (политика) • ProtectedDocuments (объект защиты)	

Атрибут	Тип данных	Описание	Пример заполнения
		• Report (отчет) • Role (роль) • Selection (запрос) • ServiceLog (лог сервиса) • SystemList (список тематик ресурсов) • SystemListItem (список ресурсов заданной тематики) • Tag (тег) • Term (термин) • TextObject (текстовый объект) • User (пользователь) • VisibilityArea (область видимости) • Config (Конфигурация) • License (Лицензия) • Object (Событие) • ProtectedCatalog (Каталог Объекта Защиты)	
`entity_display_name`	varchar2(4000)	Наименование объекта, над которым осуществлялось действие.	Statistics1
`property_changes`	clob	Описание произошедших изменений в формате json. Данное поле заполняется только в случае событий управления пользователями, ролями, областями видимости и при осуществлении входа в консоль управления. Возможны три формата заполнения поля.	
	-	**Формат №1.** Актуален только для событий управления ролями и областями видимости.	

Атрибут	Тип данных	Описание	Пример заполнения

		{
		“old”: {
		“<ТИП ОБЪЕКТА>”:{
		{
		“<ПОЛЕ>”: “<ЗНАЧЕНИЕ>”,
		“<ПОЛЕ>”: “<ЗНАЧЕНИЕ>”
		}
		}
		}
		{
		“new”: {
		“<ТИП ОБЪЕКТА>”:{
		{
		“<ПОЛЕ>”: “<ЗНАЧЕНИЕ>”,
		“<ПОЛЕ>”: “<ЗНАЧЕНИЕ>”
		}
		}
		}
		}
			{
			"old":{
			"visibilityareas":{
			}
			},
			"new":{
			"visibilityareas":[
			{
			"VISIBILITY_AREA_ID":"F00207A1E7E7743EE0433D003C0A5DD400000000",
			"DISPLAY_NAME":"<idclip>",
			"NOTE":"<idclip>",
			"VISIBILITY_AREA_CONDITION":"{"data":{"link_operator":"and","children":[]}}",
			"IS_SYSTEM":1
			}
			}
			}

	-	**Формат №2.** Актуален только для событий управления пользователями.	

		{
		“old”:{
		“<ПОЛЕ>”: “<ЗНАЧЕНИЕ>”,
		“<ПОЛЕ>”: “<ЗНАЧЕНИЕ>”
			{
			"old":{
			"EMAIL":"asdasd@asdasd.ru",
			"CHANGE_DATE":"01-07-2014 09:46:29.000000"

Атрибут	Тип данных	Описание	Пример заполнения
		} “new”:{ “<ПОЛЕ>”: “<ЗНАЧЕНИЕ>”, “<ПОЛЕ>”: “<ЗНАЧЕНИЕ>” } }	}, "new":{ "EMAIL":"asdasd11@asdasd.ru", "CHANGE_DATE":"01-07-2014 09:46:44.000000" } }
	-	**Формат №3.** Актуален только для событий входа пользователя в систему. -	
		{ "request":{ "hostname": “<ИМЯ ХОСТА>”, "ip":"<IP-АДРЕС>", "login":"<ЛОГИН ПОЛЬЗОВАТЕЛЯ>" } }	{ "request":{ "hostname":*null*, "ip":"127.0.0.1", "login":"officer" } }

7.9.2 Настройки на стороне TM

Чтобы настроить передачу консолидированной информации из TM:

1. Создайте учетную запись БД для SIEM (см. "Создание пользователя siem").
2. Настройте передачу информации из TM в SIEM (см. "Передача логов в SIEM");
3. При необходимости измените настройки логирования аудита сессий пользователей БД TM (см. "Управление логированием сессий пользователей БД TM").

Управление пользователем siem

В данном разделе описывается создание, удаление и смена пароля для пользователя **siem**, от имени которого SIEM будет взаимодействовать с TM:

- Создание пользователя siem;
- Смена пароля пользователя siem;

- Удаление пользователя siem.

Создание пользователя siem

Чтобы создать пользователя siem:

1. Для запуска скрипта выполните команду:
   `/opt/iw/tm5/bin/config/iwtm-siem.conf.py`
2. Выберите опцию **DB siem user**.
3. В появившемся меню выберите опцию **Create DB siem user**.
4. Задайте параметры:

Параметр	Описание
`Enter login for DB administrative account`	Логин для учетной записи администратора PostgreSQL **Примечание:** используются логин `postgres`
`Enter password for DB administrative account`	Пароль для учетной записи администратора PostgreSQL
`Enter password DB siem user`	Пароль для новой учетной записи siem

5. Нажмите **Create**.

Смена пароля пользователя siem

Чтобы сменить пароль пользователю siem:

1. Для запуска скрипта выполните команду:
   `/opt/iw/tm5/bin/config/iwtm-siem.conf.py`
2. Выберите опцию **DB siem user** .
3. Выберите опцию **Change password for DB siem user**.
4. Задайте параметры:

Параметр	Описание
`Enter login for DB administrative account`	Логин для учетной записи администратора PostgreSQL **Примечание:** используются логин `postgres`
`Enter password for DB administrative account`	Пароль для учетной записи администратора PostgreSQL
`Enter new password user siem`	Новый пароль для учетной записи siem

5. Нажмите **Change password**.

Удаление пользователя siem

Чтобы удалить пользователя siem:

1. Для запуска скрипта выполните команду: `/opt/iw/tm5/bin/config/iwtm-siem.conf.py`
2. Выберите опцию **DB siem user** .
3. Выберите опцию **Delete DB siem user**.
4. Задайте параметры:

Параметр	Значение
`Enter login for DB administrative account`	Логин для учетной записи администратора PostgreSQL
`Enter password for DB administrative account`	Пароль для учетной записи администратора PostgreSQL

5. Нажмите **Delete**.

Передача логов в SIEM

Чтобы настроить передачу записей из лог файлов Linux-сервера TM в SIEM:

1. Запустите скрипт: `/opt/iw/tm5/bin/config/iwtm-siem.conf.py`
2. Выберите опцию **Log messages forwarding configuration**.
3. Задайте в параметры передачи логов:

Параметр	Описание
`Enable forwarding`	Включение/выключение пересылки логов в siem. (Возможные значения: `Yes/No`)
`Forwarding server`	IP-адрес или dns-имя сервера SIEM
`Forwarding server port`	Порт сервера SIEM, на котором работает **syslog**
`Forwarding protocol`	Протокол передачи данных в SIEM. (Возможные значения: `TCP/UDP`)
`Log messages severity`	Минимальный уровень лог-сообщений, пересылаемых на сервер SIEM. (Возможные значения: `Emergency, Alert, Critical, Error, Warning, Notice, Info, Debug`)
`Size (MB)`	Допустимый размер очереди сообщений, требующих отправку в SIEM. (Значение по умолчанию: `500`)

4. Нажмите **Ok**.

Важно!

Отключение пользователей с правами администратора от БД не логируется, поэтому в SIEM данная информация попадать не будет.

Управление логированием сессий пользователей БД TM

Включение и выключение логирования сессий пользователей БД TM в системном журнале ОС осуществляется через **rsyslog**.

Чтобы включить логирование сессий пользователей Базы Данных TM:

1. Запустите скрипт: `/opt/iw/tm5/bin/config/iwtm-siem.conf.py`
2. Выберите опцию **DB audit logging management**.
3. Введите учетные данные администратора БД:
   - **Enter login for DB administrative account** - логин для учетной записи;
   - **Enter password for DB administrative account** - пароль для учетной записи.
4. Нажмите **Check logging status**.
5. Если:
   - **DB audit logging to OS journal: OFF** - нажмите **Change logging status**, чтобы включить логирование.
   - **DB audit logging to OS journal: ON** - нажмите **Change logging status**, чтобы выключить логирование.

7.9.3 Типы логов, передаваемых в SIEM

Rsyslog формирует общий системный журнал и является протоколом, по которому логи передаются в SIEM.

Существует четыре основных типа логов, которые передаются в SIEM:

1. Логи от процессов **iw_***. Например:
   ```
   Mar 11 14:32:30 iw-VMware-4224da3ab iw_xapi: 5 (18485:0x00007f53001b97e0) [INFO   ] : <Root> Runtime environment initialized. Starting...
   ```
2. Логи от скрипта **pguard**, который отслеживает состояние сервисов и перезапускает их в случае необходимости:
   ```
   Mar 11 14:32:26 iw-VMware-4224da3ab pguard: 148 (18256:0) [INFO] : xapi Terminating pid 18257 (signal 15)
   ```
3. Логи входа и выхода **пользователей Linux**:
   ```
   Mar 11 14:32:02 iw-VMware-4224da3ab runuser: pam_unix(runuser:session): session opened for user iwtm by root(uid=0)
   ```
4. Логи **Nagios**:
   ```
   Feb 10 20:05:39 iw-VMware-4224289a4 nagios: SERVICE ALERT: IWTM;TM_DAEMONS_STATE;CRITICAL;SOFT;1;CRITICAL - updater(3)
   ```

7.10 Удаление временных файлов

Возможность удаления временных файлов реализована в продукте в виде сценария **clean_temporary_files.sh**.

Сценарий удаляет следующую информацию:

1. Файлы из директории временных файлов операционной системы (директория `/opt/iw/tm5/tmp/`).
2. Данные из директорий файловых очередей Traffic Monitor (директория `/opt/iw/tm5/queue/`).

> **Примечание:**
>
> Во время удаления временных файлов обработка поступающих в Систему событий будет остановлена. Данные обо всех необработанных на момент начала удаления событиях будут удалены.

Чтобы удалить временные файлы:

1. Запустите сценарий **clean_temporary_files.sh**:
   `/opt/iw/tm5/bin/clean_temporary_files.sh`
2. Согласитесь на удаление временных файлов:
   `yes`

8 Настройка сквозной аутентификации между продуктами InfoWatch

Сквозная аутентификация позволяет вам переходить между установленными продуктами InfoWatch без повторного ввода логина и пароля. В данной версии InfoWatch Traffic Monitor поддерживается технология единого входа с продуктом InfoWatch Vision версии 2.5 и возможна только при явном переходе из одного продукта в другой и только для пользователей, созданных на основании одинаковой учетной записи в Active Directory.

Чтобы настроить сквозную аутентификацию:

1. В командной строке на сервере Vision введите команду:
   ```
   kubectl get secret guardkeys-central -n infowatch -o 'go-template={{index .data "ec256-public.pem"}}' | base64 -d
   ```
2. Скопируйте выданный системой публичный ключ в pem-формате. Пример ключа:
   ```
   -----BEGIN PUBLIC KEY-----
   MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAExqBY7YcuTlucSX6s8LBTrBmWRuxT
   WkEmnwCBkurgc9Ntm+o+QOOmCZube3M8yimt2Xqy3sgq/WJe+PoGjvhVCg==
   -----END PUBLIC KEY-----
   ```
3. Подключитесь к БД Traffic Monitor:
   ```
   psql postgres iwtm
   ```
4. Укажите в Traffic Monitor список доверенных систем с их открытыми ключами, которые могут авторизоваться в Traffic Monitor, используя свою учетную запись. Для каждой доверенной системы задайте название доверенной системы и открытый ключ доверенной системы. Например, выполните команду:
   ```
   insert into trusted_system(system_name, public_key)
   values('my system','-----BEGIN PUBLIC KEY-----
   MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAExqBY7YcuTlucSX6s8LBTrBmWRuxT
   WkEmnwCBkurgc9Ntm+o+QOOmCZube3M8yimt2Xqy3sgq/WJe+PoGjvhVCg==
   -----END PUBLIC KEY-----');
   commit;
   ```
5. Выйдите из БД:
   ```
   \q
   ```
6. Откройте новую вкладку браузера и войдите в консоль Traffic Monitor.
7. Настройте и запустите синхронизацию с Active Directory для импорта пользователя, который и будет работать в обеих системах. Дождитесь успешного завершения синхронизации.
8. В разделе **Управление доступом** добавьте нового пользователя из LDAP, назначьте ему роль на просмотр событий и область видимости.
9. Выйдите из учетной записи пользователя.
10. Откройте консоль Vision на другой вкладке браузера.
11. Настройте и запустите ту же синхронизацию с Active Directory для импорта того же пользователя.
12. После успешной синхронизации на вкладке **Пользователи** добавьте нового пользователя из LDAP, назначьте ему роль.
13. Выйдите из учетной записи пользователя и войдите под новым созданным пользователем LDAP.
14. Перейдите в раздел **События** , щелчком мыши выберите в списке одно или несколько событий и нажмите **Просмотр события**.

Будет автоматически открыт раздел **События** консоли Traffic Monitor, а также зарегистрирован вход пользователя из внешней системы в журнале аудита.

9 Взаимодействие с внешними системами

Обмен данными между Traffic Monitor и внешними системами осуществляется посредством API. Для удобства работы с ним реализован пакет **iwtm-apidocs-publicapi**, который входит в состав Traffic Monitor и требует установки. Для этого выполните команду на сервере Traffic Monitor: `yum install iwtm-apidocs-publicapi`

При этом будут установлены пакеты:

- **iwtm-apidocs-publicapi**, содержащий OpenAPI-схемы публичного API;
- **iwtm-apidocs-common**, содержащий UI с полным описанием методов, включая модели, коды ответов, параметры запросов. Доступ к нему осуществляется по адресу: <имя_сервера_TM>/doc/

Далее описаны протоколы обмена данными между Traffic Monitor и внешними системами:

- Загрузка событий в InfoWatch Traffic Monitor (pushAPI SDK);
- Загрузка эталонных документов и выгрузок из БД в Traffic Monitor (REST API SDK);
- Получение доступа к событиям Traffic Monitor внешними системами (DataExport API SDK).

9.1 Загрузка событий в InfoWatch Traffic Monitor (pushAPI SDK)

Программный интерфейс pushAPI SDK предназначен для интеграции сторонних компонентов с Traffic Monitor для осуществления перехвата информации с различного рода каналов и их последующего анализа.
Сторонние компоненты самостоятельно формируют события перехвата данных и передают их в Traffic Monitor. После этого Traffic Monitor классифицирует события, анализирует их и выносит свой вердикт на основе настроенных политик защиты данных. Переданные события и вердикты можно просматривать в консоли Traffic Monitor.

9.1.1 Принципы использования pushAPI SDK

Для организации передачи событий на Traffic Monitor необходимо учитывать следующие принципы:

- События перехвата передаются в Traffic Monitor в виде объектов «событие»;
- Каждое «событие» имеет набор обязательных атрибутов, которые задаются пользователем программного интерфейса;
- Каждое «событие» может иметь набор необязательных атрибутов, которые задаются пользователем программного интерфейса;
- Каждое «событие» имеет «отправителя» и «получателя»;
- Каждый «отправитель» и «получатель» может иметь связанный с ним список «контактов», которые позволяют уникальным образом идентифицировать отправителя и получателя;
- Каждый «получатель» и «отправитель» может иметь связанный с ним набор атрибутов;
- Каждое «событие» имеет связанные с ним «данные», т.е. последовательность байт и/или символов текста, которые представляют собой содержимое перехваченного события;
- Каждый элемент «данных» может иметь связанный с ним набор атрибутов.

Все возможные «события», которые может принимать и анализировать Traffic Monitor, разбиты на группы – классы событий, которые объединены общими характеристиками. Класс события определяет бизнес-логику обработки события, а также правила его отображения в консоли Traffic Monitor.
Собственные типы данных, т.е. классы событий, в Traffic Monitor создать нельзя. Пользователь SDK должен выбрать необходимые ему из списка поддерживаемых:

- **Почта** – любые события обмена электронными письмами, включая вложения. Характеризуется списком персон-отправителей и персон-получателей, датой отправки, темой, телом и вложениями к письму. Включает в себя корпоративную почту (IMAP, POP3, SMTP, MAPI, NRPC) и Веб-почту (HTTP, HTTPS). Название сервиса в TM: **Почта**;
- **Беседа** – любые события беседы, представляющие собой обмен текстовыми сообщениями или файлами в рамках одного приложения. Приложение может являться как тонким клиентом на рабочих станциях и мобильных устройствах, так и web-приложением. Характеризуется списком реплик. Для каждой реплики должна быть указана персона-отправитель, время отправления и текст сообщения. Следующие сервисы уже предустановлены в этом классе: MS Lync, ICQ, Mail.ru, Jabber, Skype (передача SMS и обычных текстовых сообщений). Название сервиса в TM: **Мессенджеры**;
- **Печать, копирование и сканирование** – события преобразования данных на специализированных устройствах ввода и вывода изображений/документов (например: Принтер, Сканер). Характеризуется персоной-отправителем и устройством-получателем, количеством копий, а также файлом с оригиналом документа/образом документа/заданием на печать. В рамках этого класса предустановлен сервис Принтер. Название сервиса в TM: **Принтер и МФУ**;
- **Интернет-активность** – события формирования запросов из Web-браузеров и аналогичных программ, осуществляемое по протоколам HTTP, HTTPS. В событие данного типа попадают все данные, для которых не нашлось другого типа. Примером может служить отправка веб-сообщения. Характеризуется персоной-отправителем и ресурсом-получателем в случае POST-запросов или персоной-получателем и ресурсом-отправителем в случае GET-запросов, а также текстом "сырого" запроса. В рамках этого класса предустановлен сервис Интернет-приложение. Название сервиса в TM: **Интернет-активность**;
- **Фотосъемка** – события фотосъемки с помощью камеры устройства. Характеризуется персоной-отправителем и устройством-получателем, а также перехваченным изображением. Название сервиса в TM: **Запись мультимедиа**;
- **Обмен файлами** – любые события, связанные с передачей файлов между произвольными отправителями и получателями. Характеризуется персоной-отправителем и персоной-/устройством-/ресурсом-получателем, а также перехваченным файлом. Включает в себя копирование на USB-устройство, в облачное хранилище, обмен файлами по FTP-протоколу. Следующие сервисы уже предустановлены в этом классе: FTP, Съемное устройство, MS Lync, ICQ, Mail.ru, Jabber, Skype, Облачные хранилища, Терминальная сессия, Сетевые ресурсы. Название сервиса в TM: **Мессенджеры** (если отправители и получатели события являются персонами) или **Обмен файлами** (если хотя бы один отправитель или получатель события не является персоной);
- **Звонок** – любое событие голосового или видеоразговора между людьми, который может происходить по мессенджеру или телефону. Характеризуется персоной-отправителем (инициатор звонка) и персонами-получателями, а также файлом и длиной записи разговора. Примером могут служить отправка/получение SMS и MMS или перехват разговоров в сотовых сетях. Следующие сервисы уже предустановлены в этом классе: MS Lync, Mail.ru, Jabber, Skype. Название сервиса в TM: **Мессенджеры**;
- **Запись мультимедиа** – события создания аудио- и видеозаписи с помощью устройства. Характеризуются персоной-отправителем и устройством-получателем, а также перехваченным изображением. Примером могут служить записи с микрофонов/диктофонов или записи на видеокамеру, а также любые события, связанные с

фотографированием потенциально конфиденциальных документов на камеру устройства. Название сервиса в TM: **Запись мультимедиа**.

Пользователь SDK должен обязательно определить класс событий и «Протокол» для своих перехваченных объектов. «Протокол» определяет, что именно было перехвачено с точки зрения пользовательского сервиса, который использовал «отправитель» для передачи информации. Добавление собственных «Протоколов» осуществляется через файл регистрации стороннего компонента (подробнее см. «Регистрация стороннего компонента pushAPI SDK»). На основании «Протокола» объекта и идентификатора компании-производителя перехватчика реализуются политики лицензирования внутри Traffic Monitor. Также «Протокол» перехваченного события может задаваться как одно из условий в формировании «политики» Traffic Monitor и аналитических запросах.

На основе классов событий пользователь SDK может определить и зарегистрировать свой сервис или выбрать из числа встроенных в Traffic Monitor.

Пример:

В Traffic Monitor есть встроенный сервис «file_copy_removable» для перехвата событий класса «Обмен файлами» при копировании файлов на съемные носители.

Пример:

Если пользователь SDK реализовал перехватчик для мессенджера Viber, он может определить и зарегистрировать сервис "Viber" для перехвата событий, например, таких классов, как «Беседа», «Голосовая беседа», «Обмен файлами». На каждый сервис одного производителя будет выдаваться отдельная лицензия.

У каждого «События» должны быть определены «отправитель(и)» и «получатель(и)». Поддерживаются следующие типы «получателей» и «отправителей»:

- «Персона» – представляет собой «пользователя», который определяется как создатель, отправитель или получатель перехваченного объекта. Для персоны можно определить набор «контактов» для ее идентификации. Тип контакта можно выбрать из списка, который поддерживается SDK. В качестве примера типа контакта можно привести email, телефон или учетную запись Windows;
- «Компьютеры» – представляет собой компьютер, мобильное устройство или терминальный сервер, на котором был «создан» перехваченный объект. Для данного типа «отправителя» можно определить список «контактов»;
- «Ресурс» – представляет собой адрес в сети Интернет, на который были переданы данные;
- «Устройство» – представляет собой устройство, которое получает данные при копировании файлов на съемные носители.

Список контактов, связанных с «Персоной», позволяет идентифицировать ее по любому ее контакту. Таким образом, любые перехваченные действия «Персоны», в которых определен хотя бы один ее контакт, позволяют идентифицировать конкретного пользователя в Traffic Monitor.
Интерфейс SDK позволяет связывать различные типы контактов между собой логическими связями, например, отправителя «Персону» с отправителем «Компьютеры». Это позволяет точно определить контекст перехватываемой операции, т.е. кто и где инициировал действие по передаче данных.
Интерфейс SDK позволяет передавать данные в сеть по частям. Это позволяет внешним разработчикам самим определять стратегию ограничения нагрузки на сеть.
Общий алгоритм передачи события через интерфейс SDK выглядит так:

1. Создать объект «Событие» и установить его атрибуты;
2. Создать отправителей/получателей события, а также установить их контакты и атрибуты;
3. Создать данные события и установить его атрибуты;
4. Создать соединение с Traffic Monitor;
5. Проверить доступ к Traffic Monitor и лицензию плагина внешнего разработчика;
6. Передать «метаданные» события;
7. Передать по частям содержимое данных события;
8. Закрыть соединение;
9. Удалить объект «Событие».

Более подробно процесс передачи событий можно посмотреть в Примерах SDK.

9.1.2 Общее описание программного интерфейса pushAPI SDK

Программный интерфейс реализован в виде Thrift-сервиса. Подробнее о Thrift-сервисах можно прочитать тут: https://thrift.apache.org.

Для работы pushAPI SDK установите последнюю доступную версию Thrift.

Для операционных систем, подобных Red Hat Linux, можно воспользоваться готовым пакетом из стороннего репозитория EPEL.

Установка Thrift на примере операционной системы Red Hat Enterprise Linux 7.x

1. Установите репозиторий EPEL, выполнив в терминале команду:
   `yum install epel-release`
2. Установите Thrift, выполнив команду:
   `yum -y install thrift`

В состав примеров SDK входит Thrift-схема, которая описывает сервис, реализованный в Traffic Monitor. Далее приведено описание этой схемы:

- Типы данных pushAPI SDK
- Исключения pushAPI SDK
- Сервисы Traffic Monitor pushAPI SDK
- Типы контактов Traffic Monitor pushAPI SDK
- Поддерживаемые атрибуты Traffic Monitor pushAPI SDK
- Сервис EventProcessor pushAPI SDK

Типы данных pushAPI SDK

Код	**Описание**
Структура Identity	
`typedef i32 ItemId`	Идентификатор элемента в событии
`enum IdentityItemType { kPerson, kWorkstation, kDevice,`	Вид элемента идентификации

Код	**Описание**
`kResource,` `}`	

`struct Attribute {`
`1: required string name,`
`2: required string value`
`}`
	Отдельный атрибут любой сущности события (самого события, идентификации, потока данных):
	1. контакт;
	2. метаинформация о контакте

`struct ContactWithMeta {`
`1: required Attribute contact,`
`2: optional string meta`
`}`
	Контакты идентификации с метаинформацией:
	1. контакт;
	2. метаинформация о контакте

`struct Identity {`
`1: required ItemId identity_id,`
`2: required IdentityItemType identity_type,`
`3: required list<Attribute> identity_contacts,`
`4: optional list<Attribute> identity_attributes,`
`5: optional list<ContactWithMeta> identity_contacts_with_meta`
`}`
	Идентификация получателя или отправителя объекта:
	1. уникальный номер идентификации внутри объекта, используется для ссылок на идентификацию;
	2. вид элемента идентификации;
	3. набор контактов идентификации;
	4. не обязательный набор атрибутов идентификации;
	5. набор контактов идентификации с метаинформацией

`struct EventData {`
`1: required ItemId data_id,`
`2: optional list<Attribute> data_attributes`
`}`
	Описание потока данных события:
	1. уникальный идентификатор потока данных внутри события, используется для ссылок на поток;
	2. необязательный набор атрибутов потока данных

`struct ChatMessage {`
`1: required ItemId sender_id,`
`2: required string sent_time,`
`3: required string utf8_text,`
`4: optional ItemId mes_data_id`
`}`
	Описание одного сообщения класса "Беседа":
	1. ссылка на идентификацию, которая является отправителем сообщения;
	2. локальное время передачи сообщения в формате ГГГГ-ММ-ДДТЧЧ:ММ:СС+ЧЧ:ММ (ISO 8601 с форматированием yyyy-MM-ddTHH:mm:sszzz);
	3. текст сообщения в кодировке UTF-8;
	4. идентификатор данных объекта, в которых будет передано содержимое сообщения, если utf8_text пустой

Код	**Описание**

`typedef string LinkNameType`
`struct ItemLink {`
`1: required ItemId id,`
`2: required ItemId link_to_id,`
`3: required LinkNameType link_name`
`}`
	Описание связи между различными сущностями события:
	1. идентификатор сущности события, от которой идёт связь;
	2. идентификатор сущности события, к которой идёт связь;
	3. имя (тип) связи

`const LinkNameType link_name_ws = "workstation_link"`	Список имён поддерживаемых связей между сущностями: • имя для задания связи идентификаций типа "Персона" и "Рабочая станция"

`struct Event {`
`1: required EventClass evt_class,`
`2: required TmServiceType evt_service,`
`3: optional list<Attribute> evt_attributes,`
`4: required list<Identity> evt_senders,`
`5: required list<Identity> evt_receivers,`
`6: required list<EventData> evt_data,`
`7: optional list<ChatMessage> evt_messages,`
`8: optional list<ItemLink> evt_links,`
`9: optional Identity evt_source,`
`10: optional Identity evt_destination`
`}`
	Описание события, которое нужно передать в Traffic Monitor:
	1. класс события;
	2. сервис события;
	3. атрибуты события;
	4. отправители события;
	5. получатели события;
	6. данные события;
	7. сообщения события для класса "Беседа";
	8. связи между сущностями в событии;
	9. источник события;
	10. приемник события

`struct Credentials {`
`1: required string company_name,`
`2: required string token`
`}`
	Структура, описывающая параметры доступа к Traffic Monitor:
	1. название клиента (или компании в нашем случае). Этот параметр связывает плагин внешнего разработчика с лицензией;
	2. токен авторизации. Эта строка является «секретом», который знают плагин и Traffic Monitor. Так реализована авторизация источника данных

`typedef i64 EventId`	Идентификатор загружаемого события

Код	**Описание**
`typedef i64 StreamId`	Идентификатор загружаемого потока данных события
`typedef i32 InterfaceVersion` `const InterfaceVersion pushapi_version = 1`	Версия PushAPI интерфейса

`enum EventClass {`
`kChat,`
`kEmail,`
`kMfp,`
`kWeb,`
`kFileExchange,`
`kPhoto,`
`kMultimedia,`
`kVoiceTalk`
`}`
	Поддерживаемые "классы" событий в PushAPI:
	• Беседа
	• Почта
	• Печать, копирование и сканирование
	• Интернет-активность
	• Обмен файлами
	• Фотосъемка
	• Запись мультимедиа
	• Звонок

Примечание:

Если необходимо реализовать сценарий блокировки передачи данных по результатам анализа, то при отправке посредством PushAPI событий в Traffic Monitor используйте один из "классов" событий: `kChat, kEmail , kWeb.` Только для этих классов, соответствующих типам событий **Мессенджер**, **Почта**, **Интернет-активность** в Traffic Monitor реализована возможность назначения событию вердикта **Заблокировано** в политике защиты данных.

Исключения pushAPI SDK

Код	**Описание**

`exception EventNotFound {`
`1: string message,`
`}`
	EventId, указанный в списке параметров, не найден в рамках текущей сессии передачи события

`exception DataNotFound {`
`1: string message,`
`}`
	DataId для указанного EventId некорректный, т.е. указанный DataId не найден в рамках текущей сессии передачи события

`exception StreamNotFound {`
`1: string message,`
`}`
	StreamId для указанного EventId некорректный, т.е. указанный StreamId не найден в рамках текущей сессии передачи события

`exception InvalidEventFormat {`	Некорректный формат события, т.е. не прошла проверка на содержимое атрибутов события. Каждый класс события имеет набор обязательных атрибутов

Код	**Описание**
`1: string message,` `}`	

`exception InvalidCredentials {`
`1: string message`
`}`
	Некорректные параметры доступа к интерфейсу. Нужно указать правильный токен

`exception LicenseError {`
`1: string message`
`}`
	На сервере Traffic Monitor нет правильной лицензии для передачи событий

Примечание: message – уточняет причину возникновения исключения	

Сервисы Traffic Monitor pushAPI SDK

Встроенные в Traffic Monitor сервисы, которые можно использовать для событий:

```
const TmServiceType service_email = "email" //!< Почта на Клиенте
const TmServiceType service_email_web = "email_web" //!< Почта в Браузере
const TmServiceType service_im_icq = "im_icq" //!< ICQ
const TmServiceType service_im_skype = "im_skype" //!< Skype
const TmServiceType service_im_xmpp = "im_xmpp" //!< XMPP
const TmServiceType service_im_mmp = "im_mail_ru" //!< MMP
const TmServiceType service_im_lync = "im_lync" //!< MS Lync
const TmServiceType service_web = "web_common" //!< Web-сообщения
const TmServiceType service_file_removable_storage = "file_copy_removable" //!< Обмен файлами с внешним устройством (съёмным устройством хранения)
const TmServiceType service_file_ftp = "ftp" //!< Обмен файлами по FTP
const TmServiceType service_print = "print" //!< Печать
const TmServiceType service_cloud_storage = "cloud_storage" //!< Обмен файлами с облачным хранилищем
const TmServiceType service_terminal_session = "terminal_session" //!< Обмен файлами через терминальную сессию
const TmServiceType service_network_resource = "network_resource" //!< Обмен файлами с сетевыми ресурсами
```

Наименование сервиса	**Классы**	**Описание**
email	«Электронная почта»	Сервис перехвата корпоративной почты, который не зависит от используемого протокола или типа почтового сервера. При расширении класса «Электронная почта» рекомендуется добавлять к названию сервиса префикс "email_", например "email_google"
email_web	«Электронная почта»	Сервис перехвата почты в браузере

Наименование сервиса	Классы	Описание
im_icq	«Беседа», «Голосовая беседа», «Обмен файлами»	Сервис перехвата программы обмена сообщениями "ICQ" (протокол "OSCAR")
im_skype	«Беседа», «Голосовая беседа», «Обмен файлами»	Сервис перехвата программы обмена сообщениями "Skype"
im_xmpp	«Беседа», «Голосовая беседа», «Обмен файлами»	Сервис перехвата любой программы обмена сообщениями, которая работает по протоколу XMPP (Jaber)
im_mail_ru	«Беседа», «Голосовая беседа», «Обмен файлами»	Сервис перехвата любой программы обмена сообщениями, которая работает по протоколу MMP(Mail.ru Agent)
im_lync	«Беседа», «Голосовая беседа», «Обмен файлами»	Сервис перехвата программы обмена сообщениями "MS Lync"
web_common	«Интернет-активность»	Сервис перехвата неразобранных POST-запросов
file_copy_removable	«Обмен файлами»	Сервис, который создает копии файлов, копируемых на съёмные устройства хранения данных
ftp	«Обмен файлами»	Сервис, который создает копии файлов, передаваемых по FTP-протоколу
print	«Принтеры и МФУ»	Сервис, который создает «образы» распечатываемых на принтере документов
cloud_storage	«Обмен файлами»	Сервис, который создает копии файлов, копируемых на облачные хранилища
terminal_session	«Обмен файлами»	Сервис, который создает копии файлов, передаваемых через терминальную сессию
network_resource	«Обмен файлами»	Сервис, который создает копии файлов, копируемых на сетевые ресурсы
Примечание: При добавлении нового сервиса, который представляет собой программу обмена сообщениями, рекомендуется добавить к его названию префикс "im_". Например "im_viber"		

Примечание:

Сервисы с другими типами событий могут быть зарегистрированы с помощью плагинов

Типы контактов Traffic Monitor pushAPI SDK

Типы контактов, поддерживаемые бизнес-логикой Traffic Monitor:

Код	**Описание**
`const TmContactType contact_type_email = "email"`	Почтовый адрес. Используется для идентификаций с типом «Персона»
`const TmContactType contact_type_auth = "auth"`	Логин в ОС в формате login@domain. Используется для идентификаций с типом «Персона»
`const TmContactType contact_type_dnshostname = "dnshostname"`	DNS имя. Используется для отправителей с типом «Рабочая станция»
`const TmContactType contact_type_icq = "icq"`	Icq-номер. Используется для идентификаций с типом «Персона»
`const TmContactType contact_type_skype = "skype"`	Skype-логин. Используется для идентификаций с типом «Персона»
`const TmContactType contact_type_phone = "phone"`	Номер телефона. Используется для идентификаций с типом «Персона»
`const TmContactType contact_type_ip = "ip"`	IPv4-адрес. Используется для отправителей с типом «Рабочая станция»
`const TmContactType contact_type_sid = "sid"`	Security Identifier в операционной системе Windows, представленный в текстовом формате. Используется для идентификаций с типом «Персона»
`const TmContactType contact_type_webaccount = "webaccount"`	Название учетной записи на сетевом ресурсе
`const TmContactType contact_type_lotus = "lotus"`	Почтовый адрес в Lotus. Используется для идентификаций с типом «Персона»
`const TmContactType contact_type_domain = "domain"`	Домен

Поддерживаемые атрибуты Traffic Monitor pushAPI SDK

Список имён атрибутов, которые поддерживаются в Traffic Monitor:

Код	Описание атрибута
`typedef string EventAttributeType`	**Атрибуты события:**
`const EventAttributeType event_attr_capture_date = "capture_ts"`	Локальное время перехвата события в формате ГГГГ-ММ-ДДТЧЧ:ММ:СС+ЧЧ:ММ (ISO 8601 с форматированием yyyy-MM-ddTHH:mm:sszzz) (*)
`const EventAttributeType event_attr_capture_server_ip = "capture_server_ip"`	IP-адрес сервера перехвата (позволяет различать различные экземпляры одного плагина) (*)
`const EventAttributeType event_attr_capture_server_fqdn = "capture_server_fqdn"`	DNS-имя сервера перехвата (позволяет различать различные экземпляры одного плагина) (*)
`const EventAttributeType event_attr_inteceptor_id = "interceptor_id"`	Идентификатор объекта в перехватчике
`const EventAttributeType event_attr_inteceptor_protocol = "protocol_mnemo"`	Протокол, по которому передавался объект
`const EventAttributeType event_attr_print_copies = "print_copies"`	Количество копий, целое число в строковом представлении
`const EventAttributeType event_attr_destination_type = "destination_type"`	Тип приёмника
`typedef string DataAttributesType`	**Общие атрибуты потока данных:**
`const DataAttributesType data_attr_mimetype = "mimetype"`	MIME-тип содержимого (заполняется только для узлов, которые уже разобраны клиентом)
`const DataAttributesType data_attr_size = "data_size"`	Размер данных, если он известен заранее

Код	Описание атрибута
`const DataAttributesType data_attr_sent_date = "sent_date"`	Дата отправки объекта в формате ГГГГ-ММ-ДДТЧЧ:ММ:СС+ЧЧ:ММ. Позволяет независимо устанавливать время формирования события и время реальной передачи данных
`const DataAttributesType data_attr_comment = "data_comment"`	Комментарий
	Атрибуты потока данных для класса "Электронная почта":
`const DataAttributesType data_attr_email_subject = "subject"`	Тема письма
`const DataAttributesType data_attr_email_is_encrypted = "encrypted"`	Признак шифрования (строки "true" или "false")
`const DataAttributesType data_attr_email_is_signed = "is_signed"`	Признак наличия цифровой подписи (строки "true" или "false")
	Атрибуты потока данных для класса "Обмен файлами":
`const DataAttributesType data_attr_file_filename = "filename"`	Короткое имя файла (*)
`const DataAttributesType data_attr_file_source_file_path = "source_path"`	Полный путь к файлу источнику события
`const DataAttributesType data_attr_file_destination_file_path = "destination_path"`	Полный путь к переданному или скопированному файлу или полный адрес веб-ресурса
	Атрибуты потока данных для класса "Сотовая связь":

Код	Описание атрибута
`const DataAttributesType data_attr_voice_duration_sec = "call_duration"`	Длительность в секундах (целое число в строковом представлении)
	Атрибуты потока данных для класса "Принтер и МФУ":
`const DataAttributesType data_attr_print_job_name = "print_job_name"`	Название задания на печать
`const DataAttributesType data_attr_print_copies = "event_attr_print_copies"`	Количество копий (целое число в строковом представлении)
`typedef string IdentityAttrType`	**Общие атрибуты идентификаций:**
`const IdentityAttrType identity_attr_comment = "identity_comment"`	Комментарий
`typedef string TWorkstation`	**Атрибуты идентификаций типа "Компьютеры":**
`const IdentityAttrType identity_attr_ws_type = "workstation_type"`	Тип рабочей станции:
`const TWorkstation ws_type_computer = "ws_type_computer"`	Компьютер
`const TWorkstation ws_type_mobile = "ws_type_mobile"`	Мобильное устройство
`const TWorkstation ws_type_terminal = "ws_type_terminal"`	Терминальный сервер
	Атрибуты идентификаций типа "Устройство":
`const IdentityAttrType identity_attr_device_name = "device_name"`	Имя устройства (*)

Код	Описание атрибута
`const IdentityAttrType identity_attr_device_type = "device_type"`	Тип устройства
	Атрибуты идентификации типа "Устройство", "Ресурс", "Рабочая станция" для источника и приемника:
`const IdentityAttrType identity_attr_endpoint_name = "name"`	Имя устройства или ресурса (*)
`const IdentityAttrType identity_attr_endpoint_path = "path"`	Путь на заданном ресурсе или устройстве (*)
`const IdentityAttrType identity_attr_endpoint_type = "type"`	Тип ресурса или устройства. Заполняется мнемониками из справочника
`const IdentityAttrType identity_attr_endpoint_device_id`	ID съемного устройства
`const string identity_attr_print_location = "print_location"`	Расположение устройства принтера
`const string identity_attr_print_port_name = "print_port_name"`	Имя порта устройства принтера
`const string identity_attr_print_server = "print_server"`	Имя сервера устройства принтера
	Атрибуты идентификаций типа "Ресурс":
`const IdentityAttrType identity_attr_resource_url = "res_destination_url"`	URL ресурса (*)
`const IdentityAttrType identity_attr_resource_address = "res_destination_host"`	DNS или IP адрес ресурса (*)
Примечание: (*) - обязательный атрибут	

Примечание:

Если внешняя система присылает в Traffic Monitor событие, в котором атрибут `destination_path` заполнен несколькими значениями, то в Traffic Monitor событие будет сохранено с первым значением этого атрибута, а все остальные значения будут отброшены.

Сервис EventProcessor pushAPI SDK

В состав pushAPI SDK входит сервис EventProcessor. Он предоставляет API, который позволяет создавать и пересылать события и объекты перехвата. Ниже приведено его описание.

Код	Описание
`InterfaceVersion GetVersion()`	Функция получения текущей версии интерфейса
`void VerifyCredentials(1: Credentials cred) throws(1: InvalidCredentials ex);`	Функция проверки параметров доступа к интерфейсу pushAPI SDK. Позволяет проверить возможность передачи данных до их реальной передачи:
`EventId BeginEvent (1: Event event, 2: Credentials cred) throws(1: InvalidEventFormat ex1, 2: InvalidCredentials ex2, 3: LicenseError ex3, 4: EventNotFound ex4, 5: DataNotFound ex5);`	Функция инициирует процесс передачи события перехвата на Traffic Monitor, используя pushAPI SDK, где: • `event` – заполненная структура содержащая все «метаданные», которые описывают событие перехвата. Атрибуты: отправитель, получатель, список потоков данных, которые связаны с событием; • `cred` – структура, которая позволяет авторизовать процесс передачи события. Функция возвращает идентификатор события с сессии передачи данных (в одной и той же TCP-сессии можно передавать несколько событий).
`StreamId BeginStream (1: EventId event_id, 2: ItemId event_data_id) throws(1: EventNotFound ex1, 2: DataNotFound ex2);`	Функция инициирует передачу потока данных, который связан с событием перехвата данных, где: • `event_id` – идентификатор события, с которым связан данный поток данных. Возвращается функцией `BeginEvent`; • `event_data_id` – идентификатор передаваемых данных. Этот идентификатор передается в составе «метаданных» события, в поле `evt_data`; Функция возвращает идентификатор загружаемого потока.
`void SendStreamData(1: EventId object_id, 2: StreamId stream_id, 3: binary chunk)`	Функция передает буфер данных в составе потока данных, который связан с событием перехвата данных, где:

Код	Описание
`throws(1: EventNotFound ex1, 2: StreamNotFound ex2, 3: InvalidEventFormat ex3);`	• `object_id` – идентификатор события, с которым связан данный поток данных. Возвращается функцией `BeginEvent`; • `stream_id` – идентификатор потока данных, в составе которого передается буфер. Возвращается функцией `BeginStream`; • `chunk` – буфер данных. Пользователь SDK сам должен выбирать разумный размер буфера данных, чтобы обеспечить достаточную скорость загрузки событий, но не перегружать сетевую инфраструктуру
`void EndStream(1: EventId event_id, 2: StreamId stream_id) throws(1: EventNotFound ex1, 2: StreamNotFound ex2, 3: InvalidEventFormat ex3);`	Функция фиксирует конец передачи потока данных, связанного с событием перехвата, где: • `event_id` – идентификатор события, с которым связан данный поток данных. Возвращается функцией `BeginEvent`; • `stream_id` – идентификатор потока данных, в составе которого передается буфер. Возвращается функцией `BeginStream`
`string GetEventDatabaseId(1: EventId object_id) throws(1: EventNotFound ex1);`	Функция получения уникального идентификатора события перехвата в системе Traffic Monitor, где: • `object_id` – идентификатор события, полученный вызовом `BeginEvent`. Это значение можно использовать для ссылки на загруженное событие. Функцию можно вызывать после удачного выполнения функции `BeginEvent`
`void EndEvent(1: EventId event_id, 2: bool abort) throws(1: EventNotFound ex1, 2: InvalidEventFormat ex2, 3: LicenseError ex3);`	Функция фиксирует конец передачи события перехвата, где: • `event_id` – идентификатор события, передача которого завершается; • `abort` – флаг прекращения обработки события в Traffic Monitor. Если выставлен флаг, то событие не попадает в БД

9.1.3 Регистрация стороннего компонента pushAPI SDK

Для интеграции стороннего модуля перехвата данных с системой Traffic Monitor нужно выполнить следующие шаги:

1. Зарегистрировать в компании InfoWatch строку идентификатор компании-производителя компонента. Это позволит связывать компанию-производителя с лицензией, которая будет устанавливаться в систему Traffic Monitor;
2. Получить лицензию разработчика, связанную с идентификатором компании. Это позволит загружать в Traffic Monitor события перехвата этого производителя. Лицензия разработчика позволяет загружать произвольные типы (Сервисы) событий;
3. Создать файл регистрации стороннего компонента в описанном ниже формате. К этому моменту должны быть определены названия сервисов, которые будут перехватываться сторонним компонентом. В файл регистрации добавляется лицензия, полученная на шаге 2;
4. Загрузить файл регистрации стороннего компонента в систему Traffic Monitor через интерфейс консоли Traffic Monitor. Раздел «Управление» - «Плагины». (см. "InfoWatch Traffic Monitor. Руководство пользователя");
5. Получить токен доступа в свойствах загруженного стороннего компонента.

Система готова принимать события от заданной компании производителя с указанным сервисом перехвата.

Формат файла регистрации стороннего компонента pushAPI SDK

Плагин представляет собой архив в формате .zip. В состав архива входят:

- папка **licenses**, содержащая файлы лицензий;
- папка **icon**, содержащая файлы с используемыми пиктограммами для регистрируемых событий;
- файл **manifest.json**, содержащий информацию о плагине.

> **Примечание:**
>
> Папки **licenses** и **icon** не являются обязательными. Файлы лицензий и файлы с используемыми пиктограммами могут находиться в корне.

Для внешних систем-источников событий файл **manifest.json** должен содержать следующую информацию:

Содержимое	Тип данных	Является обязательным	Описание
`{`			
`"PLUGIN_ID": "",`	Строка	Да	Уникальный идентификатор (UUID) плагина, который создается его разработчиком. До 40 символов. Идентификатор используется для обновления плагина. **Пример:** `"PLUGIN_ID": "189C38D390396EB6E0530100007F1CA200000001",`

Содержимое	Тип данных	Является обязательным	Описание
`"DISPLAY_NAME": "",`	Строка	Да	Отображаемое имя плагина. **Пример:** `"DISPLAY_NAME": "Имя плагина",`
`"DESCRIPTION": "",`	Строка	Нет	Отображаемое описание плагина. **Пример:** `"DESCRIPTION": "Тестовый плагин для демонстрации",`
`"VERSION": "",`	Строка	Да	Версия плагина. **Пример:** `"VERSION": "1.0.0",`
`"IS_SYSTEM": ,`	Логический тип	Нет	Признак "Предустановленный". Входит плагин ли в состав Системы. **Пример:** `"IS_SYSTEM": false,`
`"VENDOR": "",`	Строка	Да	Идентификатор компании-разработчика, соответствующий названию компании в лицензии. **Пример:** `"VENDOR": "infowatch",`
`"LICENSE": [ { "PATH":"" } ],`	Массив Строка	Да	Файлы лицензии. Должны быть указаны пути к файлам лицензий относительно корня архива. **Пример** `"LICENSE": [ { "PATH": "licenses/файл лицензии 1.license" }, { "PATH": "licenses/файл лицензии 2.license" }, ],`
`"PATTERN_SEARCH_LICENSE": "",`	Строка	Нет	Шаблон для поиска загруженных ранее лицензий для привязки их к плагину. Имеет формат: `{operator:or|and,child[{name:value}]}"`. **Пример:** `"PATTERN_SEARCH_LICENSE": "{\"operator\":\"or\", \"conditions\":[{\"origin\":`

Содержимое	Тип данных	Является обязательным	Описание
			`\"dm\"},{\"origin\": \"dmmobile\"}]}",`
`"ADDS_EVENTS": { "OBJECT_TYPE": [ { "MNEMO": "", "SERVICE_MNEMO": "", "ICON": "", "LOCALE": { "": "" } } ], "PROTOCOL": [ { "MNEMO": "", "LOCALE": { "": "" } } ] },`	Объект Массив Строка Строка Строка Объект Массив Строка Объект	Да	Регистрируемые типы событий и протоколы. Массив типов регистрируемых событий с указанием типов сервисов, к которым они относятся. Тип события. Сервис. Файлы с иконками регистрируемых контактов. Локализации наименований типов контактов хотя бы на одном языке. **Пример** `"ADDS_EVENTS": { "OBJECT_TYPE": [{ "MNEMO": "cloud_storage", "SERVICE_MNEMO": "file", "LOCALE": { "rus": "Облачное хранилище", "eng": "Cloud storage" } }, { "MNEMO": "im_telegram", "SERVICE_MNEMO": "im", "ICON": "icon/event_icon_telegram.png", "LOCALE": { "rus": "Telegram", "eng": "Telegram" } }], "PROTOCOL": [{ "MNEMO": "telegram", "LOCALE": { "rus": "Telegram", "eng": "Telegram" } }] },`

Содержимое	Тип данных	Является обязательным	Описание
`"USES_EVENTS": { "OBJECT_TYPE": [ { "MNEMO": "", "PROTOCOL": [ { "MNEMO": "" } ], "ORIGIN": [ "" ] } ] },`	Объект Массив Строка Массив Строка Массив Строка	Да	Существующие типы событий. Массив типов регистрируемых событий с указанием типов сервисов, к которым они относятся. Тип события. Массив протоколов, относящихся к данному типу события. Протоколы. Массив кодов типов перехватчиков для Плагинов InfoWatch. Коды. **Пример** `"USES_EVENTS": { "OBJECT_TYPE": [{ "MNEMO": "email", "PROTOCOL": [{ "MNEMO": "pop3" }, { "MNEMO": "imap" }, { "MNEMO": "mapi" }, { "MNEMO": "smtp" }], "ORIGIN": ["dm", "dmmobile"] }, { "MNEMO": "email_web", "PROTOCOL": [{ "MNEMO": "http" }, { "MNEMO": "https" }], "ORIGIN": ["dm", "dmmobile"] }, { "MNEMO": "im_whatsapp",`

Содержимое	Тип данных	Является обязательным	Описание

			<pre>"PROTOCOL": [
			{
			"MNEMO": "whatsapp"
			}
			],
			"ORIGIN": ["dm", "dmmobile"]
			}
			]
			},</pre>

<pre>"ADDS_SERVICES": {
"SERVICE_TYPE": [
{
"SERVICE_MNEMO": "",
"DATA_CLASS": [""],
"ICON": "",
"LOCALE": {
"": "",
"": ""
}
}
],
},</pre>
	Объект
	Массив
	
	Строка
	Массив строк
	Строка
	Объект
		Да	Регистрируемые типы событий через сервисы SDK.
			Регистрируемые типы перехватываемых сервисов.
			
			Имя перехватываемого сервиса.
			Список классов данных, которые перехватывает данный перехватчик в сервисе.
			Файл с иконкой регистрируемого сервиса.
			Локализации наименований типов регистрируемого сервиса хотя бы на одном языке.
			
			Пример
			<pre>"ADDS_SERVICES": {
			"SERVICE_TYPE": [{
			"SERVICE_MNEMO": "newservice",
			"DATA_CLASS": ["kChat", "kFileExchange"],
			"ICON": "icon/TM_icon.png",
			"LOCALE": {
			"rus": "сервис#5",
			"eng": "service#5"
			}
			},
			{
			"SERVICE_MNEMO": "newservice2",
			"DATA_CLASS": ["kPhoto"],
			"ICON": "icon/TM_icon.png",
			"LOCALE": {
			"rus": "сервис#5+2",
			"eng": "service#5+2"
			}
			}</pre>

Содержимое	Тип данных	Является обязательным	Описание
			] },

"OBJECT_HEADER": [
{
"NAME": "",
 	
"NOTE": {
"": "",
"": ""
},
"TYPE": "",
"FORMAT": "",
 	
"DATA_CLASS": [""],
"USE_IN_POLICY": "",
"USE_IN_NOTIFICATION": "",
"USE_IN_QUERY": "",
"USE_IN_LIST": "",
"USE_IN_SHOW": "",
"USE_IN_DETAIL": "",
"IS_MULTIPLE_VALUE": "",
"VALIDATION_RULE": [
{
"type": ""
}]
}
]
	Массив
	
	Строка
	
	Объект
	Строка
	
	Перечисление
	Перечисление
	
	Массив строк
	Перечисление
	Перечисление
	
	Перечисление
	Перечисление
	Перечисление
	Перечисление
	
	Перечисление
	Объект
	
	Перечисление
		Нет	Пользовательские атрибуты событий.
			
			Уникальный код атрибута, который будет использован в заголовках события и для идентификации в xAPI/PushAPI.
			Если идентификатор разработчика не "IW", код должен начинаться с префикса "<Идентификатор компании-разработчика>_".
			Пример: CISCO_CALL_DURATION
			Название атрибута, отображаемое в консоли Traffic Monitor.
			
			Тип значения.
			Формат записи. Возможны значения: число (целое, дробное), строка, дата и время в UTC + указание смещение часового пояса, длительность, гиперссылка, перечисление, логический тип.
			Сервисы, к событиям которых будет добавлен новый пользовательский атрибут.
			Использование атрибута в политиках. "1" - да, "0" - нет.
			Использование атрибута в почтовых уведомлениях. "1" - да, "0" - нет.
			
			Использование атрибута в запросах. "1" - да, "0" - нет.
			Использование атрибута в табличной форме просмотра списка событий. "1" - да, "0" - нет.
			Использование атрибута в краткой форме просмотра события. "1" - да, "0" - нет.
			Использование атрибута в детальной форме просмотра события. "1" - да, "0" - нет.
			Множественное значение. "1" - да, "0" - нет.
			Правило проверки вводимых пользователем значений в запросах и политиках. Определяет возможные

Содержимое	Тип данных	Является обязательным	Описание

			значения пользовательского атрибута. Можно задать следующие типы данных:
			• Логический
			• Строка
			• Целочисленный
			• Число с плавающей точкой
			• Длительность
			• Перечисление
			• Ссылка
			• Регулярное выражение
			Регулярное выражение задается в формате PCRE (Perl Compatible Regular Expressions).
			Примеры поля VALIDATION RULE
			`"VALIDATION_RULE": [ { "type": "boolean", "value": ["yes","no"] } ]`
			`"VALIDATION_RULE": [ { "type": "string", "min": 1, "max": 100 } ]`
			`"VALIDATION_RULE": [ { "type": "integer", "min": 0, "max": 100 } ]`
			`"VALIDATION_RULE": [ { "type": "float", "min": 3, "max": 4 } ]`
			`"VALIDATION_RULE": [ {`

Содержимое	Тип данных	Является обязательным	Описание

			<pre>"type": "duration",
			"value": [[1,31],[33,59]]
			}
			]
			
			"VALIDATION_RULE": [
			{
			"type": "enum",
			"enum": [
			"value1",
			"value2",
			"value3"
			]
			}
			]
			
			"VALIDATION_RULE": [
			{
			"type": "link",
			"link": "http://ya.ru"
			}
			]
			
			"VALIDATION_RULE": [
			{
			"type": "regexp",
			"pattern": "\d"
			}
			]</pre>
			Пример описания пользовательского атрибута
			<pre>"OBJECT_HEADER": [
			{
			"NAME": "header_multi_string",
			"NOTE": {
			"rus": "Заголовок строк1",
			"eng": "Header strings1"
			},
			"DATA_CLASS": ["kEmail"],
			"USE_IN_POLICY": "1",
			"USE_IN_QUERY": "1",
			"USE_IN_NOTIFICATION": "1",
			"USE_IN_LIST": "1",
			"USE_IN_SHOW": "1",
			"USE_IN_DETAIL": "1",
			"TYPE": "string",
			"FORMAT": "string",
			"IS_MULTIPLE_VALUE": "1"
			},</pre>

Содержимое	Тип данных	Является обязательным	Описание

			<pre>{
			"NAME": "header_date",
			"NOTE": {
			"rus": "Заголовок дата1",
			"eng": "Header date1"
			},
			"DATA_CLASS": ["kEmail"],
			"USE_IN_POLICY": "1",
			"USE_IN_QUERY": "1",
			"USE_IN_NOTIFICATION": "1",
			"USE_IN_LIST": "0",
			"USE_IN_SHOW": "1",
			"USE_IN_DETAIL": "1",
			"TYPE": "string",
			"FORMAT": "date",
			"IS_MULTIPLE_VALUE": "0"
			}
			]

}			

Примечание:

Корневые поля `"ADDS_EVENTS"` и `"USES_EVENTS"` используются только для плагинов, разрабатываемых компанией InfoWatch. Для плагинов сторонних разработчиков используется поле `"ADDS_SERVICES".`

Полная JSON-схема входит в состав примеров SDK (см. раздел "Описание примеров PushAPI SDK"). Пример файла регистрации модуля входит в состав SDK, а также приведен в статье Пример файла регистрации стороннего компонента.

Важно!

В состав файла входит просроченная лицензия. Для регистрации плагина на основании файла примера нужно будет получать новую лицензию.

Пример файла регистрации стороннего компонента

Перед созданием файла регистрации стороннего компонента определите идентификатор компании-разработчика, соответствующий названию компании, зарегистрируйте его в компании InfoWatch, а также определите название разрабатываемого сервиса.
В качестве исходных данных для примера возьмем:

- Идентификатор компании: `MY_COMPANY;`
- Название сервиса: `im_my_chat` – сервис перехвата сообщений в мессенджере.

Чтобы создать файл регистрации стороннего компонента:

1. Получите лицензию разработчика.
2. При необходимости выберите иконки для плагина и регистрируемых им событий.
3. Опишите manifest-файл плагина.

Получение лицензии разработчика

После определения названия сервиса получите в компании InfoWatch лицензию для плагина (подробнее см. статью "Запрос лицензии" в Руководстве пользователя). Для компании *MY_COMPANY* файл лицензии будет иметь следующий вид:

```
    "License": {
        "ActiveDays": 60,
        "Features": [
            {
                "common_name": "MY_COMPANY",
                "object_type": "im_my_chat",
                "protocol": "*"
            }
        ],
```

Где поле `protocol` может принимать следующие значения:

- `"none"` – отсутствие протокола;
- значение конкретного протокола;
- `"*"` – любой протокол.

Описание manifest-файла плагина

На основании исходных данных заполните все обязательные корневые поля `"PLUGIN_ID"`, `"DISPLAY_NAME"`, `"VERSION"`, `"VENDOR"`, `"LICENSE"` и `"ADDS_SERVICES"`.

Так как плагин `im_my_chat` предназначен для перехвата сообщений в мессенджере, значение поля `"DATA_CLASS"` в корневом поле `"ADDS_SERVICES"` будет равно `"kChat"`. Перехватываться будут события класса «**Беседа**».

Чтобы описать представление контакта пользователя в перехватываемых событиях добавьте поле `"CONTACT_TYPE"` в корневое поле `"ADDS_SERVICES"`.

Чтобы добавить пользовательские атрибуты события, используйте конструкцию `"OBJECT_HEADER"`. Название пользовательского атрибута события должно начинаться с названия компании-разработчика плагина. Количество атрибутов не ограничено. В данном примере добавим два атрибута:

- `"MY_COMPANY_FORWARDING_STATUS"` – статус пересылки сообщения, т.е. является ли сообщение пересланным.
  Может принимать значения:
  - `original`;
  - `forwarded`;
- `"MY_COMPANY_ORIGINAL_SENDER"` – оригинальный отправитель. Может принимать любое значение типа данных "Строка".

Manifest-файл плагина будет выглядеть следующим образом:

```
{
    "PLUGIN_ID": "189C38D390396EB6E0540100007F1CA200000001",
    "DISPLAY_NAME": "Test plugin",
    "DESCRIPTION": "Test plugin to demonstrate",
    "VERSION": "1.0.0",
    "IS_SYSTEM": false,
    "VENDOR": "MY_COMPANY",
```

```
   "ADDS_SERVICES": {
      "SERVICE_TYPE": [{
         "SERVICE_MNEMO": "im_my_chat",
              "DATA_CLASS": ["kChat"],
              "ICON": "icon/TM_icon.png",
              "LOCALE": {
                        "rus": "Новый сервис",
                        "eng": "New service"
              },
              "CONTACT_TYPE" : [{
                                  "MNEMO" : "my_chat_Id",
                                  "SCOPE" : ["person"],
                                  "ICON": "icon/TM_icon.png",
                                  "LOCALE": {
                                            "rus": "Контакт тестового плагина",
                                            "eng": "Test plugin contact"
                                  }
                        }
              ]
      }]
   },
   "LICENSE": [
   { "PATH" : "licenses/tm_license_2023-07-24_.license" }
   ],
   "OBJECT_HEADER": [{
        "NAME": "MY_COMPANY_FORWARDING_STATUS",
        "NOTE": {
                "rus": "Статус пересылки сообщения",
                "eng": "Forwarding Message Status"
        },
        "TYPE": "string",
        "FORMAT": "enum",
        "DATA_CLASS": ["kChat"],
        "USE_IN_POLICY": "1",
        "USE_IN_QUERY": "1",
        "USE_IN_NOTIFICATION": "1",
        "USE_IN_LIST": "1",
        "USE_IN_SHOW": "1",
        "USE_IN_DETAIL": "1",
        "IS_MULTIPLE_VALUE": "0",
        "VALIDATION_RULE": [{
                            "type": "enum",
                            "enum": [
                                    "original",
                                    "forwarded"
                            ]
   }]
   },
   {
        "NAME": "MY_COMPANY_ORIGINAL_SENDER",
        "NOTE": {
                "rus": "Оригинальный отправитель",
                "eng": "Original sender"
        },
        "TYPE": "string",
        "FORMAT": "string",
```

```
        "DATA_CLASS": ["kChat"],
        "USE_IN_POLICY": "0",
        "USE_IN_QUERY": "0",
        "USE_IN_NOTIFICATION": "1",
        "USE_IN_LIST": "0",
        "USE_IN_SHOW": "1",
        "USE_IN_DETAIL": "1",
        "IS_MULTIPLE_VALUE": "0"
    }]
}
```

В нашем случае файл лицензии будет располагаться в файле регистрации плагина, т.е. в zip-архиве, в папке licenses. Это описано в manifest-файле следующим образом:

```
    "LICENSE": [
   { "PATH" : "licenses/tm_license_2023-07-24_.license" }
   ],
```

Если файл лицензии отдельно загружен в Traffic Monitor в разделе **Управление** -> **Лицензии**, то в manifest-файле плагина необходимо задать шаблон для поиска загруженных ранее лицензий:

```
        "PATTERN_SEARCH_LICENSE": "{\"operator\":\"and\",\"conditions\":
[{\"common_name\":\"MY_COMPANY\"},{\"object_type\":\"im_my_chat\"},
{\"protocol\":\"*\"}]}",
    "LICENSE": []
}
```

Полям `"VENDOR"` и `"SERVICE_MNEMO"` manifest-файла соответствуют наименования `"common_name"` и `"object_type"` в Traffic Monitor. Поэтому в шаблоне для поиска лицензий их значения равны соответственно `MY_COMPANY` и `im_my_chat`.

Работа с плагином

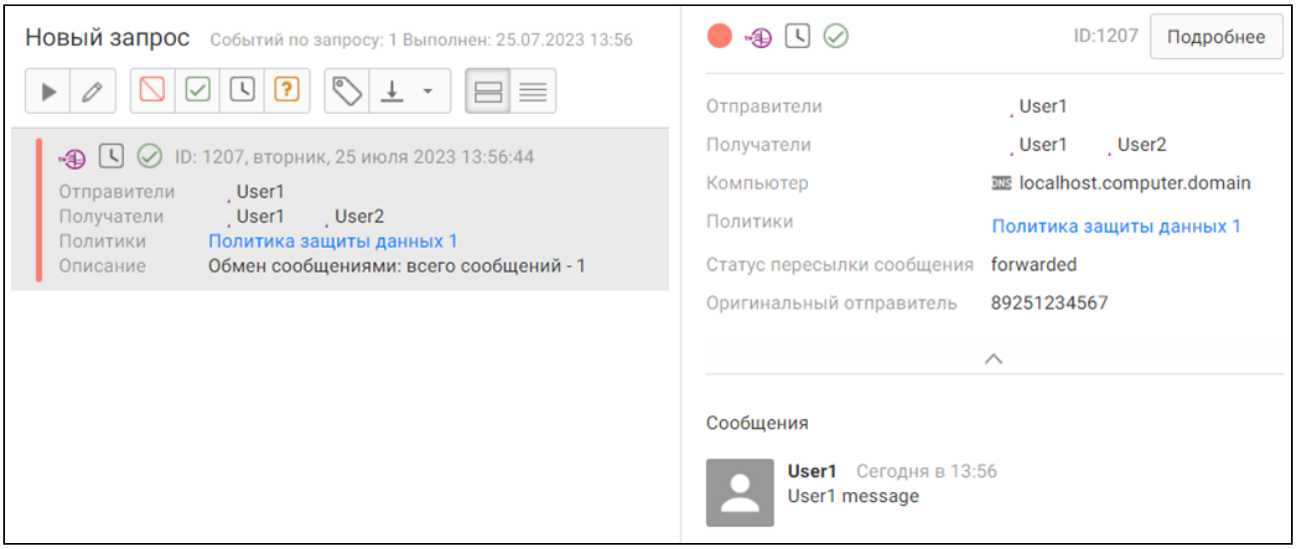
После того как лицензия получена, выбраны иконки и описан manifest-файл, создайте файл регистрации стороннего компонента и загрузите его в Traffic Monitor в разделе **Управление** -> **Плагины**.

Для начала работы с зарегистрированным плагином можно использовать утилиту pushapi-util. Подробнее о ее использовании см. Описание примеров PushAPI SDK.

Чтобы отправить в Traffic Monitor событие класса «**Беседа**» сервиса «**im_my_chat**», используйте команду вида:

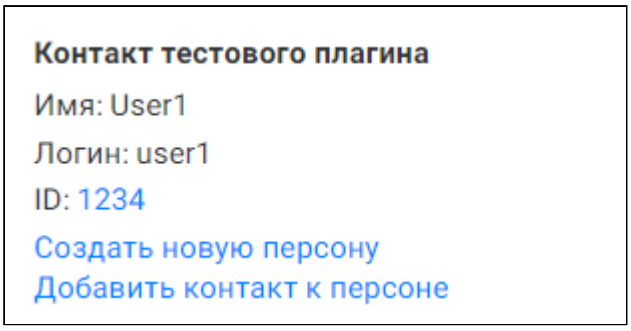
```
pushapi-util \
--host TM_host \
--token im_my_chat_token_value \
--id MY_COMPANY \
--class imchat \
--service im_my_chat \
--evtattr event name:"new service chat Event" \
--sender "my_chat_Id:1234:'{\"display_name\":\"User1\",\"login\":\"user1\",
\"url\":\"https://my_chat.ru/user1\"}'" \
--receiver "my_chat_Id:76543:'{\"display_name\":\"User2\",\"login\":\"user1\",
\"url\":\"https://my_chat.ru/user2\"}'" \
--mes "my_chat_Id:1234:'{\"display_name\":\"User1\",\"login\":\"user1\",
\"url\":\"https://my_chat.ru/user1\"}'","User1 message" \
--evtattr MY_COMPANY_FORWARDING_STATUS:forwarded \
--evtattr MY_COMPANY_ORIGINAL_SENDER:89251234567
```

В Консоли Управления Traffic Monitor событие будет выглядеть следующим образом:



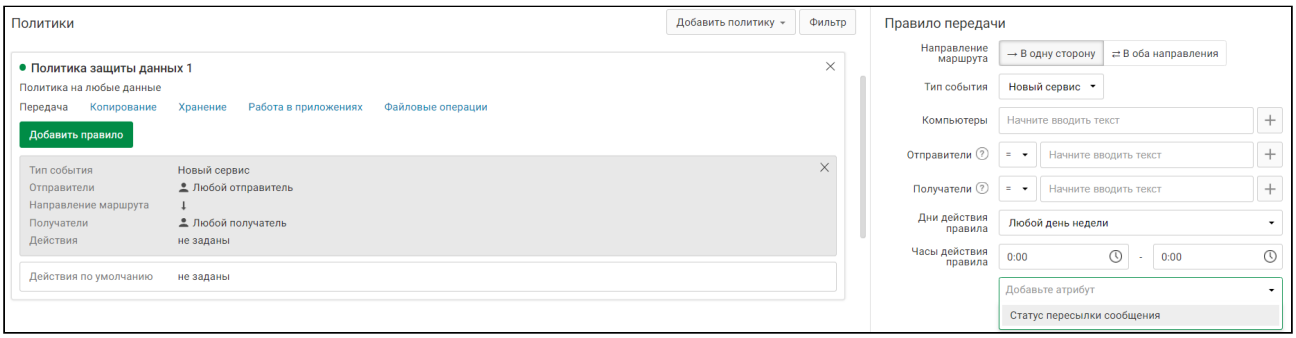
Заданные пользовательские атрибуты отображаются в описании события. Они не являются обязательными, а значит, если они не будут определены в команде, то и в событии они будут отсутствовать.

Так как заполнено поле `"CONTACT_TYPE"` в manifest-файле, контакт пользователя в событии отображается следующим образом:

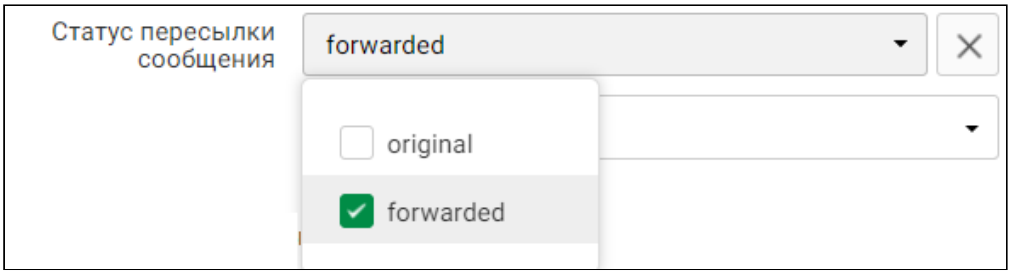


Использование пользовательского атрибута в политике защиты данных

У пользовательского атрибута `"MY_COMPANY_FORWARDING_STATUS"` в поле `"USE_IN_POLICY"` задано значение `"1"`. Это означает, что данный атрибут можно использовать при создании политики защиты данных.



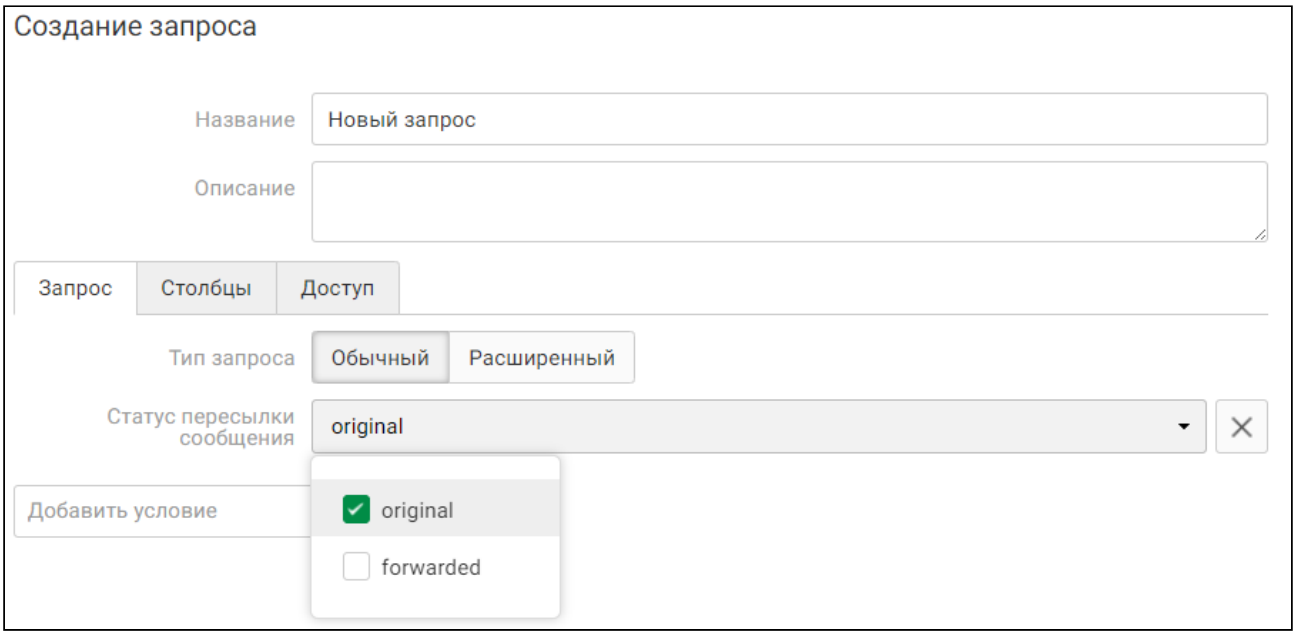
В параметрах *Правила передачи* отображается атрибут «*Статус пересылки сообщения*». Для перехвата пересылаемых сообщений задайте значение «*forwarded*» для этого атрибута.



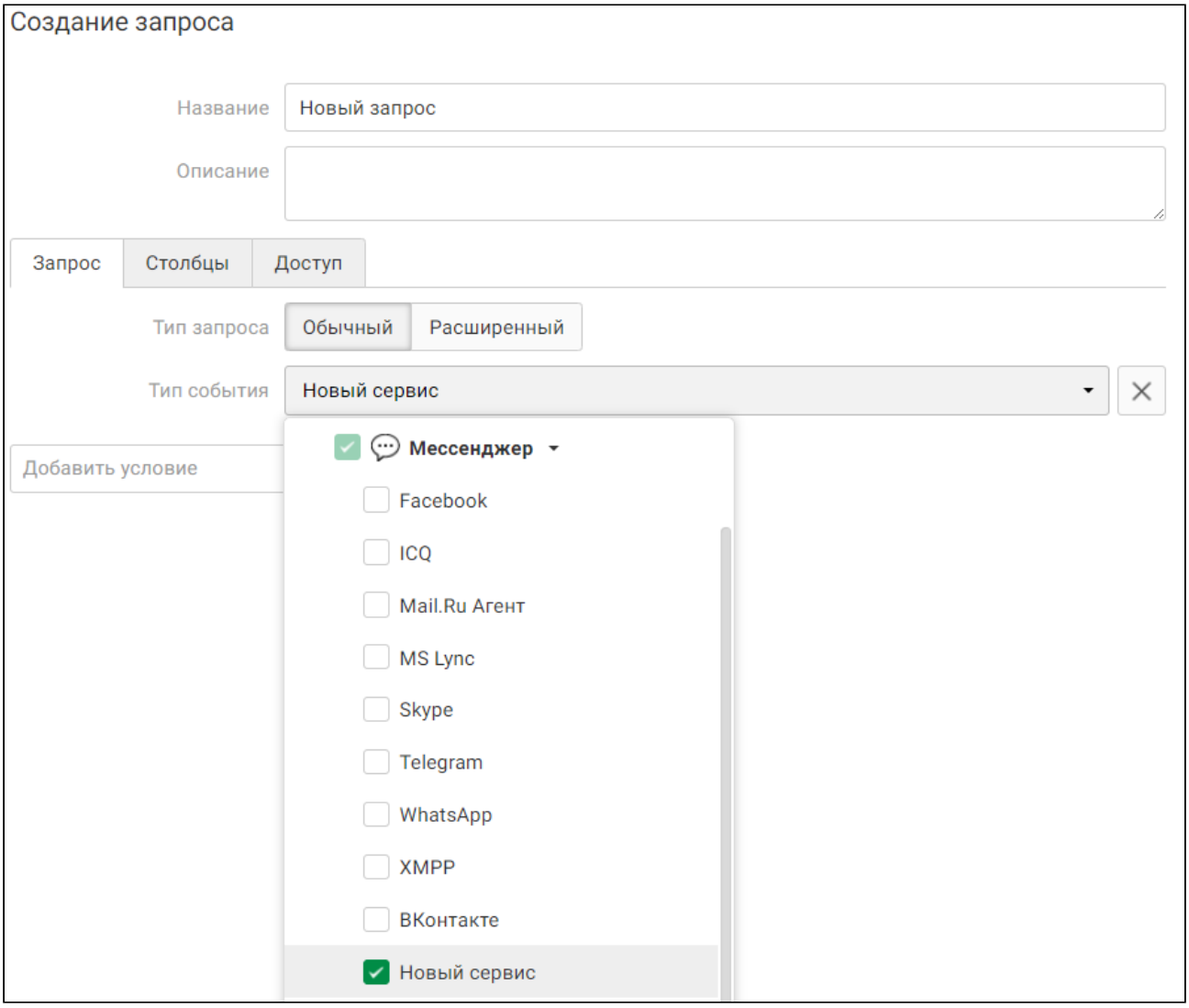
В результате политика будет срабатывать для событий с атрибутом `MY_COMPANY_FORWARDING_STATUS:forwarded` .

Создание запросов с новым сервисом

У пользовательского атрибута `"MY_COMPANY_FORWARDING_STATUS"` в поле `"USE_IN_QUERY"` задано значение `"1"` . Данный атрибут можно использовать при создании запросов в разделе **События**.



Также при создании запросов можно использовать и сам сервис. Он доступен при выборе типа события.



9.1.4 Описание примеров PushAPI SDK

Примеры представляют собой исходные коды консольных утилит, которые реализуют загрузку событий в Traffic Monitor, используя Thrift-интерфейс pushAPI SDK.

Первая утилита – pushapi-cpp. Реализована на языке C++. Подготовлена к сборке в операционных системах Windows и Linux. Данная утилита демонстрирует сценарии загрузки всех типов событий, которые уже зарегистрированы в системе Traffic Monitor.

Вторая утилита – pushapi-csharp. Реализована на языке C#. Подготовлена к сборке в операционной системе Windows. Данная утилита демонстрирует сценарии загрузки всех типов событий, которые уже зарегистрированы в системе Traffic Monitor.

Третья утилита – pushapi-demo.py. Реализована на языке Python. Подготовлена к сборке в операционных системах Windows и Linux. Демонстрирует сценарии загрузки всех типов событий, которые уже зарегистрированы в системе Traffic Monitor.

Четвертая утилита – pushapi-util. Реализована на языке C++. Подготовлена к сборке в операционных системах Windows и Linux. Данная утилита имеет развитый интерфейс командной строки, который позволяет загружать произвольные события в систему Traffic Monitor, в том числе те, которые зарегистрированы внешней системой. Вот описание командной строки:

```
./pushapi-util
--host <pushapi address > [--port <pushapi port>]
--token <plugin token>
--id <company>
--class <event class>
--service <event service>
--evtattr <attr_name:attr_value,…, attr_name:attr_value>
--sender
<contact_type:contact_value,attr_name:attr_value,…,attr_name:attr_value>
--receiver
<contact_type:contact_value,attr_name:attr_value,…,attr_name:attr_value>
--data file:<data_falie_name>, attr_name:attr_value,…, attr_name:attr_value
--mes <contact_type:contact_value,"message text"
[--addctx]
```

Где:

- <pushapi address > – адрес pushAPI-сервера;
- <pushapi port> – порт pushAPI-сервера, если не указан, используется значение по умолчанию;
- <plugin token> – токен авторизации плагина. Скопируйте его значение в разделе данного плагина консоли управления Traffic Monitor;
- <company> – идентификатор компании-производителя плагина. Произвольная строка, обязательно совпадающая с той, что использовалась для генерации лицензии для плагина;
- <event class> – класс передаваемого события;
- <event service> – сервис передаваемого события. Плагин может зарегистрировать свои перехватываемые сервисы по имени и передавать их;
- --evtattr – атрибуты передаваемого события. Посмотреть можно в списке команд с примером. Некоторые имена атрибутов имеют фиксированные значения и бизнес-смысл;
- --sender – отправитель события. Первая пара параметров определяет контакт отправителя. Далее указываются атрибуты отправителя. Существуют три «особых» типа контактов: dev (отправитель/получатель является устройством), res (отправитель/получатель является ресурсом), ws (отправитель/получатель является «Рабочей станцией»). Если не задан ни один отправитель, то будет сгенерировано два отправителя с типом «Персона» и контактом типа «auth»; и с типом «Рабочая станция» и контактом dnshostname. Задан может быть только один отправитель;

  Пример:

  Для события с отправителем с аккаунтом Вконтакте и указанием мета-информации:
  ```
  --sender "vkontakte:349051292:'{\"display_name\":\"some name\",\"url\":
  \"some_url\",\"login\":\"some_login\"}'"
  ```

- --receiver – получатель события. Параметры аналогичны отправителю;
- --data – данные события. Первая пара атрибутов указывает на путь к файлу, где содержится тело события. Остальные параметры описывают атрибуты этих данных. Для некоторых типов событий некоторые атрибуты обязательны;
- --mes – передача сообщений класса «Беседа». Указывается контакт отправителя сообщения и его содержимое. Сообщений может быть несколько;

- --addctx – необязательный параметр. Автоматически добавляет к событию двух отправителей с типом «Персона» и контактом типа «auth»; и с типом «Рабочая станция» и контактом типа dnshostname.

Утилиту pushapi-util можно установить из пакета QATOOLS. Для этого:

1. На рабочей станции с установленным Traffic Monitor извлеките pushapi-util в каталог `/opt/iw/tm5/bin`. Утилита pushapi-util находится в пакете `iwtm-qatools-x.x.x.xxx-release.xrpm/iwtm-qatools-x.x.x.xxx-release.x86_64.cpio/./opt/iw/tm5/bin`. Здесь `x.x.x.xxx` – это номер сборки.

   > **Примечание:**
   >
   > Если pushapi-util извлечена в другой каталог, то добавьте его в переменные окружения: `export PATH=$PATH:<directory>`

2. Перейдите в каталог, куда извлечена утилита pushapi-util.
3. Сделайте файл утилиты исполняемым: `chmod u+x pushapi-util`

Примеры вызова утилиты pushapi-util есть в исходных кодах примеров, а также приведены в статье Примеры использования утилиты pushapi-util.

Особенности сборки примеров PushAPI SDK приведены в следующих статьях:

- Сборка примеров под Linux pushAPI SDK
- Сборка примеров под Windows pushAPI SDK

В состав примеров SDK также входят:

- файл plugin.json – JSON-схема файла manifest.json, который входит в состав файла регистрации внешнего плагина;
- файл plugin.zip – пример файла регистрации внешнего плагина.

  > **Важно!**
  >
  > В состав файла входит просроченная лицензия. Для регистрации плагина на основании файла примера нужно получить новую лицензию.

Примеры использования утилиты pushapi-util

С помощью тестовой утилиты pushapi-util можно отправить в Traffic Monitor события для любых встроенных сервисов и классов, используя встроенный плагин Device Monitor.

> **Примечание:**
>
> В командах в качестве значения атрибута *token* используйте значение токена Device Monitor. Его можно скопировать в разделе **Управление** -> **Плагины** Консоли Управления Traffic Monitor.

Примеры отправки событий класса «Почта»

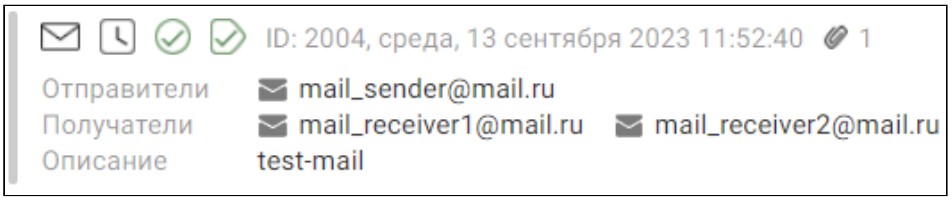
Пример 1:

Чтобы отправить в Traffic Monitor событие класса «**Почта**» одноименного сервиса, используйте команду вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class email \
--service email \
--evtattr event_name:"Send email" \
--sender email:"mail_sender@mail.ru" \
--receiver email:"mail_receiver1@mail.ru" \
--receiver email:"mail_receiver2@mail.ru" \
--data file:/root/test_email.eml
```

Где `test_email.eml` – это файл, содержащий передаваемое письмо. Атрибуты `data`, `sender` и `receiver` являются обязательными.

В Консоли Управления Traffic Monitor событие будет выглядеть следующим образом:



Пример 2:

Чтобы отправить в Traffic Monitor событие класса «**Почта**» сервиса «**Веб-почта**», используйте команду вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class email \
--service email_web \
--evtattr event_name:"Send email_web" \
--sender email:"mail_sender@mail.ru" \
--receiver email:"mail_receiver1@mail.ru" \
--data file:/root/test_email.eml
```

Примеры отправки событий класса «Беседа»

Пример 1:

Чтобы отправить в Traffic Monitor событие класса «**Беседа**» для мессенджера ICQ, используйте команду вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class imchat \
--service im_icq \
```

```
--evtattr event_name:"ICQ chat event" \
--receiver icq:IcqReceiver \
--mes icq:IcqSender,"Confidential information"
```

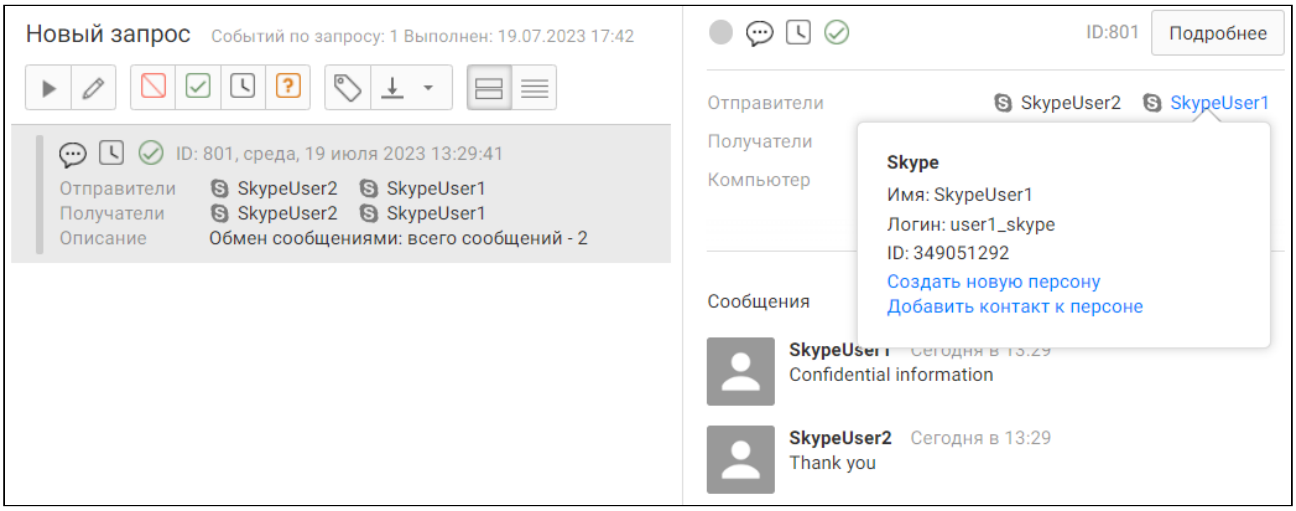
Атрибуты `mes` и `receiver` являются обязательными.

Пример 2:

Чтобы отправить в Traffic Monitor событие класса «**Беседа**» для мессенджера Skype, содержащее метаинформацию об отправителях, используйте команду вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class imchat \
--service im_skype \
--evtattr event_name:"Skype event" \
--mes "skype:349051292:'{\"display_name\":\"SkypeUser1\",\"login\":
\"user1_skype\"}'","Confidential information" \
--mes "skype:349051100:'{\"display_name\":\"SkypeUser2\",\"login\":
\"user2_skype\"}'","Thank you"
```

Здесь для отправителей сообщений указана метаинформация, включающая в себя отображаемое имя и логин отправителя. В Консоли Управления Traffic Monitor событие будет выглядеть следующим образом:



Примеры отправки событий класса «Печать, копирование и сканирование»

Чтобы отправить в Traffic Monitor событие класса «**Печать, копирование и сканирование**» сервиса «**Принтер и МФУ**», используйте команды следующего вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class mfp \
--service print \
--evtattr event_name:"Print event" \
--receiver dev:"Some Printer Name" \
--data file:/root/test.png,filename:test.png
```

и

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class mfp \
--service print \
--evtattr event_name:"Print event" \
--receiver dev:"Some Printer Name",print_port_name:"Some print port
name",print_location:"Some print location" \
--data file:/root/test.png,print_copies:3,print_job_name:"Some print job name"
```

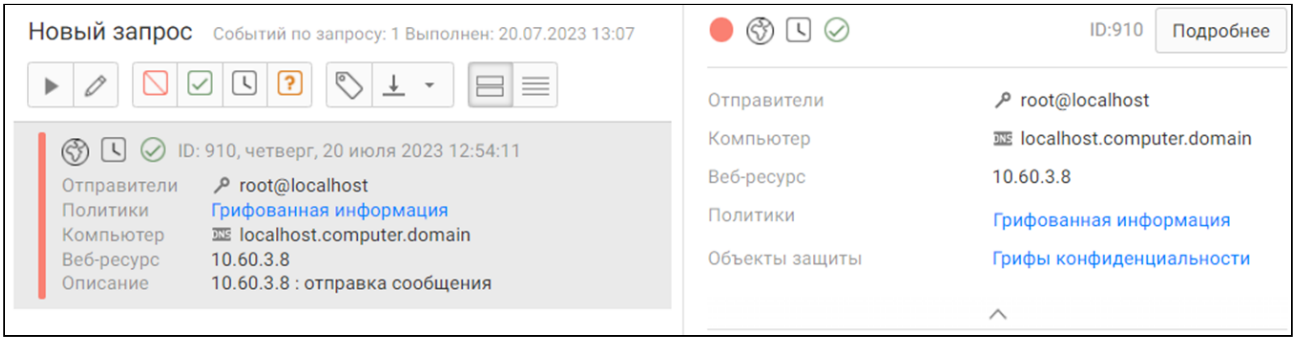
Атрибуты `receiver` и `data` являются обязательными. Также команда должна содержать один из атрибутов `filename` и `print_job_name`.

Примеры отправки событий класса «Интернет-активность»

Чтобы отправить в Traffic Monitor событие класса «**Интернет-активность**» одноименного сервиса, используйте команду следующего вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class web \
--service web_common \
--evtattr event_name:"Web event1" \
--receiver res:"drive.google.com",res_destination_url:"/file/test" \
--data file:/root/test
```

Где `test` – это файл, содержащий POST-запрос. В нашем примере в передаваемом запросе присутствует фраза «строго конфиденциально». При настроенных политиках защиты данных событие будет выглядеть следующим образом:



Атрибуты `data` и `receiver` являются обязательными.

Пример отправки событий класса «Фотосъемка»

Для отправки событий класса «**Фотосъемка**» нет встроенных сервисов, поэтому для этого необходимо зарегистрировать свой сервис и использовать команду следующего вида:

```
pushapi-util \
--host TM_host \
--token your_plugin_token_value \
--id your_company \
--class photo \
--service your_multimedia_service \
--evtattr event_name:"Photo event" \
```

```
--receiver ws:"ws1.domain.org",workstation_type:"ws_type_computer" \
--data file:/root/test.png,filename:test.png
```

Примеры отправки событий класса «Обмен файлами»

Атрибуты `receiver`, `data` и `destination_file_path` обязательны для событий сервисов «**Облачные хранилища**», «**Терминальная сессия**», «**Сетевые ресурсы**», «**FTP**» и «**Съемное устройство**».

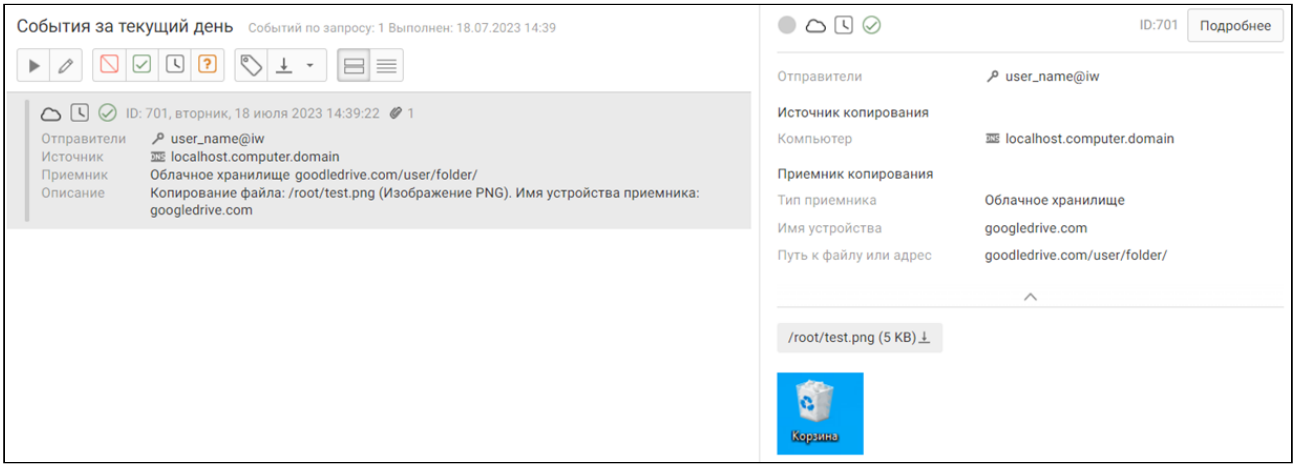
Атрибуты `receiver` и `data` обязательны для событий класса **"Обмен файлами"** сервиса «**Мессенджеры**».

Пример 1:

Чтобы отправить в Traffic Monitor событие класса «**Обмен файлами**» сервиса «**Облачные хранилища**», используйте команду вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class fileexch \
--service cloud_storage \
--evtattr event_name:"cloud_storage event" \
--sender auth:user_name@iw \
--receiver res:googledrive.com,res_destination_url:"goodledrive.com/user/
folder/"  \
--data file:/root/test.png,destination_file_path:"Send file here"
```

Событие в Консоли Управления Traffic Monitor:



Пример 2:

Чтобы отправить в Traffic Monitor событие класса «**Обмен файлами**» сервиса «**Терминальная сессия**», используйте команду вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class fileexch \
--service terminal_session \
--evtattr event_name:"terminal_session event" \
--sender phone:+79057777777 \
```

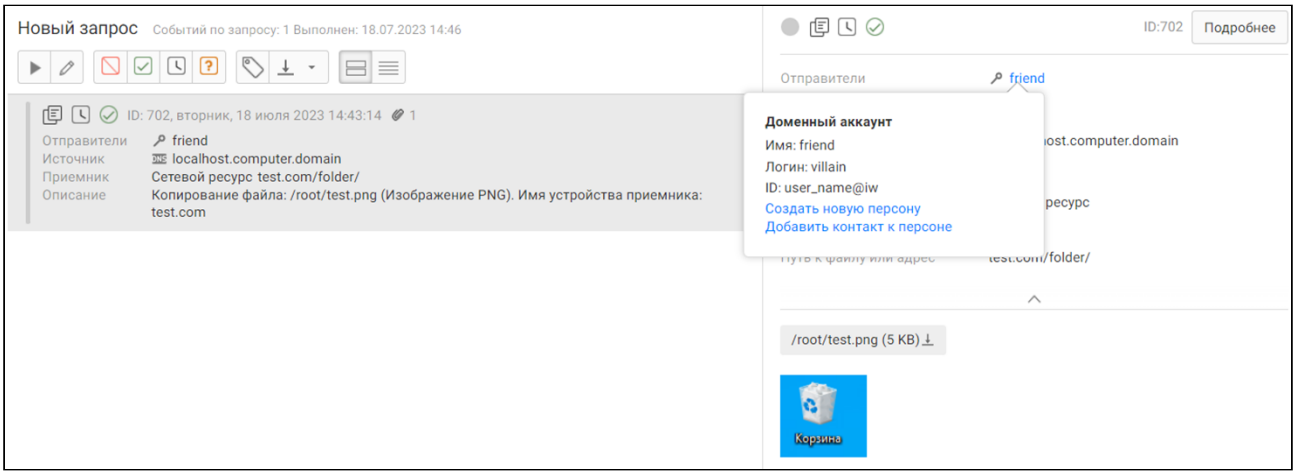
```
--receiver res:root@test.com \
--data file:/root/test.png,destination_file_path:"test.com/folder"
```

Пример 3:

Чтобы отправить в Traffic Monitor событие класса «**Обмен файлами**» сервиса «**Сетевые ресурсы**», используйте команду вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class fileexch \
--service network_resource \
--evtattr event_name:"network_resource event" \
--sender "auth:user_name@iw:'{\"display_name\":\"friend\",\"login\":
\"villain\"}'" \
--receiver res:test.com,res_destination_url:"test.com/folder/ "  \
--data file:/root/test.png,destination_file_path:"Send file here"
```

Событие в Консоли Управления Traffic Monitor:



Пример 4:

Чтобы отправить в Traffic Monitor событие класса «**Обмен файлами**» сервиса «**FTP**», используйте команду вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class fileexch \
--service ftp \
--evtattr event_name:"FTP Event" \
--sender login:villain \
--receiver res:ftp.site.org,res_destination_url:"catalog\catalog\res.file" \
--data file:/root/test.png,destination_file_path:"Send file here"
```

Пример 5:

Чтобы отправить в Traffic Monitor событие класса «**Обмен файлами**» сервиса «**Съемное устройство**», используйте команду вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
```

```
--id iw \
--class fileexch \
--service file_copy_removable \
--evtattr event_name:"Copy file on device" \
--receiver dev:"Removable device" \
--data file:/root/test.png,source_file_path:"Copy from
here",destination_file_path:"Send file here"
```

Пример 6:

Чтобы отправить в Traffic Monitor событие класса «**Обмен файлами**» для мессенджера ICQ, используйте команду вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class fileexch \
--service im_icq \
--evtattr event_name:"ICQ File Transfer Event" \
--receiver icq:IcqReceiver \
--data file:/root/test.png
```

Пример 7:

Чтобы отправить в Traffic Monitor событие класса «**Обмен файлами**» для мессенджера Skype, используйте команду вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class fileexch \
--service im_skype \
--evtattr event_name:"Skype File Transfer Event" \
--sender skype:SkypeSender \
--receiver skype:SkypeReceiver1 \
--receiver skype:SkypeReceiver2 \
--data file:/root/test.png,filename:test.png
```

Пример отправки события класса «Звонок»

Чтобы отправить в Traffic Monitor событие класса «**Звонок**» сервиса «**Мессенджеры**», используйте команду вида:

```
pushapi-util \
--host TM_host \
--token device_monitor_token_value \
--id iw \
--class voicetalk \
--service im_skype \
--evtattr event_name:"Skype Voice Talk" \
--sender skype:SkypeSender \
--receiver skype:SkypeReceiver \
--data file:/root/sound_msg.wav,call_duration:7
```

Событие содержит аудиофайл `sound_msg.wav`, а длительность разговора составляет 7 секунд. В Консоли Управления Traffic Monitor аудиофайл будет доступен для скачивания. Атрибуты `receiver`, `data` и `call_duration` обязательны для событий класса «**Звонок**» сервиса «**Мессенджеры**».

Примеры отправки событий класса «Запись мультимедиа»

Для отправки событий класса «**Запись мультимедиа**» нет встроенных сервисов, поэтому для этого необходимо зарегистрировать свой сервис и использовать команду следующего вида:

```
pushapi-util \
--host TM_host \
--token your_plugin_token_value \
--id your_company \
--class multimedia \
--service your_multimedia_service \
--evtattr event_name:"Camera Event" \
--receiver ws:"ws1.domain.org",workstation_type:"ws_type_computer" \
--data file:/root/video.avi
```

Сборка примеров под Linux pushAPI SDK

Для сборки примеров под Linux нужно иметь установленный пакет **cmake** и компилятор с поддержкой стандарта C++11. Все остальные зависимости определяются **cmake**-скриптом. Для сборки может понадобиться установка Thrift Runtime-библиотек, библиотеки Boost и OpenSSL, а именно:

- boost 1.62
- openssl 1.0.1
- thrift 0.11.0

Сборка примеров под Windows pushAPI SDK

Для сборки примеров под Windows нужно иметь установленную программу Visual Studio 2013. В состав примеров входит Toolchain, который содержит boost, OpenSSL, а также исходники Thift Runtime-библиотек, которые собираются в составе примера. В Toolchain входит утилита thrift, которая компилирует thift-схему в соответствующий исходный код. Пользователь может использовать свою реализацию boost, OpenSSL и Thrift – для этого нужно определить свои переменные окружения:
`$(BOOST_ROOT)`
`$(OPENSSL_ROOT_DIR)`
`$(THRIFT_INSTALL_DIR)`

> **Важно!**
>
> Самостоятельную сборку Thrift необходимо осуществлять с OpenSSL, т.к. pushAPI работает по SSL-соединению.

9.1.5 Диагностика ошибок при отправке событий

Проблемы с отправкой событий по Push API могут быть связаны с:

1. Неполадками в веб-интерфейсе, Чтобы включить режим отладки на сервере:
   a. Зайдите на сервер по SSH.
   b. Откройте на редактирование файл `/opt/iw/tm5/etc/web.conf .`
   c. Включите режим отладки, установив значение: `"debug": true .`
   d. Сохраните файл.
   e. Обновите страницу в веб-интерфейсе.

f. Перейдите в отладочную консоль с подробными логами по запросам к API. Для этого введите url: `https://<IP_сервера>/api/debug/default .`

2. Отсутствие доверенных сертификатов. Добавьте самоподписанные сертификаты в доверенные. Пути до сертификатов:
   - Веб: `/opt/iw/tm5/etc/certification/web-server.pem`
   - XAPI: `/opt/iw/tm5/etc/cert/server.pem`
3. Файловыми очередями. Чтобы посмотреть файловые очереди и очереди ошибок на сервере TM, перейдите в:
   - `opt/iw/tm5/queue`
   - `opt/iw/tm5/errors`
4. Файлами с ошибками экстракции. Они находятся в папке, указанной в параметре "Dreamcatcher" для каждого экстрактора в конфигурационном файле `/opt/iw/tm5/etc/extractors.conf`
5. Работой служб iw_xapi_xapi и iw_xapi_puppy. Чтобы получить информацию от логгеров разных программных компонентов:
   a. На сервере Traffic Monitor откройте на редактирование конфигурационный файл `/opt/iw/tm5/etc/xapi.conf`
   b. Установите следующие значения логгеров:
   ```
   "GlobalLevel": "debug",

         "Loggers":
   {

            "Filequeue": "debug",
            "RawMailDump": "fatal",
            "Root": "debug",

            "puppy": "debug",

            "thrift_handler": "warning"

         }
   ```
   где:
      - Filequeue - логирование действий файловой очереди.
      - RawMailDump - сырые данные почты
      - puppy - логгер Push API
      - thrift_handler - данные thrift-структур, поступивших по сети, для проверки трафика
      - Root - корневой логгер, отвечающий за чтение из БД, создание индексов, архивирование, удаление индексов
   c. Сохраните файл.
   d. Выполните команды :
   ```
   iwtm restsrt xapi_xapi
   iwtm restart xapi_puppy
   ```
   e. Отправьте событие в Traffic Monitor по Push API (подробнее см. "Примеры использования утилиты pushapi-util"). Лог-файлы будут сохранены в директории `/var/log/infowatch`

9.2 Загрузка эталонных документов и выгрузок из БД в Traffic Monitor (REST API SDK)

Данный программный интерфейс предназначен для интеграции сторонних компонентов с Traffic Monitor. Сторонние компоненты могут самостоятельно формировать документы, представляющие собой «Эталонные выгрузки баз данных» и «Эталонные документы» и далее, используя предоставленный интерфейс, передавать их в Traffic Monitor. После этого переданные выгрузки и эталонные документы будут детектироваться в анализируемых Traffic Monitor потоках данных. Правила детектирования выгрузок и ЭД задаются в консоли Traffic Monitor. Сторонний компонент только загружает или обновляет содержимое выгрузки или ЭД. Сторонний компонент должен быть зарегистрирован в системе согласно процедуре, которая описана в разделе «Регистрация стороннего компонента REST API».

Документ поможет реализовать механизм автоматической загрузки эталонных документов и выгрузок из БД средствами API. Документ предназначен для разработчиков - сотрудников организации, ее технологических партнеров или подрядных организаций.

9.2.1 Конфигурирование программного интерфейса REST API

В конфигурировании программного интерфейса участвуют следующие параметры:

- Адрес подключения с сервису SDK (сетевой адрес, порт);
- Токен авторизации. Строка символов, которая позволяет авторизовать компоненты, использующие SDK. Данная строка может быть получена в разделе "Управление" (вкладка "Токены") в консоли управления Traffic Monitor;
- Идентификатор передаваемых данных. Строка, которую производитель внешнего компонента получает от InfoWatch. Данная строка характеризует тип передаваемых данных и используется в политике лицензирования;
- Идентификатор компании-производителя внешнего компонента. Строка, которую производитель внешнего компонента получает от InfoWatch. Данная строка характеризует компанию-производителя и используется в политике лицензирования.

9.2.2 Общее описание программного интерфейса REST API

Программный интерфейс представляет собой REST API, т.е. последовательность запросов к Web-серверу. Программный интерфейс предоставляет следующие возможности:

- Создание эталонной выгрузки REST API;
- Обновление эталонной выгрузки REST API;
- Добавление содержимого к эталонной выгрузке REST API;
- Получение состояния эталонной выгрузки REST API;
- Инициирование распространения нового содержимого загруженных эталонных документов на модули анализа REST API;
- Получение версии интерфейса, который поддерживается Traffic Monitor REST API;
- Создание каталога эталонных документов;
- Удаление каталога эталонных документов;
- Создание эталонного документа;
- Удаление эталонного документа;
- Добавление содержимого эталонного документа;
- Получение состояния эталонного документа.

Создание эталонной выгрузки REST API

```
FingerPrintDsc Create(string name, string[] columns)
```

Функция создает эталонную выгрузку с заданным именем и списком имён колонок эталонной выгрузки. Возвращается описание созданного эталонного объекта (FingerPrint).

Обновление эталонной выгрузки REST API

```
FingerPrintDsc UpdateContent(string FingerPrintId,string[]content)
```

Функция позволяет загрузить в существующий эталонный объект (выгрузку) его содержимое. Эталонный объект определяется его идентификатором (поле структуры FingerPrintDsc). Возвращается описание эталонного объекта (FingerPrint). После загрузки содержимого начинается процесс преобразования и сохранения содержимого эталонного объекта. Данная функция целиком заменяет содержимое эталонного объекта.

Добавление содержимого к эталонной выгрузке REST API

```
FingerPrintDsc AddContent(string FingerPrintId, string[] content)
```

Функция позволяет добавить в существующий эталонный объект (выгрузку) дополнительное содержимое. Эталонный объект определяется его идентификатором (поле структуры FingerPrintDsc). Возвращается описание эталонного объекта (FingerPrint). После загрузки содержимого начинается процесс преобразования и сохранения содержимого эталонного объекта. Данная функция целиком заменят содержимое эталонного объекта.

Получение состояния эталонной выгрузки REST API

```
FingerPrintDsc GetInfo(string FingerPrintId)
```

Функция позволяет получить информацию о существующей эталонной выгрузке. Возвращается описание эталонного объекта (FingerPrint). Важным полем результата вызова этой функции является поле статуса эталонной выгрузки.

Инициирование распространения нового содержимого загруженных эталонных документов на модули анализа REST API

```
Void Ditribute()
```

Функция инициирует процедуру распространения нового содержимого эталонных выгрузок по модулям анализа. Функция синхронная. После её выполнения система Traffic Monitor готова к детектированию нового содержимого в анализируемых потоках данных.

Получение версии интерфейса, который поддерживается Traffic Monitor REST API

```
VersionInfo GetInterfaceVersion()
```

Функция позволяет получить информацию об интерфейсе, который поддерживает Traffic Monitor.

Каждая часть выгрузки, которая загружается в Traffic Monitor, проходит этапы преобразования во внутренний формат и сохранения в базе данных. После этого сторонний компонент должен дать команду о передаче новой выгрузки системам анализа Traffic Monitor. Таким образом можно загрузить по частям достаточно большую выгрузку. Т.е. выгрузка загружается в несколько запросов, а затем одной командой распространяется по системам анализа.

Каждый этап обработки загруженной эталонной выгрузки занимает некоторое время. На каждом этапе могут произойти ошибки.

Программный интерфейс позволяет отслеживать этапы обработки выгрузки и реагировать на ошибки, которые могут произойти во время обработки.

Создание каталога эталонных документов

```
CatalogId CreateEtDocCatalog(CatalogId parentCatalog, string name);
```

Функция создания каталога эталонных документов, возвращающая идентификатор созданного каталога, где:

- `parentCatalog` – идентификатор родительского каталога, в котором создаётся каталог (0 - создать в корневом каталоге);
- `name` - имя каталога эталонных документов. Под этим именем каталог появится в консоли TM.

```
CatalogId GetEtDocCtalogIdByName(string name);
```

Функция получения идентификатора каталога по имени, возвращающая идентификатор каталога.

```
CatalogList GetEtDocCtalogs();
```

Функция получения каталогов, созданных плагином, возвращающая список каталогов.

Удаление каталога эталонных документов

```
void DeleteEtDocCatalog(CatalogId id)
```

Функция удаления каталога эталонных документов в том случае, если он не входит в Объект защиты. Функция возвращает идентификатор удаляемого каталога.

Создание эталонного документа

```
DocumentId CreateEtDoc(CatalogId parentCatalog, string name, Type type);
```

Функция создания эталонного документа, возвращающая идентификатор эталонного документа, где:

- `parentCatalog` - идентификатор каталога, в котором создаётся эталонный документ. Не может быть "0", так как создавать документы в корневом каталоге запрещено;
- `name` - имя эталонного документа. Под этим именем документ появится в консоли TM;
- `type` - тип эталонного документа. Эталонный документ может быть создан на основании извлечённого текста или своего бинарного представления.

Удаление эталонного документа

```
void DeleteEtDoc(DocumentId doc);
```

Функция удаления эталонного документа в том случае, если он не входит в Объект защиты.

Добавление содержимого эталонного документа

```
void AddEtDocContent(DocumentId doc, Buffer buf);
```

Функция добавления содержимого эталонного документа, которая позволяет также загружать содержимое по частям, где:

- `doc` - идентификатор эталонного документа, к которому добавляется содержимое;
- `buf` - буфер данных, добавляемый к документу.

```
void UpdateEtDocContent(DocumentId doc, Buffer buf);
```

Функция обновления или замены содержимого эталонного документа, где:

- `doc` - идентификатор эталонного документа, в котором меняется содержимое;
- `buf` - буфер данных, который будет представлять собой новое содержимое эталонного документа.

Получение состояния эталонного документа

```
DocState GetEtDocState(DocumentId doc);
```

Функция получения состояния эталонного документа, где:

- `doc` - идентификатор эталонного документа.

9.2.3 Регистрация стороннего компонента REST API

Для интеграции стороннего модуля перехвата данных с системой Traffic Monitor нужно выполнить следующие шаги:

1. Зарегистрировать в компании InfoWatch строку-идентификатор компании-производителя компонента. Это позволит связывать компанию-производителя с лицензией, которая будет устанавливаться в систему Traffic Monitor.
2. Получить лицензию разработчика, связанную с идентификатором компании. Это позволит загружать в Traffic Monitor эталонные выгрузки и эталонные документы указанного производителя. Лицензия разработчика позволяет загружать произвольные эталонные выгрузки и эталонные документы.
3. Создать файл регистрации стороннего компонента в описанном ниже формате. К этому моменту должны быть определены названия источника данных эталонной выгрузки и эталонного документа, которая будет формироваться сторонним компонентом. В файл регистрации добавляется лицензия, полученная на шаге 2.
4. Загрузить файл регистрации стороннего компонента в систему Traffic Monitor через интерфейс консоли Traffic Monitor. Раздел «Управление» - «Плагины». (см. "InfoWatch Traffic Monitor. Руководство пользователя").
5. Получить токен доступа в свойствах загруженного стороннего компонента.

Система готова принимать сформированные эталонные выгрузки и эталонные документы от заданной компании-производителя с указанным источником эталонной выгрузки или эталонного документа.

Формат файла регистрации стороннего компонента REST API

Плагин представляет собой архив в формате .zip. В состав архива входят:

- папка **licenses**, содержащая файлы лицензий;
- папка **icon**, содержащая файлы с используемыми пиктограммами для регистрируемых событий;
- файл **manifest.json**, содержащий информацию о плагине.

Примечание:

Папки **licenses** и **icon** не являются обязательными. Файлы лицензий и файлы с используемыми пиктограммами могут находиться в корне.

Для внешних систем-источников событий файл **manifest.json** должен содержать следующую информацию:

Содержимое	Тип данных	Является обязательным	Описание
`{`			
`"PLUGIN_ID": "",`	Строка	Да	Уникальный идентификатор (UUID) плагина. До 40 символов. **Пример:** `"PLUGIN_ID": "189C38D390396EB6E0530100007F1CA200000001",`
`"DISPLAY_NAME": "",`	Строка	Да	Отображаемое имя плагина. **Пример:** `"DISPLAY_NAME": "Имя плагина",`
`"DESCRIPTION": "",`	Строка	Нет	Отображаемое описание плагина. **Пример:** `"DESCRIPTION": "Тестовый плагин для демонстрации",`
`"VERSION": "",`	Строка	Да	Версия плагина. **Пример:** `"VERSION": "1.0.0",`
`"VENDOR": "",`	Строка	Да	Идентификатор компании-разработчика, соответствующий названию компании в лицензии. **Пример:** `"VENDOR": "infowatch",`
`"LICENSE": [ { "PATH":"" } ],`	Массив Строка	Да	Файлы лицензии. Должны быть указаны пути к файлам лицензий относительно корня архива. **Пример:** `"LICENSE": [ { "PATH":"licenses/файл лицензии 1.license" }, {`

Содержимое	Тип данных	Является обязательным	Описание
			`"PATH":"licenses/файл лицензии 2.license" }, ],`
`"PATTERN_SEARCH_LICENSE": "",`	Строка	Нет	Шаблон для поиска загруженных ранее лицензий для привязки их к плагину. Имеет формат: `{operator:or|and,child[{name:value}]}"` . **Пример:** `"PATTERN_SEARCH_LICENSE": "{\"operator\":\"or\", \"conditions\": [{\"origin\":\"dm\"}, {\"origin\": \"dmmobile\"}]}",`
`"DATATYPE": [ { "VALUE": "" } ]`	Массив Строка	Да	Список типов данных с указанием систем-источников. **Пример:** `"DATATYPE": [ { "VALUE": "type1_s1", "VALUE": "type2_s2" } ],`
`}`			

Важно!

В состав файла входит просроченная лицензия. Для регистрации плагина на основании файла примера нужно будет получать новую лицензию.

9.2.4 Формат эталонной выгрузки и эталонного документа REST API

Эталонная выгрузка представляет собой последовательность строк в кодировке UTF-8. Строки должны быть разделены символами CRLF или LF. Значения ячеек в строке должны быть разделены символами TAB или ‘,’(Comma). Данные ограничения соответствуют формату TSV (Tab separated values) или CSV (Comma separated values). Число ячеек во всех строках должно совпадать. При создании выгрузки указывается число ячеек. Оно равно числу имён столбцов, которые передаются в функцию Create. В существующую эталонную выгрузку можно загружать только содержимое, где число ячеек в строке

равно числу, которое было указано при создании. Любое другое число ячеек сделает выгрузку невалидной.

Эталонный документ представляет собой файл любого формата. Если файл является незашифрованным архивом, Система извлекает файлы из архива, обрабатывает их независимо друг от друга и создает отдельный эталонный документ на основе каждого извлеченного файла (эталонный документ на основе всего архива не создается). Все созданные эталонные документы помещаются в каталог, выбранный при создании эталонного документа. Если файл является зашифрованным архивом, Система не создает эталонный документ. Если файл является файлом типа "контейнер", Система создает один эталонный документа на основе файла.

9.2.5 Функциональное описание программного интерфейса REST API

Программный интерфейс представляет собой REST API, т.е. последовательность GET/POST-запросов к Web-серверу на определенный ресурс и ответов на них. Формат оформления запросов и формат ответов представлен ниже. Защита данных во время их передачи осуществляется по протоколу SSL/TLS без авторизации клиента. Передача данных в открытом виде программным интерфейсом не поддерживается. При передаче данных используется кодировка UTF-8.

Общие заголовки GET/POST-запросов

Каждый запрос к серверу должен содержать следующие обязательные заголовки:

- X-API-Auth-Token – указывается токен авторизации;
- X-API-DataType – указывается идентификатор передаваемого типа данных;
- X-API-CompanyId – указывается идентификатор компании-производителя;
- X-API-Version – указывается номер версии программного интерфейса. Сейчас используется номер: «1».

Работа с конфигурацией REST API

После того, как содержимое эталонной выгрузки передано на сервер SDK и удачно скомпилировано, можно дать команду на распространение изменённой конфигурации на системы анализа.

Распространение конфигурации REST API

Описание: метод нужен для распространения конфигурации на системы анализа.

Характеристики метода: POST, синхронный.

Ресурс: xapi/configuration..

JSON Scheme тела запроса:

```
{
    "STATUS": "distribute"
}
```

В случае успеха – код возврата «200» и тело ответа:

```
{
    "STATUS": "distributed"
}
```

В случае ошибки – код возврата «500». Возможные значение поля code:

- error – конфигурация не может быть применена.

Работа с эталонными выгрузками REST API

Глава содержит следующую информацию:

- Создание выгрузки REST API;
- Обновление данных выгрузки REST API;
- Добавление данных выгрузки REST API;
- Получение состояния выгрузки REST API;
- Получение текущей версии REST API.

Создание выгрузки REST API

Описание: метод нужен для добавления новой выгрузки без наполнения её данными. Сами данные будут добавлены через отдельный метод.

Характеристики метода: POST, синхронный.

Ресурс: `xapi/etalonTable`.

JSON Scheme тела запроса:

```
{
    "properties": {
        "DISPLAY_NAME": {
            "description": "Название выгрузки",
            "type": "string",
            "errors": ["required", "not_match", "not_unique_field"]
        },
        "CONDITION_COLUMNS": {
            "description": "строка, закодированный JSON-массив с названием столбцов, например: [\"first_column\", \"second_column\"]",
            "type": "string",
            "errors": ["required", "not_valid_json"]
        },
        "NOTE": {
            "description": "Произвольное описание выгрузки",
            "type": "string"
        }
    },
    "required": [
        "DISPLAY_NAME",
        "CONDITION_COLUMNS"
    ]
}
```

В случае успеха – код возврата «200» и тело ответа:

```
{
    "FINGERPRINT_ID":    "string",
    "TYPE":              "string",
    "SOURCE":            "string",
    "DISPLAY_NAME":      "string",
    "NOTE":              "string",
    "CONDITION_COLUMNS": "string",
    "CREATE_DATE":       "string",
    "CHANGE_DATE":       "string",
    "STATUS":            "string"
}
```

Важно!

Число столбцов, которые указываются при создании выгрузки, не может быть изменено. При загрузке содержимого число ячеек в загружаемых стоках должно совпадать с тем значением, которое было указано при создании эталонной выгрузки.

Обновление данных выгрузки REST API

Описание: метод предназначен для обновления содержимого выгрузки. Все записи эталонной выгрузки будут удалены, и после этого добавится новая запись. Данный метод можно использовать для первичного наполнения.

Важно!

Наполнение должно иметь целое число строк, минимум 2 строки.
Минимальное количество колонок – 2.
Минимальный байтовый размер загружаемого контента – 128 байт.
Максимальный байтовый размер загружаемого контента – 2 ГБ.

Характеристики метода: POST, асинхронный.

Ресурс: `xapi/etalonTable/{id}/content/replace`.

Параметры:

- id – fingerprint_id редактируемого объекта.

Например: `enatalonTable/0123456789ABCDEF/content/replce`

Content-type тела запроса должно быть не `application/json`, а `multipart/form-data` (подробнее см.: https://ru.wikipedia.org/wiki/Multipart/form-data).

Content-Disposition, который будет содержать файл, должен иметь поле `name` со значением CONTENT и атрибут `filename` с названием файла с расширением TSV или CSV в зависимости от формата файла, разделенного Tab или запятыми соответственно.

Возможные значение ошибок поля CONTENT:

- not_valid – файл имеет неверную сигнатуру;
- not_allowed_file_extension – неразрешенное расширение файла, на данный момент разрешены 2 формата: TSV и CSV;
- too_long_size – слишком большой размер файла;
- duplicate – такой файл уже существует в системе.

В случае успеха – код возврата «200» и тело ответа:

```
{
    "FINGERPRINT_ID":    "string",
    "TYPE":              "string",
    "SOURCE":            "string",
    "DISPLAY_NAME":      "string",
    "NOTE":              "string",
    "CONDITION_COLUMNS": "string",
    "CREATE_DATE":       "string",
    "CHANGE_DATE":       "string",
    "STATUS":            "string"
}
```

После получения кода возврата «200» нужно взять поле FINGERPRINT_ID и по нему проверять состояние выгрузки, пока статус не перейдет в состояние ready.

Важно!

Число ячеек в загружаемых строках должно совпадать со значением числа столбцов, которое было указано при создании выгрузки.
При удалении выгрузки через консоль TrafficMonitor она считается существующей до момента распространения конфигурации, т.е. с этой выгрузкой возможны операции обновления и добавления содержимого.
При загрузке «ошибочного» содержимого, т.е. содержимого, которое является дубликатом уже загруженной выгрузки (части выгрузки) или содержимого, строки которого содержат неправильное количество ячеек, весь эталонный объект получает невалидный статус.
Внешняя подсистема должна перестать загружать такое содержимое и перейти к следующему содержимому. Загрузка корректного содержимого исправляет общий статус эталонного объекта. Некорректное содержимое никогда не попадает на модули анализа.

Добавление данных выгрузки REST API

Описание: метод предназначен для добавления содержимого в выгрузку. Все записи содержимого будут сохранены, будет добавлена новая запись.

Характеристики метод: POST.

Ресурс: `xapi/etalonTable/{id}/content`.

Подробнее см. статью "Обновление данных выгрузки REST API"

Получение состояния выгрузки REST API

Описание: метод нужен для получения сведений о состоянии одной выгрузки.

Характеристики метода: GET.

Ресурс: `xapi/etalonTable/{id}`.

Параметры:

- id – fingerprint_id редактируемого объекта.

Например: `etalonTable/0123456789ABCDEF`

Тело ответа:

```
{
    "FINGERPRINT_ID":    "string",
    "TYPE":              "string",
    "SOURCE":            "string",
    "DISPLAY_NAME":      "string",
    "NOTE":              "string",
    "CONDITION_COLUMNS": "string",
    "CREATE_DATE":       "string",
    "CHANGE_DATE":       "string",
    "STATUS":            "string"
    "content":           [
        {
            "CONTENT_ID":   "string",
            "CHECKSUM":     "string",
            "CONTENT_SIZE": "integer",
            "CONTENT_MIME": "string"
        }
    ]
}
```

Поле статус может иметь следующие значения:

- sending – отправка данных на компилятор;
- sent – данные отправлены на компилятор, и начинается компиляция;
- compiling – идёт компиляция документа;
- compiled – документ готов;
- saving – идет сохранение;
- ready – документ сохранён, готов для дальнейшего внесения в него изменений и не блокирует применение конфигурации;
- fatal_error – произошла ошибка во время сохранения документа, и операция не может быть повторена или продолжена;
- duplicate – произошло дублирование данных content (можно продолжать загружать содержимое);

- not_match_delimiter – строки в выгрузке не содержат разделителя символов (COMMA или TAB);
- wrong_column_count – количество колонок в строке не совпадает с количеством колонок, которое было указано при создании выгрузки;
- error – временная ошибка, можно попробовать ещё раз через некоторое время (лучше через час);
- max_words - превышено допустимое количество слов;
- max_columns - превышено допустимое количество столбцов.

Получение текущей версии REST API

Описание: метод нужен для получения поддерживаемых версий.

Если код возврата будет «200», то можно продолжать работу.

Метод может использоваться для проверки лицензии, версии и авторизации. В случае успеха – код возврата «200». В случае неисправности возвращается код ошибки.

Характеристики метод: GET.

Ресурс: `xapi/version`.

Тело ответа:

```
{
    "data":       [
        {
            "NAME":    "string",
            "STATUS": "supported|deprecated|removed"
        }
    ]
}
```

Пример ответа при указании неправильной версии:

HTTP/1.1 400 Bad Request

Content-Type: application/json

```
{
    "error": {
        "code": "unsupported_version",
        "meta": {
            "versions": [
                {
                    "NAME":   "1.0.0",
                    "STATUS": "removed"
                },
                {
                    "NAME":   "2.0.0",
                    "STATUS": "deprecated"
```

```
                },
                {
                    "NAME":   "3.0.0",
                    "STATUS": "supported"
                }
            ]
        }
    }
}
```

Общие ошибки REST API

Описание возможных ошибок:

- not_unique_field – неуникальное значение;
- unique – уникальное значение;
- too_short – слишком короткое значение;
- too_long – слишком длинное значение;
- wrong_length – значение неправильной длины;
- repeat_exactly – значение не совпадает со значением в другом поле;
- required – у поля обязательно должно быть значение;
- not_exist_in_list – значение не входит в список;
- invalid – значение невалидно;
- contains_child – узел является не листовым;
- contains_items – узел категории содержит сущности;
- parent_own_child – родитель содержит детей;
- parent_self_set_current – попытка установить родителем самого себя;
- not_valid_json – JSON невалиден;
- invalid_cron_format – не правильный формат cron;
- read_only – объект доступен только для чтения;
- empty – значение не может быть пустым.

Успешный ответ REST API

Код возврата: «200».

Формат ответа:

```
{
    "data": "mixed",
    "meta": "mixed"
}
```

Ошибка валидации данных/версии REST API

Код возврата: «400».

Формат ответа в случае неправильной версии API (JSON Scheme) REST API

```
{
```

```
    "properties": {
        "error": {
            "description": "Контейнер ошибки",
            "properties":  {
                "code":       {
                    "description": "Общее описание ошибки",
                    "type":        "string",
                    "value":       "unsupported_version"
                },
                "meta": {
                    "description": "Дополнительные данные об ошибке",
                    "type":        "object",
                    "properties":  {
                        "supported_versions": {
                            "description": "Массив поддерживаемых версий",
                            "type":        "array",
                            "items": {
                                "description": "Поддерживаемая версия",
                                "type": "string"
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Формат ответа в случае ошибки валидации запроса (JSON Scheme) REST API

{

```
    "properties": {
        "error": {
            "description": "Контейнер ошибки",
            "properties":  {
                "code":       {
                    "description": "Общее описание ошибки",
                    "type":        "string",
                    "value":       "validation"
```

```
                },
                "meta": {
                    "description": "Дополнительные данные об ошибке",
                    "type":        "object",
                    "properties":  {
                        "validation": {
                            "description": "Название полей модели",
                            "type":        "object",
                            "properties":  {
                                "field":     {
                                    "description": "Поле модели",
                                    "type":        "array",
                                    "items":       {
                                        "description": "Ошибка",
                                        "type":        "string",
                                        "values": [
                                            "not_unique_field",
                                            "too_short",
                                            "too_long",
                                            "wrong_length",
                                            "repeat_exactly",
                                            "required",
                                            "not_exist_in_list",
                                            "invalid",
                                            "contains_child",
                                            "contains_items",
                                            "parent_own_child",
                                            "unique",
                                            "not_valid_json",
                                            "invalid_cron_format",
                                            "read_only",
                                            "parent_self_set_current",
                                            "empty"
                                        ]
                                    }
                                },
                                "sub_model": {
```

```
                                    "description": "Название вложенной модели",
                                    "type":        "object",
                                    "$ref":       "#/properties/error/
properties/meta/properties/validation"
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Пример ошибки при указании неправильной версии REST API

```
{
    "error": {
        "code": "unsupported_version",
        "meta": {
            "supported_versions": ["2", "3"]
        }
    }
}
```

Пример ошибки проверки на уникальность имени эталонной выгрузки REST API

```
{
    "error": {
        "code": "validation",
        "meta": {
            "validation":    {
                "DISPLAY_NAME": ["unique"]
            },
            "unique_models": [
                {
                    "FINGERPRINT_ID": "0123456789ABCDEF",
                    "DISPLAY_NAME":   "test"
                }
            ]
        }
```

```
    }
}
```

Необходима авторизация REST API

Код возврата: «401».

Описание: возвращается в любом запросе, где требуется авторизация.

Тело ответа: нет.

Недостаточно прав REST API

Код возврата: «403».

Описание: возвращается в любом запросе, если у текущего пользователя недостаточно прав для выполнения данной операции.

Тело ответа: нет.

Сущность в системе не найдена REST API

Код возврата: «404».

Описание: возвращается, если определенная модель не найдена в системе. Например, не найдена эталонная выгрузка.

Тело ответа: нет.

Тело запроса слишком большое REST API

Код возврата: «413».

Описание: возвращается, если передан слишком большой запрос.

Тело ответа: нет.

Превышено количество запросов к серверу REST API

Код возврата: «429».

Описание: возвращается, если на сервер отправлено слишком много запросов за определенный промежуток времени. Может сопровождаться заголовком Retry-After, указывающим, через какое время можно повторить запрос.

Тело ответа (JSON Scheme):

```
{
    "properties": {
        "error": {
            "description": "Контейнер ошибки",
            "properties":  {
                "code": {
                    "description": "Общее описание ошибки",
                    "type":        "string",
                    "value":       "too_many_requests "
```

```
                },
                "meta": {
                    "description": "Дополнительные данные об ошибки",
                    "type":        "object",
                    "properties":  {
                        "time": {
                            "description": "HTTP-дата, либо целое число в секундах, через которые можно повторить запрос",
                            "type":        "string"
                        }
                    }
                }
            }
        }
    }
}
```

Ошибка системы REST API

Код возврата: «500».

Описание: возвращается, если в системе произошла логическая ошибка, и далее невозможно выполнение запроса.

Тело ответа (JSON Scheme):

```
{
    "properties": {
        "error": {
            "description": "Контейнер ошибки",
            "properties":  {
                "code": {
                    "description": "Общее описание ошибки",
                    "type":        "string"
                },
                "meta": {
                    "description": "Дополнительные данные об ошибке",
                    "type":        "object"
                }
            }
        }
```

```
    }
}
```

Сервер временно не доступен REST API

Код возврата: «503».

Описание: информирует, что в данный момент невозможно выполнить запросы, и сервер находится на техническом обслуживании. Приблизительное время окончания технического обслуживания содержится в заголовке Retry-After (значением этого заголовка может быть либо HTTP-дата, либо целое число в секундах).

Тело ответа (JSON Scheme):

```
{
    "properties": {
        "error": {
            "description": " Контейнер  ошибки ",
            "properties":  {
                "code": {
                    "description": " Общее  описание  ошибки ",
                    "type":        "string",
                    "value":       "maintains"
                },
                "meta": {
                    "description": " Дополнительные  данные  об  ошибке ",
                     "type":        "object",
                    "properties":  {
                        "time": {
                            "description": "HTTP-дата, либо цело число в секундах, через которые можно повторить запрос",
                            "type":        "string"
                        }
                    }
                }
            }
        }
    }
}
```

Работа с эталонными документами REST API

Глава содержит следующую информацию:

- Создание каталога эталонных документов REST API;

- Просмотр каталогов эталонных документов REST API;
- Редактирование каталога эталонных документов REST API;
- Удаление каталога эталонных документов REST API;
- Создание эталонного документа REST API;
- Просмотр эталонных документов REST API;
- Обновление и замена содержимого эталонного документа REST API;
- Удаление эталонного документа REST API.

Создание каталога эталонных документов REST API

Метод: `POST`

Ресурс: `etalonDocumentCategory`

Тело запроса:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {},
    "id": "http://infowatch.com/tm/etalonDocumentCategory/edit",
    "type": "object",
    "properties": {
        "PARENT_CATEGORY_ID": {
            "examples": [
                "AC3DF073B60107890A4C704A10577FE700000000"
            ],
            "title": "GUID категории родителя, если null, то проставляется папка
автоматических эталонных документов",
            "type": "string"
        },
        "DISPLAY_NAME": {
            "title": "Название",
            "type": "string"
        },
        "DIR_PATH (возможно изменение в названии)": {
            "title": "Полное имя директории",
            "type": "string"
        },
        "FP_BIN_VALUE_THRESHOLD": {
            "title": "Порог цитируемости бинарных данных",
            "type": "integer"
        },
        "FP_TEXT_VALUE_THRESHOLD": {
            "title": "Порог цитируемости текстовых данных",
            "type": "integer"
        },
        "NOTE": {
            "title": "Описание",
            "type": [
                "null",
                "string"
            ]
        }
    }
```

```
}
```

Тело ответа :

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {},
    "id": "http://infowatch.com/tm/etalonDocumentCategory",
    "type": "object",
    "properties": {
        "CATEGORY_ID": {
            "examples": [
                "AC3DF073B60107890A4C704A10577FE700000000"
            ],
            "title": "GUID категории",
            "type": "string"
        },
        "PARENT_CATEGORY_ID": {
            "examples": [
                "AC3DF073B60107890A4C704A10577FE700000000"
            ],
            "title": "GUID категории родителя",
            "type": "string"
        },
        "DISPLAY_NAME": {
            "title": "Название",
            "type": "string"
        },
        "DIR_PATH (возможно изменение в названии)": {
            "title": "Полное имя директории",
            "type": "string"
        },
        "FP_BIN_VALUE_THRESHOLD": {
            "title": "Порог цитируемости бинарных данных",
            "type": "integer"
        },
        "FP_TEXT_VALUE_THRESHOLD": {
            "title": "Порог цитируемости текстовых данных",
            "type": "integer"
        },
        "NOTE": {
            "title": "Описание",
            "type": [
                "null",
                "string"
            ]
        },
        "OWNER": {
            "title": "ID токена",
            "type": "integer"
        },
        "CREATE_DATE": {
            "examples": [
```

```
                "2017-07-04 10:45:45.000000"
            ],
            "title": "Дата создания",
            "type": "string"
        },
        "CHANGE_DATE": {
            "examples": [
                "2017-07-04 10:45:45.000000"
            ],
            "title": "Дата изменения",
            "type": "string"
        }
    }
}
```

Просмотр каталогов эталонных документов REST API

Метод: `GET`

Ресурс: `etalonDocumentCategory[ /{category_id}]`

Параметры:

- `filter` — список фильтрации:
  - `create_date[from]` — начало диапазона для фильтрации по дате создания в формате UNIX-timestamp (включая передаваемую дату, можно использовать отдельно от `create_date[to]`)
  - `create_date[to]` — окончание диапазона для фильтрации по дате создания в формате UNIX-timestamp (включая передаваемую дату, можно использовать отдельно от `create_date[from]`)
  - `change_date[from]` — начало диапазона для фильтрации по дате изменения в формате UNIX-timestamp (включая передаваемую дату, можно использовать отдельно от `change_date[to]`)
  - `change_date[to]` — окончание диапазона для фильтрации по дате изменения в формате UNIX-timestamp (включая передаваемую дату, можно использовать отдельно от `change_date[from]`)
  - `display_name[]` — имя каталога, можно использовать * в конце, чтобы включить поиск по `LIKE` , например: `?filter[display_name][]=foo*`
  - `dir_path[]` — полный путь до каталога, можно использовать * в конце, чтобы включить поиск по `LIKE` , например: `?filter[dir_path][]=foo*`
  - `fingerprint_id []` — GUID эталонного документа, например: `?filter[ fingerprint_id ][]= 82EAE496A1686E407B1162E3C3159999F404F559&filter[fingerprint_id][]=AF8EAC3C3C8532C3C780481B7D8C5B8E68D0148F`
- `with[]` — список дополнительных сущностей, которые должны быть добавлены к объекту:
  - `etalonDocumentsCount` — количество эталонных документов в каталоге
- `sort[]` — поле, позволяющее сортировать результат. Возможные ключи:
  - `create_date` — `desc/asc` , например: `?sort[create_date]=desc`
  - `change_date` — `desc/asc` , например: `?sort[change_date]=asc`

Ответ:

Массив объектов, отдаваемых при создании/редактировании

Примеры :

```
GET /xapi/etalonDocumentCategory?filter[create_date][to]=1&filter[display_name]
[]=foo*&filter[display_name][]=bar&with[]=etalonDocumentsCount&sort[create_date]=desc
GET /xapi/etalonDocumentCategory/FAB85F61531BB9E428088EFC81F266FA3C2959E6
```

Редактирование каталога эталонных документов REST API

Метод: PUT

Ресурс: `etalonDocumentCategory/{category_id}`

Тело запроса :

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {},
    "id": "http://infowatch.com/tm/etalonDocumentCategory/edit",
    "type": "object",
    "properties": {
        "PARENT_CATEGORY_ID": {
            "examples": [
                "AC3DF073B60107890A4C704A10577FE700000000"
            ],
            "title": "GUID категории родителя, если null, то проставляется папка
автоматических эталонных документов",
            "type": "string"
        },
        "DISPLAY_NAME": {
            "title": "Название",
            "type": "string"
        },
        "DIR_PATH (возможно изменение в названии)": {
            "title": "Полное имя директории",
            "type": "string"
        },
        "FP_BIN_VALUE_THRESHOLD": {
            "title": "Порог цитируемости бинарных данных",
            "type": "integer"
        },
        "FP_TEXT_VALUE_THRESHOLD": {
            "title": "Порог цитируемости текстовых данных",
            "type": "integer"
        },
        "NOTE": {
            "title": "Описание",
            "type": [
                "null",
                "string"
            ]
        }
    }
}
```

Тело ответа:


```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {},
    "id": "http://infowatch.com/tm/etalonDocumentCategory",
    "type": "object",
    "properties": {
        "CATEGORY_ID": {
            "examples": [
                "AC3DF073B60107890A4C704A10577FE700000000"
            ],
            "title": "GUID категории",
            "type": "string"
        },
        "PARENT_CATEGORY_ID": {
            "examples": [
                "AC3DF073B60107890A4C704A10577FE700000000"
            ],
            "title": "GUID категории родителя",
            "type": "string"
        },
        "DISPLAY_NAME": {
            "title": "Название",
            "type": "string"
        },
        "DIR_PATH (возможно изменение в названии)": {
            "title": "Полное имя директории",
            "type": "string"
        },
        "FP_BIN_VALUE_THRESHOLD": {
            "title": "Порог цитируемости бинарных данных",
            "type": "integer"
        },
        "FP_TEXT_VALUE_THRESHOLD": {
            "title": "Порог цитируемости текстовых данных",
            "type": "integer"
        },
        "NOTE": {
            "title": "Описание",
            "type": [
                "null",
                "string"
            ]
        },
        "OWNER": {
            "title": "ID токена",
            "type": "integer"
        },
        "CREATE_DATE": {
            "examples": [
                "2017-07-04 10:45:45.000000"
            ],
```

```
            "title": "Дата создания",
            "type": "string"
        },
        "CHANGE_DATE": {
            "examples": [
                "2017-07-04 10:45:45.000000"
            ],
            "title": "Дата изменения",
            "type": "string"
        }
    }
}
```

Пример редактирования каталога CURL

```
curl 'https://example.com/xapi/etalonDocumentCategory/0123456789ABCDEF123400000000' \
  -X PUT \
  -H 'X-API-Auth-Token: 1792jmclf7fer1ikuhby' \
  -H 'X-API-Version: 1' \
  -H 'X-API-CompanyId: IW' \
  -H 'X-API-DataType: sap' \
  -H 'Content-Type: application/json' \
  --data-
binary
'{"PARENT_CATEGORY_ID":"0123456789ABCDEF123400000000","DISPLAY_NAME":"world.txt","DIR_
PATH":"c:\\hello\
\world.txt","FP_BIN_VALUE_THRESHOLD":10,"FP_TEXT_VALUE_THRESHOLD":10,"NOTE":null}'
```

Удаление каталога эталонных документов REST API

Метод: `DELETE`
Ресурс: `etalonDocumentCategory /{category_id}`

Создание эталонного документа REST API

Предполагается, что сначала создаётся эталонный документ, с помощью `POST /xapi/ etalonDocument` со статусом new, а дальше происходит его компиляция, статус которой нужно проверять с помощью получения эталонного документа (`GET /xapi/ etalonDocument/{id}`) и его отслеживания. При получении дубликатов эталонного документа (STATUS=duplicate), такие файлы будут удалены.

Метод: `POST`
Ресурс: `etalonDocument`
Тело запроса:

Поля:

Название поля	Описание
DISPLAY_NAME	Имя эталонного документа (если не указано, то берётся из поля FILE)
FILE_PATH*	Путь к эталонному документу в источнике, например: smb://example.com/top_secret/etalon.docx, \\example\test\text.rtf, ftp://

Название поля	Описание
	example.com/foo/bar.txt, oracle://example.com:1521/database/table/row, https://examlpe.com/same/html/document По факту, это просто подсказка для автоматического загрузчика, по-которой он может искать
NOTE	Описание эталонного документа
BIN_VALUE_THRESHOLD	Порог цитируемости бинарных данных (по умолчанию берется из первого указанного каталога)
TEXT_VALUE_THRESHOLD	Порог цитируемости текстовых данных (по умолчанию берется из первого указанного каталога)
CATEGORY_ID*	Список GUID-ов категорий, разделенных запятой, например: "E77D893219A24EEAB3D049804FD86751,C743F88121B8449DAF37291474830679,A7AF8EFCF4D54E9393A9B3D750E5ABBB"
FILE*	Сам файл (Передаётся сам файл, а не путь к нему. Сервер не сможет подключиться к вашему компьютеру и забрать его сам. Поэтому весь файл передаётся в запросе)
COMPILE_TYPE*	Тип цифрового отпечатка: • TEXT — Извлекать только текст • ALL — Извлекать и компилировать всё что возможно

* — поле обязательно для заполнения

Ответ: JSON-Schema:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {},
    "id": "http://infowatch.com/tm/etalonDocument",
    "type": "object",
    "properties": {
        "FINGERPRINT_ID": {
            "examples": [
                "139C7026ED48BE13459669024786F7517E5687AE"
            ],
            "title": "Идентификатор эталонного документа",
            "type": "string"
        },
        "DISPLAY_NAME": {
            "title": "Название",
            "type": "string"
        },
        "FILE_PATH": {
            "title": "Полное имя файла",
```

```
        "type": "string"
    },
    "FILE_SIZE": {
        "title": "Размер файла в байтах",
        "type": "integer"
    },
    "MIME": {
        "title": "Mime-type файла",
        "type": "string"
    },
    "BIN_VALUE_THRESHOLD": {
        "title": "Порог цитируемости бинарных данных",
        "type": "integer"
    },
    "TEXT_VALUE_THRESHOLD": {
        "title": "Порог цитируемости текстовых данных",
        "type": "integer"
    },
    "NOTE": {
        "title": "Описание",
        "type": [
            "null",
            "string"
        ]
    },
    "OWNER": {
        "title": "ID токена",
        "type": "integer"
    },
    "STATUS": {
        "examples": [
            "new",
            "compiled",
            "compiling",
            "sending",
            "sent",
            "saving",
            "ready",
            "fatal_error",
            "duplicate",
            "error"
        ],
        "title": "Статус обработки",
        "type": "string"
    },
    "CREATE_DATE": {
        "examples": [
            "2017-07-04 10:45:45.000000"
        ],
        "title": "Дата создания",
        "type": "string"
    },
    "CHANGE_DATE": {
```

```
            "examples": [
                "2017-07-04 10:45:45.000000"
            ],
            "title": "Дата изменения",
            "type": "string"
        },
        "categories": {
            "type": "array",
            "title": "Массив категорий",
            "items": {
                "type": "object",
                "properties": {
                    "CATEGORY_ID": {
                        "examples": [
                            "AC3DF073B60107890A4C704A10577FE700000000"
                        ],
                        "title": "GUID категории",
                        "type": "string"
                    },
                    "DISPLAY_NAME": {
                        "title": "Название",
                        "type": "string"
                    },
                    "DIR_PATH (возможно изменение в названии)": {
                        "title": "Полное имя директории",
                        "type": "string"
                    },
                    "FP_BIN_VALUE_THRESHOLD": {
                        "title": "Порог цитируемости бинарных данных",
                        "type": "integer"
                    },
                    "FP_TEXT_VALUE_THRESHOLD": {
                        "title": "Порог цитируемости текстовых данных",
                        "type": "integer"
                    },
                    "NOTE": {
                        "title": "Описание",
                        "type": [
                            "null",
                            "string"
                        ]
                    },
                    "OWNER": {
                        "title": "ID токена",
                        "type": "integer"
                    },
                    "CREATE_DATE": {
                        "examples": [
                            "2017-07-04 10:45:45.000000"
                        ],
                        "title": "Дата создания",
                        "type": "string"
                    },
```

```
                    "CHANGE_DATE": {
                        "examples": [
                            "2017-07-04 10:45:45.000000"
                        ],
                        "title": "Дата изменения",
                        "type": "string"
                    }
                }
            }
        }
    }
}
```

Пример:

```
POST /xapi/etalonDocument HTTP/1.1
Host: tm.example.com
X-API-Auth-Token: brse99dugp1cdbr69axz
X-API-Version: 1
X-API-DataType: cap
X-API-CompanyId: liw
Content-Type: multipart/form-data; boundary=Asrf456BGe4h
Content-Length: (суммарный объём, включая дочерние заголовки)
(пустая строка)
(отсутствующая преамбула)
--Asrf456BGe4h
Content-Disposition: form-data; name="CATEGORY_ID"
(пустая строка)
2ABECC84F2360B94E0533D003C0A80AE00000000
--Asrf456BGe4h
Content-Disposition: form-data; name="COMPILE_TYPE"
(пустая строка)
ALL
--Asrf456BGe4h
Content-Disposition: form-data; name="FILE"; filename="foo.doc"
Content-Type: application/msword
(пустая строка)
(двоичное содержимое документа)
```

Пример создания файла с помощью CURL

```
curl -i -X POST \
   -H "Content-Type:multipart/form-data" \
   -H "X-API-Auth-Token:brse99dugp1cdbr69axz" \
   -H "X-API-Version:1" \
   -H "X-API-DataType:cap" \
   -H "X-API-CompanyId:liw" \
   -F "CATEGORY_ID=2ABECC84F2360B94E0533D003C0A80AE00000000" \
   -F "FILE=@\"./foo.doc\";filename=\"foo.doc\"" \
   -F "COMPILE_TYPE=ALL" \
 'https://tm.example.com/xapi/etalonDocument'
```

Просмотр эталонных документов REST API

Метод: `GET`
Ресурс: `etalonDocument[/{fingerprint_id}]`
Параметры:

- `filter` — список фильтрации:
  - `create_date[from]` — начало диапазона для фильтрации по дате создания в формате UNIX-timestamp (включая передаваемую дату, можно использовать отдельно от `create_date[to]`)
  - `create_date[to]` — окончание диапазона для фильтрации по дате создания в формате UNIX-timestamp (включая передаваемую дату, можно использовать отдельно от `create_date[from]`)
  - `change_date[from]` — начало диапазона для фильтрации по дате изменения в формате UNIX-timestamp (включая передаваемую дату, можно использовать отдельно от `change_date[to]`)
  - `change_date[to]` — окончание диапазона для фильтрации по дате изменения в формате UNIX-timestamp (включая передаваемую дату, можно использовать отдельно от `change_date[from]`)
  - `display_name[]` — имя файла, можно использовать * в конце, чтобы включить поиск по `LIKE`, например: `?filter[display_name][]=foo*`
  - `file_path[]` — полный путь до файла, можно использовать * в конце, чтобы включить поиск по `LIKE`, например: `?filter[file_path][]=foo*`
  - `category_id[]` — GUID категории, например: `?filter[category_id][]=82EAE496A1686E407B1162E3C3159999F404F559&filter[category_id][]=AF8EAC3C3C8532C3C780481B7D8C5B8E68D0148F`
  - `category_path[]` — полный путь до папки категории, можно использовать * в конце, чтобы включить поиск по `LIKE`, например: `?filter[category_path][]=bar*`
- `with[]` — список дополнительных сущностей, которые должны быть добавлены к объекту:
  - `categories` — список категорий, в которые входит эталонный документ
- `sort[]` — поле, позволяющее сортировать результат. Возможные ключи:
  - `create_date` — `desc/asc`, например: `?sort[create_date]=desc`
  - `change_date` — `desc/asc, например: ?sort[change_date]=asc`

Ответ:

Массив объектов, отдаваемых при создании/редактировании

Примеры:

```
GET /xapi/etalonDocument?filter[create_date][to]=1&filter[display_name]
[]=foo*&filter[display_name][]=bar&with[]=categories&sort[create_date]=desc
GET /xapi/etalonDocument/FAB85F61531BB9E428088EFC81F266FA3C2959E6
```

Обновление и замена содержимого эталонного документа REST API

Метод: POST

Ресурс: `etalonDocument/{fingerprint_id}/update`

Тело запроса:

Поля:

Название поля	Описание
DISPLAY_NAME	Имя эталонного документа (если не указано, то берётся из поля FILE)
FILE_PATH	Путь к эталонному документу в источнике, например: smb://example.com/top_secret/etalon.docx, \\example\test\text.rtf, ftp://example.com/foo/bar.txt, oracle://example.com:1521/database/table/row, https://examlpe.com/same/html/document
NOTE	Описание эталонного документа
BIN_VALUE_THRESHOLD	Порог цитируемости бинарных данных (по умолчанию берется из первого указанного каталога)
TEXT_VALUE_THRESHOLD	Порог цитируемости текстовых данных (по умолчанию берется из первого указанного каталога)
CATEGORY_ID	Список GUID-ов категорий, разделенных запятой, например: "E77D893219A24EEAB3D049804FD86751,C743F88121B8449DAF37291474830679,A7AF8EFCF4D54E9393A9B3D750E5ABBB"
FILE	Сам файл (Передаётся сам файл, а не путь к нему. Сервер не сможет подключиться к вашему компьютеру и забрать его сам. Поэтому весь файл передаётся в запросе)
COMPILE_TYPE	Тип цифрового отпечатка: • TEXT — Извлекать только текст • ALL — Извлекать и компилировать всё что возможно

Передавать нужно только те поля, которые нужно изменить. Остальные поля нужно опускать.

Ответ:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {},
    "id": "http://infowatch.com/tm/etalonDocument",
    "type": "object",
    "properties": {
        "FINGERPRINT_ID": {
            "examples": [
                "139C7026ED48BE13459669024786F7517E5687AE"
            ],
            "title": "Идентификатор эталонного документа",
            "type": "string"
        },
        "DISPLAY_NAME": {
            "title": "Название",
            "type": "string"
```

```
        },
        "FILE_PATH": {
            "title": "Полное имя файла",
            "type": "string"
        },
        "FILE_SIZE": {
            "title": "Размер файла в байтах",
            "type": "integer"
        },
        "MIME": {
            "title": "Mime-type файла",
            "type": "string"
        },
        "BIN_VALUE_THRESHOLD": {
            "title": "Порог цитируемости бинарных данных",
            "type": "integer"
        },
        "TEXT_VALUE_THRESHOLD": {
            "title": "Порог цитируемости текстовых данных",
            "type": "integer"
        },
        "NOTE": {
            "title": "Описание",
            "type": [
                "null",
                "string"
            ]
        },
        "OWNER": {
            "title": "ID токена",
            "type": "integer"
        },
        "STATUS": {
            "examples": [
                "new",
                "compiled",
                "compiling",
                "sending",
                "sent",
                "saving",
                "ready",
                "fatal_error",
                "duplicate",
                "error"
            ],
            "title": "Статус обработки",
            "type": "string"
        },
        "CREATE_DATE": {
            "examples": [
                "2017-07-04 10:45:45.000000"
            ],
            "title": "Дата создания",
```

```
            "type": "string"
        },
        "CHANGE_DATE": {
            "examples": [
                "2017-07-04 10:45:45.000000"
            ],
            "title": "Дата изменения",
            "type": "string"
        },
        "categories": {
            "type": "array",
            "title": "Массив категорий",
            "items": {
                "type": "object",
                "properties": {
                    "CATEGORY_ID": {
                        "examples": [
                            "AC3DF073B60107890A4C704A10577FE700000000"
                        ],
                        "title": "GUID категории",
                        "type": "string"
                    },
                    "DISPLAY_NAME": {
                        "title": "Название",
                        "type": "string"
                    },
                    "DIR_PATH (возможно изменение в названии)": {
                        "title": "Полное имя директории",
                        "type": "string"
                    },
                    "FP_BIN_VALUE_THRESHOLD": {
                        "title": "Порог цитируемости бинарных данных",
                        "type": "integer"
                    },
                    "FP_TEXT_VALUE_THRESHOLD": {
                        "title": "Порог цитируемости текстовых данных",
                        "type": "integer"
                    },
                    "NOTE": {
                        "title": "Описание",
                        "type": [
                            "null",
                            "string"
                        ]
                    },
                    "OWNER": {
                        "title": "ID токена",
                        "type": "integer"
                    },
                    "CREATE_DATE": {
                        "examples": [
                            "2017-07-04 10:45:45.000000"
                        ],
```

```
                        "title": "Дата создания",
                        "type": "string"
                    },
                    "CHANGE_DATE": {
                        "examples": [
                            "2017-07-04 10:45:45.000000"
                        ],
                        "title": "Дата изменения",
                        "type": "string"
                    }
                }
            }
        }
    }
}
```

Удаление эталонного документа REST API

Метод: DELETE
Ресурс: `etalonDocument/{fingerprint_id}`

9.3 Получение доступа к событиям Traffic Monitor внешними системами (DataExport API SDK)

Данный программный интерфейс предназначен для интеграции Traffic Monitor с внешними системами-приемниками данных. В рамках реализации DataExport API внешние системы получают из Traffic Monitor содержимое и метаданные событий, данные организационной структуры, списки пользователей Консоли управления и данные аудита их действий. DataExport API поддерживает двусторонний обмен данными между Traffic Monitor и внешними системами.

9.3.1 Функции DataExport API

DataExport API поддерживает следующий набор функций:

- Возможность указать набор атрибутов, которые будут включены в результат:
  - ID события;
  - Отправители: список контактов и результат идентификации с учетом возможных конфликтов;
  - Получатели: список контактов и результат идентификации с учетом возможных конфликтов;
  - Компьютер отправителя с учетом возможных конфликтов;
  - Тип события;
  - Дата и время перехвата;
  - Протокол;
  - Уровень нарушения;
  - Вердикт;
  - Путь к файлу или адрес приемника;
  - Путь к файлу или адрес источника;
  - Решение пользователя;
  - Путь к файлу;

    - Адрес веб-ресурса;
    - Объекты защиты и каталоги;
    - Размер события;
    - Теги;
    - Набор пользовательских атрибутов, зарегистрированных через плагин;
    - Набор системных заголовков событий;
    - Дата и время вставки события в БД;
    - Наименование периметра, в которое вошло событие;
    - Наименование периметра, из которых ушло событие ;
    - Список наименований политик (с ID политик), сработавших на перехваченном объекте;
    - Группа правил;
    - Сервер перехвата;
    - Сервер перехвата IP;
    - Имя устройства приемника;
    - Имя устройства источника;
    - Тема письма;
    - Вложение;
    - Мандатный уровень;
    - Мандатная категория.
- Возможность указать набор атрибутов, по которым будет осуществляться фильтрация:
    - Дата и время перехвата;
    - Тип события;
    - Протокол;
    - Каталог ОЗ;
    - Решение пользователя;
    - Размер события;
    - Теги;
    - Дата и время вставки события в БД;
    - Группа правил;
    - Уровень нарушения;
    - Вердикт;
    - Дата и время зарегистрированного действия пользователя;
    - Тип объекта, над которым осуществлялось действие.
- Возможность изменить в Traffic Monitor атрибуты события:
    - Вердикт пользователя по событию;
    - Вердикт пользователя по множеству событий.
- Получение метаданных отдельного события с возможностью указать набор атрибутов, которые будут включены в результат. При этом если в событии есть нескольких вложенных объектов, то весь извлеченный текст будет склеен в один файл;
- Получение всего извлеченного текста отдельного события;
- Получение полного списка системных заголовков и зарегистрированных с помощью плагина пользовательских атрибутов событий;
- Получение бинарных данных отдельного события;
- Получение списка пользователей Консоли управления;
- Запуск на выполнение и получение результатов выполнения запросов в Traffic Monitor:
    - список запросов;

    - статус выполнения запросов;
    - результаты запросов;
- Получение данных аудита пользователей:
    - ID записи в аудите сессий пользователей консоли;
    - Дата и время зарегистрированного действия пользователя в UTC;
    - ID пользователя, осуществившего действие;
    - Полное имя пользователя, осуществившего действие;
    - Тип действия, которое было произведено пользователем;
    - Тип объекта, над которым осуществлялось действие;
    - ID объекта, над которым осуществлялось действие;
    - Наименование объекта, над которым осуществлялось действие;
    - Описание произошедших изменений с указанием новых и старых значений.

9.3.2 Авторизация для доступа к DataExport API

Авторизация для доступа к DataExport API предоставляется по механизму токенов, аналогичному другим API InfoWatch Traffic Monitor. Для получения токена и просмотра статуса лицензирования DataExport API в Системе необходимо добавить плагин нового типа - Плагин для DataExport API. Все действия по работе с плагинами выполняются в консоли управления Traffic Monitor.

Работа с плагинами

Добавление плагина

Чтобы добавить плагин:

1. Перейдите в раздел **Управление - Плагины**;
2. В области **Плагины** нажмите  .
3. Нажмите **Добавить**.
4. Просмотрите описание плагина и нажмите **Установить**.

> **Примечание:**
>
> Чтобы обновить плагин, добавьте его обновленную версию.

Удаление плагина

Чтобы удалить плагин:

1. Перейдите в раздел **Управление - Плагины**;
2. В области **Плагины** нажмите  .
3. Нажмите **Да**, чтобы подтвердить удаление.

Работа с токенами

Токен генерируется автоматически после добавления плагина. Все действия по работе с токенами выполняются в консоли управления InfoWatch Traffic Monitor.

Добавление токена

Чтобы добавить токен:

1. Перейдите в раздел **Управление - Плагины**;
2. Выберите плагин в области **Плагины**;
3. Перейдите в закладку **Токены**;
4. Нажмите  .

> **Примечание:**
>
> Значение токена генерируется автоматически при создании токена и не может быть задано пользователем.

Редактирование данных токена

Чтобы изменить данные токена:

1. Перейдите в раздел **Управление - Плагины**;
2. Выберите плагин в области **Плагины**;
3. Перейдите в закладку **Токены**;
4. По двойному нажатию левой кнопкой мыши в графе *Имя* или *Описание* введите нужную информацию.

Обновление значения токена

Если токен скомпрометирован или появилась вероятность его утечки третьим лицам, нужно сгенерировать значение токена заново.

Чтобы обновить значение токена:

1. Перейдите в раздел **Управление - Плагины**;
2. Выберите плагин в области **Плагины**;
3. Перейдите в закладку **Токены**;
4. Нажмите  .

Копирование токена

Чтобы скопировать значение токена в буфер обмена:

1. Перейдите в раздел **Управление - Плагины**;
2. Выберите плагин в области **Плагины**;
3. Перейдите в закладку **Токены**;
4. Нажмите  .

Удаление токена

Чтобы удалить токен:

1. Перейдите в раздел **Управление - Плагины**;

2. Выберите плагин в области **Плагины**;
3. Перейдите в закладку **Токены**;
4. Нажмите  .

9.3.3 Регистрация стороннего компонента DataExport API

Для интеграции сторонней системы-приемника данных с системой Traffic Monitor нужно выполнить следующие шаги:

1. Зарегистрировать в компании InfoWatch строку-идентификатор компании-потребителя компонента. Это позволит связывать компанию-потребителя с лицензией, которая будет устанавливаться в систему Traffic Monitor;
2. Получить лицензию разработчика, связанную с идентификатором компании. Это позволит выгружать из Traffic Monitor данные о событиях;
3. Создать файл регистрации стороннего компонента в описанном ниже формате. В файл регистрации добавляется лицензия, полученная на шаге 2;
4. Загрузить файл регистрации стороннего компонента в систему Traffic Monitor через интерфейс консоли Traffic Monitor. Раздел «Управление» - «Плагины». (см. "Traffic Monitor. Руководство пользователя");
5. Получить токен доступа в свойствах загруженного стороннего компонента;

Система готова отправлять данные о событиях заданной компании.

Формат файла регистрации стороннего компонента DataExport API

Плагин представляет собой архив в формате .zip. В состав архива входят:

- папка **licenses**, содержащая файлы лицензий;
- папка **icon**, содержащая файлы с используемыми пиктограммами для регистрируемых событий;
- файл **manifest.json**, содержащий информацию о плагине.

> **Примечание:**
>
> Папки **licenses** и **icon** не являются обязательными. Файлы лицензий и файлы с используемыми пиктограммами могут находиться в корне.

Для внешних систем-приемников данных из Traffic Monitor файл **manifest.json** должен содержать следующую информацию:

Содержимое	Тип данных	Является обязательным	Описание
`{`			
`"PLUGIN_ID": "",`	Строка	Да	Уникальный идентификатор (UUID) плагина, который создается его разработчиком. До 40 символов. Идентификатор используется для

Содержимое	Тип данных	Является обязательным	Описание
			обновления плагина. **Пример:** `"PLUGIN_ID": "189C38D390396EB6E0530100007F1CA200000001",`
`"DISPLAY_NAME": "",`	Строка	Да	Отображаемое имя плагина. **Пример:** `"DISPLAY_NAME": "Имя плагина",`
`"DESCRIPTION": "",`	Строка	Нет	Отображаемое описание плагина. **Пример:** `"DESCRIPTION": "Тестовый плагин для демонстрации",`
`"VERSION": "",`	Строка	Да	Версия плагина. **Пример:** `"VERSION": "1.0.0",`
`"VENDOR": "",`	Строка	Да	Идентификатор компании-разработчика, соответствующий названию компании в лицензии. **Пример:** `"VENDOR": "infowatch",`
`"LICENSE": [ { "PATH":"" } ],`	Массив Строка	Да	Файлы лицензии. Должны быть указаны пути к файлам лицензий относительно корня архива. **Пример** `"LICENSE": [ { "PATH": "licenses/файл лицензии 1.license" }, { "PATH": "licenses/файл лицензии 2.license" }, ],`
`"PATTERN_SEARCH_LICENSE": "",`	Строка	Нет	Шаблон для поиска загруженных ранее лицензий для привязки их к плагину. Имеет формат: `{operator:or|and,child[{name:value}]}"`. **Пример:** `"PATTERN_SEARCH_LICENSE": "{\"operator\":\"or\", \"conditions\":[{\"origin\":`

Содержимое	Тип данных	Является обязательным	Описание
			`\"dm\"},{\"origin\": \"dmmobile\"}]}",`
`"DATA_RECEIVER": [ "" ]`	Массив Строка	Да	Список идентификаторов систем-приемников данных из Traffic Monitor. **Пример** `"DATA_RECEIVER": [ "Events reciever1", "Events reciever2" ]`

`"OBJECT_HEADER": [ { "NAME": "",`
 	
`"NOTE": { "": "", "": "" },`
`"TYPE": "",`
`"FORMAT": "",`
 	
`"DATA_CLASS": [""],`
`"USE_IN_POLICY": "",`
`"USE_IN_NOTIFICATION": "",`
`"USE_IN_QUERY": "",`
`"USE_IN_LIST": "",`
`"USE_IN_SHOW": "",`
`"USE_IN_DETAIL": "",`
`"IS_MULTIPLE_VALUE": "",`
`}`
`]`
	Массив
	
	Строка
	
	Объект
	Строка
	
	Перечисление
	Перечисление
	
	Массив строк
	Перечисление
	Перечисление
	
	Перечисление
	Перечисление
	Перечисление
	Перечисление
	
	Перечисление
		Нет	Пользовательские атрибуты событий.
			
			Уникальный код атрибута, который будет использован в заголовках события и для идентификации в xAPI/PushAPI.
			Если идентификатор разработчика не "IW", код должен начинаться с префикса " <Идентификатор компании-разработчика>_ ".
			Пример: CISCO_CALL_DURATION
			Название атрибута, отображаемое в консоли Traffic Monitor.
			
			Тип значения.
			Формат записи. Возможны значения: число (целое, дробное), строка, дата и время в UTC + указание смещение часового пояса, длительность, гиперссылка, перечисление, логический тип.
			Сервисы, к событиям которых будет добавлен новый пользовательский атрибут.
			Использование атрибута в политиках. `"1"` - да, `"0"` - нет.
			Использование атрибута в почтовых уведомлениях. `"1"` - да, `"0"` - нет.
			
			Использование атрибута в запросах. `"1"` - да, `"0"` - нет.
			Отображение атрибута в табличном режиме просмотра событий. `"1"` - да, `"0"` - нет.
			Отображение атрибута в краткой форме

Содержимое	Тип данных	Является обязательным	Описание
			просмотра события. `"1"` - да, `"0"` - нет. Отображение атрибута в детальной форме просмотра события. `"1"` - да, `"0"` - нет. Множественное значение. `"1"` - да, `"0"` - нет. **Пример** `"OBJECT_HEADER": [{ "NAME": "header_multi_string", "NOTE": { "rus": "Заголовок строк1", "eng": "Header strings1" }, "DATA_CLASS": ["kEmail"], "USE_IN_POLICY": "1", "USE_IN_QUERY": "1", "USE_IN_NOTIFICATION": "1", "USE_IN_LIST": "1", "USE_IN_SHOW": "1", "USE_IN_DETAIL": "1", "TYPE": "string", "FORMAT": "string", "IS_MULTIPLE_VALUE": "1" }, { "NAME": "header_date", "NOTE": { "rus": "Заголовок дата1", "eng": "Header date1" }, "DATA_CLASS": ["kEmail"], "USE_IN_POLICY": "1", "USE_IN_QUERY": "1", "USE_IN_NOTIFICATION": "1", "USE_IN_LIST": "0", "USE_IN_SHOW": "1", "USE_IN_DETAIL": "1", "TYPE": "string", "FORMAT": "date", "IS_MULTIPLE_VALUE": "0" }]`
`}`			

9.3.4 Функциональное описание программного интерфейса DataExport API

Общие заголовки GET/POST-запросов для DataExport API

Авторизация

Токен, полученный для Public API, необходимо указывать в заголовке **X-API-Auth-Token** при каждом запросе к API. Токен может быть отозван/заблокирован/удалён через web-интерфейс IWTM. В случае, если это произойдет, токен считается недействительным. При указании недействительным токена возвращается сообщение об ошибке с кодом **401**.

Пример ответа при недействительном токене

```
{
    "error": {
        "code": "unauthorized"
    }
}
```

Если токен действующий, но у пользователя API недостаточно прав, то пользователю возвращается сообщение об ошибке с кодом **403**.

Пример ответа при нехватке прав

```
{
    "error": {
        "code": "forbidden"
    }
}
```

Версионирование

Версия API передается с помощью заголовка **X-API-Version** и может иметь 3 состояния:

- supported — версия поддерживается и активно развивается;
- deprecated — версия доступна, но перестала поддерживаться. Рекомендуется перейти на новую версию, так как в следующих выпусках эта версия может быть удалена;
- removed — версия полностью удалена из продукта.

В случае, если сервер не поддерживает версию, ему возвращается сообщение об ошибке с кодом **400**.

Ответ в случае некорректной версии

```
{
    "error": {
        "code": "unsupported_version",
        "meta": {
            "versions": [
                {
                    "NAME":   "1",
```

```
                    "STATUS": "removed"
                },
                {
                    "NAME":   "1.1",
                    "STATUS": "deprecated"
                },
                {
                    "NAME":   "1.2",
                    "STATUS": "deprecated"
                },
                {
                    "NAME":   "1.3",
                    "STATUS": "deprecated"
                },
                {
                    "NAME":   "1.4",
                    "STATUS": "deprecated"
                },
                {
                    "NAME":   "1.5",
                    "STATUS": "deprecated"
                },
                {
                    "NAME":   "1.6",
                    "STATUS": "deprecated"
                },
                {
                    "NAME":   "1.7",
                    "STATUS": "supported"
                }
            ]
        }
    }
}
```

Лицензирование

Для работы модуля требуется лицензия:

```
{
    "common_name": "iw",
    "importer_name": "dome"
}
```

Где `common_name` — это название лицензируемой компании, а `importer_name` — продукт, который принимает данные от TM.

В каждом запросе нужно передавать заголовки **X-API-CompanyId** и **X-API-ProductName**. В случае, если лицензия не проходит проверку, возвращается сообщение об ошибке с кодом **403**.

Формат ответа при невалидной лицензии

```
{
    "error": {
```

```
        "code": "licence_not_valid"
    }
}
```

Таким образом, каждый запрос к серверу должен содержать следующие обязательные заголовки:

- X-API-Version – указывается номер версии программного интерфейса. Текущая версия: **1.6**;
- X-API-CompanyId – указывается идентификатор компании-производителя, например: **iw**;
- X-API-ProductName – указывается продукт, принимающий данные, например: **dome**;
- X-API-Auth-Token – указывается токен авторизации.

Пример запроса

```
X-API-Version: 1.6
X-API-CompanyId: iw
X-API-ProductName: dome
X-API-Auth-Token: gla20dos1pcdbr69caxz
```

Получение списка фич по версии Traffic Monitor

Описание: Список фич для текущей версии Traffic Monitor.

Метод: `GET`

Ресурс: `version/features`

Параметры :

Пример запроса:

```
curl -k -i -X GET    -H "X-Api-Auth-Token:1lap4tvpakbgsbj8tyeu"    -H "X-Api-Version:1.3"    -H "X-Enable-Openapi:1"    -H "X-API-CompanyId:iw"    -H "X-API-ProductName:dome"  'https://qa-797e.infowatch.ru/xapi/version/features'
```

Пример ответа:

```
{
  "data": [
    {
      "VERSION_TM": "7.3.0.285",
      "VERSION_API": "1.6",
      "FEATURES": "sso - сквозная авторизация из vision в tm"
    }
  ]
}
```

Получение данных аудита

> **Примечание:**
>
> В API получения данных аудита могут появляться новые действия. При этом версия API может не меняться.

Описание: Список данных аудита, отсортированных по дате и времени зарегистрированного действия пользователя.

Метод: `GET`

Ресурс: `audit`

Параметры:

- `filter` - список фильтрации:
  - `change_date[>]` – дата и время начала зарегистрированного действия пользователя в формате UNIX-timestamp и через точку microtime, например `filter[change_date][>]=1591368000.123456`
  - `change_date[<]` – дата и время окончания зарегистрированного действия пользователя в формате UNIX-timestamp и через точку microtime, например `filter[change_date][<]=1591368000.123456`
  - `change_date[>=]` – дата и время начала зарегистрированного действия пользователя в формате UNIX-timestamp и через точку microtime, например `filter[change_date][>=]=1591368000.123456`
  - `change_date[<=]` – дата и время окончания зарегистрированного действия пользователя в формате UNIX-timestamp и через точку microtime, например `filter[change_date][<=]=1591368000.123456`
  - `entity_type[]` – тип сущности, над которой было выполнено действие (возможно несколько значений);
- `without[] –` получить данные без дополнительных атрибутов:
  - `property_changes` – данные расширенного аудита.
- `sort[]` – параметр для сортировки:
  - `change_date` — сортировать по дате изменения `asc` .
- `start` – с какой позиции начать отдавать данные аудита (по умолчанию 0)
- `limit` – сколько комплектов данных аудита отдавать (по умолчанию 100, не может быть больше 1000)

OpenAPI-схема отдачи данных аудита

```
schemas:
    auditCommon:
      type: object
      required: [AUDIT_LOG_ID, CHANGE_DATE, OPERATION, ENTITY_TYPE]
      properties: &common
        AUDIT_LOG_ID:
          type: integer
          description: Audit log id
        USER_ID:
          type: integer
          nullable: true
```

```
          description: |
            User id who performed the action (See /xapi/user).
            USER_ID may be null on login or if changes make by plugin.
        CHANGER_NAME:
          type: string
          nullable: true
          description: |
            User name or Plugin name who performed the action.
            May be null on login.
        CHANGE_DATE:
          type: string
          description: Date and time of the registered user action with format "Y-m-d H:i:s.u" in UTC
          example: "2020-05-14 13:53:52.926560"
        OPERATION:
          type: string
          description: Operation type over which the action was performed. There may be values not from the
enum.
          enum:
            - run
            - canceled
            - start
            - stop
            - restart
            - view
            - create
            - update
            - change
            - delete
            - delete_ref
            - delete_hash
            - copy
            - move
            - login_failure
            - bruteforce
            - login
            - logout
            - change_password
            - commit
            - rollback
            - draft
            - add
            - import
            - export
            - export_object
            - create_widget
            - update_widget
            - delete_widget
            - download
            - sync
            - decision_update
            - add_tag
            - remove_tag
            - add_role
            - remove_role
            - add_visibility_area
            - remove_visibility_area
            - logging
            - apply_inbound
```

```
            - xapi_decision_update
            - start_scan_job
            - apply_result_scan_job
            - update_job_schedule
            - create_log
            - plugin_register
            - plugin_update
            - privileges_update
            - privileges_create
            - notification
            - notification_template
            - notification_test
            - upload_image
            - delete_image
            - create_contact
            - update_contact
            - delete_contact
            - compile_create
            - compile_update
            - create_template
            - update_template
            - delete_template
            - create_rule
            - update_rule
            - delete_rule
            - notification_test
        ENTITY_TYPE:
          type: string
          description: Entity type over which the action was performed. There may be values not from the
enum.
          enum: *entryTypes
        ENTITY_ID:
          type: string
          nullable: true
          description: |
            ID of the entity which the action was performed. May be GUID or integer as string
            May by null for:
            | OPERATION | ENTITY_TYPE |
            | :--- | :--- |
            | commit | Config |
            | run | QueryReportRun |
            | sync | Adlibitum |
            | login | User |
            | rollback | Config |
          example: "DBAF4F834F7647CBB297BC9DD17B77B100000000"
        ENTITY_DISPLAY_NAME:
          type: string
          nullable: true
          description: The name of the entity over which the action was carried out
          example: "Some name"
        PROPERTY_CHANGES:
          type: object
          nullable: true
          description: Null in default Audit log events
    userLoginLogout:
      type: object
      description: For ENTITY_TYPE User and OPERATION login or logout
      properties:
```

```
    <<: *common
    OPERATION:
      type: string
      enum: [login, logout]
    ENTITY_TYPE:
      type: string
      enum: [User]
    PROPERTY_CHANGES:
      type: object
      nullable: true
      required: [request]
      properties:
        request:
          required:
            - hostname
            - ip
            - result
            - login
          properties:
            hostname:
              type: string
              title: Hostname
              description: Hostname from which authorization passed
            ip:
              type: string
              title: Ip address
              description: Ip address from which authorization passed
            result:
              type: string
              title: Result
              description: Result
              enum:
                - login_failure
                - login
                - logout
            login:
              type: string
              title: User login
              description: Entered user login in input
userBruteForce:
  type: object
  description: For ENTITY_TYPE User and OPERATION login with bruteforce
  properties:
    <<: *common
    OPERATION:
      type: string
      enum: [ bruteforce ]
    ENTITY_TYPE:
      type: string
      enum: [ User ]
    PROPERTY_CHANGES:
      type: object
      nullable: true
      required: [ login ]
      properties:
        request:
          required:
            - login
```

```
            - display_name
          properties:
            login:
              type: string
              title: User login
              description: Entered user login in input
            display_name:
              type: string
              title: User full name
              description: User full name
tagPropertyChanges:
  type: object
  properties:
    <<: *common
    ENTITY_TYPE:
      type: string
      enum: [ Object ]
    OPERATION:
      type: string
      enum: [ add_tag, remove_tag ]
    PROPERTY_CHANGES:
      type: object
      properties:
        new:
          $ref: '#/components/schemas/TagDiffProperties'
        old:
          $ref: '#/components/schemas/TagDiffProperties'
TagDiffProperties:
  type: object
  properties:
    DISPLAY_NAME:
      type: string
    NOTE:
      type: string
userDesicionPropertyChanges:
  type: object
  properties:
    <<: *common
    ENTITY_TYPE:
      type: string
      enum: [ Object ]
    OPERATION:
      type: string
      enum: [ decision_update, xapi_decision_update ]
    PROPERTY_CHANGES:
      type: object
      properties:
        new:
          type: object
          properties:
            USER_DECISION:
              $ref: '#/components/schemas/UserDecisionState'
        old:
          type: object
          properties:
            USER_DECISION:
              $ref: '#/components/schemas/UserDecisionState'
UserDecisionState:
```

```
    type: string
    enum: [ NotProcessed, NoViolation, Violation, AdditionalProcessingNeeded ]
  objectExportPropertyChanges:
    type: object
    properties:
      <<: *common
      ENTITY_TYPE:
        type: string
        enum: [ Object ]
      OPERATION:
        type: string
        enum: [ export_object ]
      PROPERTY_CHANGES:
        type: object
        properties:
          new:
            type: object
            properties:
              PARAMS:
                type: object
                properties:
                  QUERY_ID:
                    type: integer
                    nullable: true
                  DISPLAY_NAME:
                    type: string
                  COMMENT:
                    type: string
                  FILTER:
                    anyOf:
                      - type: object
                      - type: array
                  SORT:
                    type: object
                  IS_ONE_REPORT:
                    $ref: '#/components/schemas/BooleanAsInteger'
                  ONE_LOAD_REPORT:
                    $ref: '#/components/schemas/BooleanAsInteger'
                  FORMAT:
                    type: string
                    enum: [ simple, full ]
                  TYPE:
                    type: object
                    properties:
                      one_report:
                        type: string
                        enum: [ docx, xls, xlsx, pdf ]
                      several_report:
                        type: string
                        enum: [ docx, pdf ]
                  IS_SEVERAL_REPORT:
                    $ref: '#/components/schemas/BooleanAsInteger'
                  SEVERAL_LOAD_REPORT:
                    $ref: '#/components/schemas/BooleanAsInteger'
                  LOAD_ATTACHMENT:
                    $ref: '#/components/schemas/BooleanAsInteger'
                  KEEP_HIERARCHY:
                    $ref: '#/components/schemas/BooleanAsInteger'
```

```
                    LOAD_SNIPPET:
                      $ref: '#/components/schemas/BooleanAsInteger'
                    LOAD_EML:
                      $ref: '#/components/schemas/BooleanAsInteger'
                    LOAD_DEBUG_OBJECT:
                      $ref: '#/components/schemas/BooleanAsInteger'
                    SCOPE:
                      oneOf:
                        - type: string
                          enum: [ all ]
                        - type: array
                          items:
                            type: integer
                    GROUPING:
                      type: array
                      items:
                        type: string
                        enum: [CaptureDate, Computer, Object, ObjectType, Person]
                    IS_REPORT_ARCHIVE:
                      $ref: '#/components/schemas/BooleanAsInteger'
    BooleanAsStringOrInteger:
      $ref: 'common.yaml#/components/schemas/BooleanAsStringOrInteger'
    BooleanAsInteger:
      $ref: 'common.yaml#/components/schemas/BooleanAsInteger'
    BooleanAsString:
      $ref: 'common.yaml#/components/schemas/BooleanAsString'
```

Пример:

Запрос

```
curl -X GET "https://qa-2944.infowatch.ru/xapi/audit?sort[CHANGE_DATE]=asc&filter[ENTITY_TYPE]
[]=Adlibitum&filter[ENTITY_TYPE][]=Audit&filter[ENTITY_TYPE][]=Config&filter[ENTITY_TYPE]
[]=Perimeter&filter[ENTITY_TYPE][]=Policy&start=0&limit=5" -H "accept: application/json" -H "X-Enable-
OpenAPI: 1"
```

Ответ


```
{
  "data": [
    {
      "AUDIT_LOG_ID": 0,
      "USER_ID": 0,
      "CHANGER_NAME": "string",
      "CHANGE_DATE": "2020-05-14 13:53:52.926560",
      "OPERATION": "run",
      "ENTITY_TYPE": "Adlibitum",
      "ENTITY_ID": "DBAF4F834F7647CBB297BC9DD17B77B100000000",
      "ENTITY_DISPLAY_NAME": "Some name",
      "PROPERTY_CHANGES": {}
    },
    {
      "AUDIT_LOG_ID": 0,
```

```
    "USER_ID": 0,
    "CHANGER_NAME": "string",
    "CHANGE_DATE": "2020-05-14 13:53:52.926560",
    "OPERATION": "login",
    "ENTITY_TYPE": "User",
    "ENTITY_ID": "DBAF4F834F7647CBB297BC9DD17B77B100000000",
    "ENTITY_DISPLAY_NAME": "Some name",
    "PROPERTY_CHANGES": {
      "request": {
        "hostname": "string",
        "ip": "string",
        "result": "login_failure",
        "login": "string"
      }
    }
  },
  {
    "AUDIT_LOG_ID": 0,
    "USER_ID": 0,
    "CHANGER_NAME": "string",
    "CHANGE_DATE": "2020-05-14 13:53:52.926560",
    "OPERATION": "run",
    "ENTITY_TYPE": "Policy",
    "ENTITY_ID": "DBAF4F834F7647CBB297BC9DD17B77B100000000",
    "ENTITY_DISPLAY_NAME": "Some name",
    "PROPERTY_CHANGES": {
      "old": {
        "DISPLAY_NAME": "string",
        "NOTE": "string",
        "STATUS": 0,
        "START_DATE": "string",
        "END_DATE": "string",
        "DATA": {
          "ITEMS": [
            {
              "TYPE": "catalog",
              "ID": "string",
              "NAME": "string"
            },
            {
              "TYPE": "fileformat",
              "ID": "string",
              "NAME": "string",
              "MIME_TYPE": "string",
              "ENCRYPTED": 0,
              "MERGED": 0,
              "MAX": 0,
              "MIN": 0,
              "MAX_TYPE": 0,
              "MIN_TYPE": 0
            }
          ]
        }
      },
      "new": {
        "DISPLAY_NAME": "string",
        "NOTE": "string",
        "STATUS": 0,
```

```
          "START_DATE": "string",
          "END_DATE": "string",
          "DATA": {
            "ITEMS": [
              {
                "TYPE": "catalog",
                "ID": "string",
                "NAME": "string"
              },
              {
                "TYPE": "fileformat",
                "ID": "string",
                "NAME": "string",
                "MIME_TYPE": "string",
                "ENCRYPTED": 0,
                "MERGED": 0,
                "MAX": 0,
                "MIN": 0,
                "MAX_TYPE": 0,
                "MIN_TYPE": 0
              }
            ]
          }
        },
        "rule": {
          "ENTITY_TYPE": "PolicyRule",
          "ENTITY_ID": "string",
          "ENTITY_DISPLAY_NAME": "string"
        }
      }
    }
  ],
  "meta": {
    "totalCount": 0,
    "start": 0,
    "limit": 0
  }
}
```


Получение организационной структуры

Получение информации по группам

Получение списка групп

Описание: Список групп

Метод: `GET`

Ресурс: `group`

Параметры:

- `filter` — список фильтрации:
  - `change_date[from]` — от даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[to]`)

    - `change_date[to]` — до даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[from]`)
    - `create_date[from]` — от даты создания в формате UNIX-timestamp (можно использовать отдельно от `create_date[to]`)
    - `create_date[to]` — до даты создания в формате UNIX-timestamp (можно использовать отдельно от `create_date[from]`
    - `is_deleted` — флаг 1 или 0. Если параметр указан, то будут выдавать сущности с указанным значением
    - `disabled` — флаг 1 или 0, признак активности. Если выставлен, то сущность неактивна (признак берется из LDAP)
    - `contact_type[]` — фильтрация по типу контакта (можно передать несколько, например: `/xapi/group?filter[contact_type][]=webaccount&filter[contact_type][]=icq`)
    - `contact_value[]` — фильтрация по значению контакта (можно передавать несколько, например: `/xapi/group?filter[contact_value][]=user@ad&filter[contact_value][]=login@example.com`)
    - `parent_group_id[]` — фильтрация по группам (можно передать несколько, например: `/xapi/group?filter[parent_group_id][]=FCE1409680E5D6469E1F7998266CE9E9&filter[parent_group_id][]=918462B6DC2C954BAAF8706F7DC53F14)`
- `with[]` — список дополнительных сущностей, которые должны быть добавлены к объекту:
    - `contacts` — контакты группы
    - `children` — подчиненные группы
- `sort` — параметр для сортировки
    - `change_date` — сортировать по дате изменения `asc` или `desc` (например: `/xapi/group?sort[change_date]=asc`) ⚠️ может замедлить выдачу

- `start` — с какой позиции начать отдавать элементы, по-умолчанию 0
- `limit` — сколько элементов отдавать, по-умолчанию 100, не может быть больше 1000

Ответ: Список персон, описанный следующей JSON-Schema:

JSON схема отдачи

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "required": [
       "data",
       "meta"
   ],
   "properties": {
       "data": {
           "description": "Массив групп",
           "type": "array",
           "items": {
               "description": "Группа",
               "type": "object",
               "required": [
                   "GROUP_ID",
                   "CREATE_DATE",
```

```
                    "CHANGE_DATE",
                    "DISPLAY_NAME",
                    "DISTINGUISHEDNAME",
                    "GROUP_TYPE",
                    "IS_VISIBLE",
                    "IS_DELETED",
                    "childrenCount"
                ],
                "properties": {
                    "GROUP_ID": {
                        "description": "Уникальный идентификатор группы",
                        "type": "string"
                    },
                    "CREATE_DATE": {
                        "description": "Дата создания записи",
                        "type": "string"
                    },
                    "CHANGE_DATE": {
                        "description": "Дата изменения записи",
                        "type": "string"
                    },
                    "DISPLAY_NAME": {
                        "description": "Название группы",
                        "type": "string"
                    },
                    "DISTINGUISHEDNAME": {
                        "description": "Полное имя",
                        "type": "string"
                    },
                    "GROUP_TYPE": {
                        "description": "Тип группы (tmRoot, tmGroup, adRoot, adGroup,
adContainer, adDomain, adOU from LDAP objectclass)",
                        "type": "string"
                    },
                    "IS_VISIBLE": {
                        "description": "Видимость группы",
                        "type": "integer"
                    },
                    "IS_DELETED": {
                        "description": "Удалена ли группа из TM",
                        "type": "integer"
                    },
                    "childrenCount": {
                        "description": "Количество детей в группе",
                        "type": "integer"
                    },
                    "contacts": {
                        "description": "Массив контактов",
                        "type": "array",
                        "items": {
                            "description": "Контакт",
                            "type": "object",
                            "required": [
```

```
                    "CONTACT_ID",
                    "CREATE_DATE",
                    "CHANGE_DATE",
                    "CONTACT_TYPE",
                    "VALUE",
                    "IS_PERSONAL",
                    "IS_PRIMARY",
                    "SOURCE"
                ],
                "properties": {
                    "CONTACT_ID": {
                        "description": "Уникальный идентификатор объекта",
                        "type": "string"
                    },
                    "CREATE_DATE": {
                        "description": "Дата создания",
                        "type": "string"
                    },
                    "CHANGE_DATE": {
                        "description": "Дата изменения",
                        "type": "string"
                    },
                    "CONTACT_TYPE": {
                        "description": "Тип",
                        "type": "string"
                    },
                    "VALUE": {
                        "description": "Значение",
                        "type": "string"
                    },
                    "IS_PERSONAL": {
                        "description": "Принадлежность (Рабочий, Личный)",
                        "type": "integer"
                    },
                    "IS_PRIMARY": {
                        "description": "Признак основной",
                        "type": "integer"
                    },
                    "SOURCE": {
                        "description": "Тип",
                        "type": "string"
                    }
                }
            }
        },
        "children": {
            "description": "Массив групп принадлежащих текущей группе",
            "type": "array",
            "items": {
                "description": "Группа",
                "type": "object",
                "properties": {
                    "GROUP_ID": {
```

```
                                    "description": "Уникальный идентификатор группы",
                                    "type": "string"
                                },
                                "PARENT_GROUP_ID": {
                                    "description": "Уникальный идентификатор группы
родителя",
                                    "type": "string"
                                },
                                "RELATION_TYPE": {
                                    "description": "Тип связи: 0 - tm, 1 - member, 2 -
distinguishedname",
                                    "type": "integer"
                                }
                            }
                        }
                    }
                }
            },
            "meta": {
                "description": "Дополнительные данные",
                "type": "object",
                "required": [
                    "totalCount",
                    "start",
                    "limit"
                ],
                "properties": {
                    "totalCount": {
                        "type": "integer",
                        "description": "Общее количество"
                    },
                    "start": {
                        "type": "integer",
                        "description": "С какой позиции началось отдаваться"
                    },
                    "limit": {
                        "type": "integer",
                        "description": "Сколько всего отдалось"
                    }
                }
            }
        }
    }
```

Пример:

Запрос

```
/xapi/group?filter[change_date][from]=1&filter[change_date][to]=2&with[]=contacts&with[
]=children&limit=10&sort[change_date]=asc
```

Получение списка с типами контактов группы

Описание : Список всех возможных типов контактов группы

Метод: `GET`

Ресурс: group `/contact_type`

- `start` — с какой позиции начать отдавать элементы, по-умолчанию 0
- `limit` — сколько элементов отдавать, по-умолчанию 100, не может быть больше 1000

Пример:

Запрос

```
GET /xapi/group/contact_type?start=0&limit=10
```

Получение статистики по группам

Описание: Статистика по группам

Метод: `GET`

Ресурс: group `/stat`

Параметры:

- `filter` — список фильтрации:
  - `change_date[from]` — от даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[to]`)
  - `change_date[to]` — до даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[from]`)

Ответ:

JSON схема отдачи

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "Статистика",
    "type": "object",
    "properties": {
        "data": {
            "description": "Не используется",
            "type": "array",
            "items": {}
        },
        "meta": {
            "description": "Статистика",
            "type": "object",
            "properties": {
                "CHANGE_DATE": {
                    "description": "Дата изменения",
                    "type": "object",
                    "properties": {
                        "from": {
                            "description": "Самая ранняя дата изменения",
                            "type": "string"
```

```
                        },
                        "to": {
                            "description": "Самая поздняя дата изменения",
                            "type": "string"
                        }
                    }
                },
                "totalCount": {
                    "description": "Общее количество элементов",
                    "type": "integer"
                }
            }
        }
    }
}
```

Пример:

```
GET /xapi/group/stat?filter[CAPTURE_DATE][to]=1494951762&filter[CAPTURE_DATE][from]=14
94951000
```

Ответ

```
{
  "data": [],
  "meta": {
    "CHANGE_DATE": {
      "from": "2017-03-24 10:40:03",
      "to": "2017-05-24 12:10:22"
    },
    "totalCount": 539693
  }
}
```

Получение информации по персоне

Получение изображений множества персон

Метод: POST

Ресурс: `person/images`

Параметры (передаются в виде JSON):

- `response_type` — тип ответа:
  - `chunked` — используется `Transfer-Encoding: chunked`, где каждое изображение отдаётся в своём блоке и в порядке, переданном в теле запроса, если изображения нет, то вместо изображения передаётся null символ "\0"
  - `multipart/mixed` — используется формат `multipart/mixed` rfc tech-ms, он не входит в стандарт HTTP, но используется в электронных письмах. Поэтому реализовать его не составляет труда. Так же он очень похож на `multipart/form-data`. Каждое изображение отдаётся в своём блоке и помечено отдельно

заголовком X-Person-ID, к какой персоне оно принадлежит. Если изображения нет, то оно либо не передаётся, либо тело этого изображения отсутствует

- `person_id` — массив ID персон

Примеры:

`chunked` :

Запрос

```
POST /xapi/person/images
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(пустая строка)
{"response_type": "chunked", "person_id": ["DEF1CC1FA015CA428D5DE3CD5B804B69", "E277B6
24BC988C4C96590A8435F634ED"]}
```

Ответ

```
HTTP/1.1 200 OK
Content-Type: image/png
Transfer-Encoding: chunked
(пустая строка)
1
\0 (символ отсутствия изображения для персоны DEF1CC1FA015CA428D5DE3CD5B804B69)
48AC
(содержимое изображения для персоны E277B624BC988C4C96590A8435F634ED)
0
```

`multipart/mixed` :

Запрос

```
POST /xapi/person/images
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(пустая строка)
{"response_type": "multipart/mixed", "person_id": ["DEF1CC1FA015CA428D5DE3CD5B804B69",
 "E277B624BC988C4C96590A8435F634ED"]}
```

Ответ

```
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=Asrf456BGe4h
Content-Length: (суммарный объём, включая дочерние заголовки, может отсутствовать!)
(пустая строка)
(отсутствующая преамбула)
--Asrf456BGe4h
Content-Type: application/x-empty
Content-Disposition: attachment; filename="DEF1CC1FA015CA428D5DE3CD5B804B69"
```

```
X-Person-ID: DEF1CC1FA015CA428D5DE3CD5B804B69
(пустая строка и после нулевое содержимое)
--Asrf456BGe4h
Content-Disposition: attachment; filename="E277B624BC988C4C96590A8435F634ED.jpg"
Content-Type: image/jpeg
X-Person-ID: E277B624BC988C4C96590A8435F634ED
(пустая строка)
(двоичное содержимое второй персоны E277B624BC988C4C96590A8435F634ED)
--Asrf456BGe4h--
(отсутствующий эпилог - пустая строка)
```

Получение изображения персоны

Метод: `GET`

Ресурс: `person/image/{person_id}`

Ответ: Изображение пользователя

Получение списка персон

Описание: Список персон

Метод: `GET`

Ресурс: `person`

Параметры:

- `filter` — список фильтрации:
  - `change_date[from]` — от даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[to]`)
  - `change_date[to]` — до даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[from]`)
  - `create_date[from]` — от даты создания в формате UNIX-timestamp (можно использовать отдельно от `create_date[to]`)
  - `create_date[to]` — до даты создания в формате UNIX-timestamp (можно использовать отдельно от `create_date[from]`
  - is_deleted — флаг 1 или 0. Если параметр указан, то будут выдавать сущности с указанным значением
  - `disabled` — флаг 1 или 0, признак активности. Если выставлен, то сущность неактивна (признак берется из LDAP)
  - `contact_type[]` — фильтрация по типу контакта (можно передать несколько, например: `/xapi/person?filter[contact_type][]=webaccount&filter[contact_type][]=icq`)
  - `contact_value[]` — фильтрация по значению контакта (можно передавать несколько, например: `/xapi/person?filter[contact_value ][]=user@ad&filter[contact_value ][]=login@example.com`)
  - `group_id[]` — фильтрация по группам (можно передать несколько, например: `/xapi/person?filter[group_id][]=webaccount&filter[group_id][]=icq)`
- `with[]` — список дополнительных сущностей, которые должны быть добавлены к объекту:
  - `contacts` — контакты персоны
  - `p2g` — связь персоны с группами
  - `status` — статусы персоны

- `sort` — параметр для сортировки
  - `change_date` — сортировать по дате изменения `asc` или `desc` (например: `/xapi/person?sort[change_date]=asc`) ⚠️ может замедлить выдачу

- `start` — с какой позиции начать отдавать элементы, по-умолчанию 0
- `limit` — сколько элементов отдавать, по-умолчанию 100, не может быть больше 1000

Ответ: Список персон, описанный следующей JSON-Schema:

JSON схема отдачи списка персон

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "required": [
        "data",
        "meta"
    ],
    "properties": {
        "data": {
            "description": "Массив персоны",
            "type": "array",
            "items": {
                "description": "Персона",
                "type": "object",
                "required": [
                    "PERSON_ID",
                    "CREATE_DATE",
                    "CHANGE_DATE",
                    "DISTINGUISHEDNAME",
                    "DISPLAY_NAME",
                    "GIVENNAME",
                    "SN",
                    "DISABLED",
                    "IS_EMPLOYEE",
                    "IS_DELETED"
                ],
                "properties": {
                    "PERSON_ID": {
                        "description": "Уникальный идентификатор объекта",
                        "type": "string"
                    },
                    "CREATE_DATE": {
                        "description": "Дата создания",
                        "type": "string"
                    },
                    "CHANGE_DATE": {
                        "description": "Дата последнего изменения",
                        "type": "string"
                    },
                    "DISTINGUISHEDNAME": {
                        "description": "Полное имя",
```

```
            "type": "string"
        },
        "DISPLAY_NAME": {
            "description": "Имя и фамилия",
            "type": "string"
        },
        "GIVENNAME": {
            "description": "Имя",
            "type": "string"
        },
        "SN": {
            "description": "Фамилия",
            "type": "string"
        },
        "DISABLED": {
            "description": "Состояние персоны",
            "type": "integer"
        },
        "IS_EMPLOYEE": {
            "description": "Является ли сотрудником компании",
            "type": "integer"
        },
        "IS_DELETED": {
            "description": "Удалена ли персона из ТМ",
            "type": "integer"
        },
        "contacts": {
            "description": "Массив контактов",
            "type": "array",
            "items": {
                "description": "Контакт",
                "type": "object",
                "required": [
                    "CONTACT_ID",
                    "CREATE_DATE",
                    "CHANGE_DATE",
                    "CONTACT_TYPE",
                    "VALUE",
                    "IS_PERSONAL",
                    "IS_PRIMARY",
                    "SOURCE"
                ],
                "properties": {
                    "CONTACT_ID": {
                        "description": "Уникальный идентификатор объекта",
                        "type": "string"
                    },
                    "CREATE_DATE": {
                        "description": "Дата создания",
                        "type": "string"
                    },
                    "CHANGE_DATE": {
                        "description": "Дата изменения",
```

```
                                    "type": "string"
                                },
                                "CONTACT_TYPE": {
                                    "description": "Тип",
                                    "type": "string"
                                },
                                "VALUE": {
                                    "description": "Значение",
                                    "type": "string"
                                },
                                "IS_PERSONAL": {
                                    "description": "Принадлежность (Рабочий, Личный)",
                                    "type": "integer"
                                },
                                "IS_PRIMARY": {
                                    "description": "Признак основной",
                                    "type": "integer"
                                },
                                "SOURCE": {
                                    "description": "Тип",
                                    "type": "string"
                                }
                            }
                        }
                    },
                    "p2g": {
                        "description": "Массив связи с группами",
                        "type": "array",
                        "items": {
                            "description": "Связь с группой",
                            "type": "object",
                            "required": [
                                "PERSON_ID",
                                "PARENT_GROUP_ID",
                                "RELATION_TYPE"
                            ],
                            "properties": {
                                "PERSON_ID": {
                                    "description": "Уникальный идентификатор персоны",
                                    "type": "string"
                                },
                                "PARENT_GROUP_ID": {
                                    "description": "Уникальный идентификатор группы",
                                    "type": "string"
                                },
                                "RELATION_TYPE": {
                                    "description": "Тип связи: 0 - tm, 1 - member, 2 -
distinguishedname, 3 - primarygroupsid",
                                    "type": "integer"
                                }
                            }
                        }
                    },
```

```
                    "status": {
                        "description": "Массив статусов",
                        "type": "array",
                        "items": {
                            "description": "Статус персоны",
                            "type": "object",
                            "required": [
                                "IDENTITY_STATUS_ID",
                                "CREATE_DATE",
                                "CHANGE_DATE",
                                "ASSIGNMENT_DATE",
                                "DISPLAY_NAME",
                                "COLOR",
                                "EDITABLE"
                            ],
                            "properties": {
                                "IDENTITY_STATUS_ID": {
                                    "description": "Уникальный идентификатор статуса",
                                    "type": "string"
                                },
                                "CREATE_DATE": {
                                    "description": "Дата создания статуса",
                                    "type": "string"
                                },
                                "CHANGE_DATE": {
                                    "description": "Дата изменения статуса",
                                    "type": "string"
                                },
                                "ASSIGNMENT_DATE": {
                                    "description": "Дата проставления статуса персоне",
                                    "type": "string"
                                },
                                "DISPLAY_NAME": {
                                    "description": "Название статуса",
                                    "type": "string"
                                },
                                "COLOR": {
                                    "description": "Цвет статуса (например #ff0000)",
                                    "type": "string"
                                },
                                "EDITABLE": {
                                    "description": "Признак, что статус не является
системным",
                                    "type": "integer"
                                }
                            }
                        }
                    }
                }
            }
        },
        "meta": {
            "description": "Дополнительные данные",
```

```
            "type": "object",
            "required": [
                "totalCount",
                "start",
                "limit"
            ],
            "properties": {
                "totalCount": {
                    "type": "integer",
                    "description": "Общее количество"
                },
                "start": {
                    "type": "integer",
                    "description": "С какой позиции началось отдаваться"
                },
                "limit": {
                    "type": "integer",
                    "description": "Сколько всего отдалось"
                }
            }
        }
    }
}
```

Пример:

Запрос

```
GET /xapi/person?start=0&limit=10&filter[change_date][from]=0&filter[change_date][to]=1&
with[]=contacts
```

Ответ

```
{
    "data": [{
        "PERSON_ID": "532ECB81F75DDC46BE023478C2E86BD9",
        "WHENCHANGED": "20151229230303.0Z",
        "WHENCREATED": "20151229230303.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Annett Rist,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Annett Rist",
        "GIVENNAME": "Annett",
        "SN": "Rist",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156702",
        "SAMACCOUNTNAME": "AC428097",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC428097@acme.com",
        "MAIL": "Annett.Rist@acme.com",
        "TELEPHONENUMBER": "+49-30-626763",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.401138",
```

```
"CHANGE_DATE": "2017-03-10 08:40:35.741001",
"SOURCE": "ad",
"HAS_TM_THUMBNAILPHOTO": false,
"HAS_THUMBNAILPHOTO": false,
"TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
"contacts": [{
    "CONTACT_ID": "4A211BA3896A7620E0530100007F388100000000",
    "IDENTITY_ID": "532ECB81F75DDC46BE023478C2E86BD9",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac428097@acme.com",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.402060",
    "CHANGE_DATE": "2017-03-07 08:54:56.402060",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389687620E0530100007F388100000000",
    "IDENTITY_ID": "532ECB81F75DDC46BE023478C2E86BD9",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac428097@trafmon",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.401926",
    "CHANGE_DATE": "2017-03-07 08:54:56.401926",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389697620E0530100007F388100000000",
    "IDENTITY_ID": "532ECB81F75DDC46BE023478C2E86BD9",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac428097@trafmon.iw",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.401998",
    "CHANGE_DATE": "2017-03-07 08:54:56.401998",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389667620E0530100007F388100000000",
    "IDENTITY_ID": "532ECB81F75DDC46BE023478C2E86BD9",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "email",
    "VALUE": "annett.rist@acme.com",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.401457",
    "CHANGE_DATE": "2017-03-07 08:54:56.401457",
    "SOURCE": "ad"
```

```
        }, {
            "CONTACT_ID": "4A211BA389677620E0530100007F388100000000",
            "IDENTITY_ID": "532ECB81F75DDC46BE023478C2E86BD9",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "phone",
            "VALUE": "+49-30-626763",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.401670",
            "CHANGE_DATE": "2017-03-07 08:54:56.401670",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3896B7620E0530100007F388100000000",
            "IDENTITY_ID": "532ECB81F75DDC46BE023478C2E86BD9",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156702",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.402130",
            "CHANGE_DATE": "2017-03-07 08:54:56.402130",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "92FB7D1EC79A9947938BE582D355B109",
        "WHENCHANGED": "20151229230304.0Z",
        "WHENCREATED": "20151229230303.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Lieselene SchÃ¶nemann,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Lieselene SchÃ¶nemann",
        "GIVENNAME": "Lieselene",
        "SN": "SchÃ¶nemann",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156706",
        "SAMACCOUNTNAME": "AC727058",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC727058@acme.com",
        "MAIL": "Lieselene.SchÃ¶nemann@acme.com",
        "TELEPHONENUMBER": "+49-2183-398305",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.402195",
        "CHANGE_DATE": "2017-03-10 08:40:35.741198",
        "SOURCE": "ad",
        "HAS_TM_THUMBNAILPHOTO": false,
        "HAS_THUMBNAILPHOTO": false,
        "TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
        "contacts": [{
            "CONTACT_ID": "4A211BA389707620E0530100007F388100000000",
            "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109",
```

```
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "auth",
        "VALUE": "ac727058@acme.com",
        "IS_PRIMARY": 0,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.403048",
        "CHANGE_DATE": "2017-03-07 08:54:56.403048",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA3896E7620E0530100007F388100000000",
        "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "auth",
        "VALUE": "ac727058@trafmon",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.402920",
        "CHANGE_DATE": "2017-03-07 08:54:56.402920",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA3896F7620E0530100007F388100000000",
        "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "auth",
        "VALUE": "ac727058@trafmon.iw",
        "IS_PRIMARY": 0,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.402984",
        "CHANGE_DATE": "2017-03-07 08:54:56.402984",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA3896C7620E0530100007F388100000000",
        "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "email",
        "VALUE": "lieselene.schÃ¶nemann@acme.com",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.402517",
        "CHANGE_DATE": "2017-03-07 08:54:56.402517",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA3896D7620E0530100007F388100000000",
        "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "phone",
        "VALUE": "+49-2183-398305",
        "IS_PRIMARY": 1,
```

```
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.402731",
            "CHANGE_DATE": "2017-03-07 08:54:56.402731",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389717620E0530100007F388100000000",
            "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156706",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.403117",
            "CHANGE_DATE": "2017-03-07 08:54:56.403117",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "F0A3298727D7E0498106AE678CC1F5F8",
        "WHENCHANGED": "20151229230304.0Z",
        "WHENCREATED": "20151229230304.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Kasimir Eggers,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Kasimir Eggers",
        "GIVENNAME": "Kasimir",
        "SN": "Eggers",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156710",
        "SAMACCOUNTNAME": "AC217045",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC217045@acme.com",
        "MAIL": "Kasimir.Eggers@acme.com",
        "TELEPHONENUMBER": "+49-7191-389863",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.403185",
        "CHANGE_DATE": "2017-03-10 08:40:35.741388",
        "SOURCE": "ad",
        "HAS_TM_THUMBNAILPHOTO": false,
        "HAS_THUMBNAILPHOTO": false,
        "TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
        "contacts": [{
            "CONTACT_ID": "4A211BA389767620E0530100007F388100000000",
            "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac217045@acme.com",
            "IS_PRIMARY": 0,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.404114",
            "CHANGE_DATE": "2017-03-07 08:54:56.404114",
```

```
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA389747620E0530100007F388100000000",
        "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "auth",
        "VALUE": "ac217045@trafmon",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.403984",
        "CHANGE_DATE": "2017-03-07 08:54:56.403984",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA389757620E0530100007F388100000000",
        "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "auth",
        "VALUE": "ac217045@trafmon.iw",
        "IS_PRIMARY": 0,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.404055",
        "CHANGE_DATE": "2017-03-07 08:54:56.404055",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA389727620E0530100007F388100000000",
        "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "email",
        "VALUE": "kasimir.eggers@acme.com",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.403507",
        "CHANGE_DATE": "2017-03-07 08:54:56.403507",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA389737620E0530100007F388100000000",
        "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "phone",
        "VALUE": "+49-7191-389863",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.403722",
        "CHANGE_DATE": "2017-03-07 08:54:56.403722",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA389777620E0530100007F388100000000",
        "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8",
        "IDENTITY_TYPE": "person",
```

```
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156710",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.404184",
            "CHANGE_DATE": "2017-03-07 08:54:56.404184",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "52BDB45E49090F4D986899E29FEB35CB",
        "WHENCHANGED": "20151229230304.0Z",
        "WHENCREATED": "20151229230304.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Leoni Huck,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Leoni Huck",
        "GIVENNAME": "Leoni",
        "SN": "Huck",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156709",
        "SAMACCOUNTNAME": "AC716847",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC716847@acme.com",
        "MAIL": "Leoni.Huck@acme.com",
        "TELEPHONENUMBER": "+49-9184-802313",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.404249",
        "CHANGE_DATE": "2017-03-10 08:40:35.741577",
        "SOURCE": "ad",
        "HAS_TM_THUMBNAILPHOTO": false,
        "HAS_THUMBNAILPHOTO": false,
        "TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
        "contacts": [{
            "CONTACT_ID": "4A211BA3897C7620E0530100007F388100000000",
            "IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac716847@acme.com",
            "IS_PRIMARY": 0,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.405104",
            "CHANGE_DATE": "2017-03-07 08:54:56.405104",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3897A7620E0530100007F388100000000",
            "IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac716847@trafmon",
```

```
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.404978",
        "CHANGE_DATE": "2017-03-07 08:54:56.404978",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA3897B7620E0530100007F388100000000",
        "IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "auth",
        "VALUE": "ac716847@trafmon.iw",
        "IS_PRIMARY": 0,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.405040",
        "CHANGE_DATE": "2017-03-07 08:54:56.405040",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA389787620E0530100007F388100000000",
        "IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "email",
        "VALUE": "leoni.huck@acme.com",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.404575",
        "CHANGE_DATE": "2017-03-07 08:54:56.404575",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA389797620E0530100007F388100000000",
        "IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "phone",
        "VALUE": "+49-9184-802313",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.404789",
        "CHANGE_DATE": "2017-03-07 08:54:56.404789",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA3897D7620E0530100007F388100000000",
        "IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "sid",
        "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156709",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.405174",
        "CHANGE_DATE": "2017-03-07 08:54:56.405174",
        "SOURCE": "ad"
```

```
        }],
        "status": []
    }, {
        "PERSON_ID": "BE20920FFD837142AAC09C87A874CCD4",
        "WHENCHANGED": "20151229230304.0Z",
        "WHENCREATED": "20151229230304.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Adelgund Krebs,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Adelgund Krebs",
        "GIVENNAME": "Adelgund",
        "SN": "Krebs",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156708",
        "SAMACCOUNTNAME": "AC944381",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC944381@acme.com",
        "MAIL": "Adelgund.Krebs@acme.com",
        "TELEPHONENUMBER": "+49-6392-145757",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.405238",
        "CHANGE_DATE": "2017-03-10 08:40:35.741765",
        "SOURCE": "ad",
        "HAS_TM_THUMBNAILPHOTO": false,
        "HAS_THUMBNAILPHOTO": false,
        "TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
        "contacts": [{
            "CONTACT_ID": "4A211BA389827620E0530100007F388100000000",
            "IDENTITY_ID": "BE20920FFD837142AAC09C87A874CCD4",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac944381@acme.com",
            "IS_PRIMARY": 0,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.406146",
            "CHANGE_DATE": "2017-03-07 08:54:56.406146",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389807620E0530100007F388100000000",
            "IDENTITY_ID": "BE20920FFD837142AAC09C87A874CCD4",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac944381@trafmon",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.406026",
            "CHANGE_DATE": "2017-03-07 08:54:56.406026",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389817620E0530100007F388100000000",
            "IDENTITY_ID": "BE20920FFD837142AAC09C87A874CCD4",
```

```
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac944381@trafmon.iw",
            "IS_PRIMARY": 0,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.406089",
            "CHANGE_DATE": "2017-03-07 08:54:56.406089",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3897E7620E0530100007F388100000000",
            "IDENTITY_ID": "BE20920FFD837142AAC09C87A874CCD4",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "email",
            "VALUE": "adelgund.krebs@acme.com",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.405584",
            "CHANGE_DATE": "2017-03-07 08:54:56.405584",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3897F7620E0530100007F388100000000",
            "IDENTITY_ID": "BE20920FFD837142AAC09C87A874CCD4",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "phone",
            "VALUE": "+49-6392-145757",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.405836",
            "CHANGE_DATE": "2017-03-07 08:54:56.405836",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389837620E0530100007F388100000000",
            "IDENTITY_ID": "BE20920FFD837142AAC09C87A874CCD4",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156708",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.406214",
            "CHANGE_DATE": "2017-03-07 08:54:56.406214",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "54C067BB3B8518489D098154447242DB",
        "WHENCHANGED": "20151229230304.0Z",
        "WHENCREATED": "20151229230304.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
```

```
"DISTINGUISHEDNAME": "CN=Kilian Adolph,OU=Testou,DC=trafmon,DC=iw",
"DISPLAY_NAME": "Kilian Adolph",
"GIVENNAME": "Kilian",
"SN": "Adolph",
"OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156707",
"SAMACCOUNTNAME": "AC598197",
"NETBIOSNAME": "TRAFMON",
"DNSROOT": "trafmon.iw",
"USERPRINCIPALNAME": "AC598197@acme.com",
"MAIL": "Kilian.Adolph@acme.com",
"TELEPHONENUMBER": "+49-30-399513",
"SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
"CREATE_DATE": "2017-03-07 08:54:56.406279",
"CHANGE_DATE": "2017-03-10 08:40:35.741957",
"SOURCE": "ad",
"HAS_TM_THUMBNAILPHOTO": false,
"HAS_THUMBNAILPHOTO": false,
"TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
"contacts": [{
    "CONTACT_ID": "4A211BA389887620E0530100007F388100000000",
    "IDENTITY_ID": "54C067BB3B8518489D098154447242DB",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac598197@acme.com",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.407140",
    "CHANGE_DATE": "2017-03-07 08:54:56.407140",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389867620E0530100007F388100000000",
    "IDENTITY_ID": "54C067BB3B8518489D098154447242DB",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac598197@trafmon",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.407000",
    "CHANGE_DATE": "2017-03-07 08:54:56.407000",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389877620E0530100007F388100000000",
    "IDENTITY_ID": "54C067BB3B8518489D098154447242DB",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac598197@trafmon.iw",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.407081",
    "CHANGE_DATE": "2017-03-07 08:54:56.407081",
```

```
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389847620E0530100007F388100000000",
            "IDENTITY_ID": "54C067BB3B8518489D098154447242DB",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "email",
            "VALUE": "kilian.adolph@acme.com",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.406603",
            "CHANGE_DATE": "2017-03-07 08:54:56.406603",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389857620E0530100007F388100000000",
            "IDENTITY_ID": "54C067BB3B8518489D098154447242DB",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "phone",
            "VALUE": "+49-30-399513",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.406812",
            "CHANGE_DATE": "2017-03-07 08:54:56.406812",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389897620E0530100007F388100000000",
            "IDENTITY_ID": "54C067BB3B8518489D098154447242DB",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156707",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.407217",
            "CHANGE_DATE": "2017-03-07 08:54:56.407217",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
        "WHENCHANGED": "20151229230305.0Z",
        "WHENCREATED": "20151229230305.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Lene Escher,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Lene Escher",
        "GIVENNAME": "Lene",
        "SN": "Escher",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156712",
        "SAMACCOUNTNAME": "AC673590",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
```

```
"USERPRINCIPALNAME": "AC673590@acme.com",
"MAIL": "Lene.Escher@acme.com",
"TELEPHONENUMBER": "+49-6131-340459",
"SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
"CREATE_DATE": "2017-03-07 08:54:56.407282",
"CHANGE_DATE": "2017-03-10 08:40:35.742722",
"SOURCE": "ad",
"HAS_TM_THUMBNAILPHOTO": false,
"HAS_THUMBNAILPHOTO": false,
"TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
"contacts": [{
    "CONTACT_ID": "4A211BA3898E7620E0530100007F388100000000",
    "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac673590@acme.com",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.408129",
    "CHANGE_DATE": "2017-03-07 08:54:56.408129",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA3898C7620E0530100007F388100000000",
    "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac673590@trafmon",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.408006",
    "CHANGE_DATE": "2017-03-07 08:54:56.408006",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA3898D7620E0530100007F388100000000",
    "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac673590@trafmon.iw",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.408071",
    "CHANGE_DATE": "2017-03-07 08:54:56.408071",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA3898A7620E0530100007F388100000000",
    "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "email",
    "VALUE": "lene.escher@acme.com",
```

```
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.407607",
            "CHANGE_DATE": "2017-03-07 08:54:56.407607",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3898B7620E0530100007F388100000000",
            "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "phone",
            "VALUE": "+49-6131-340459",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.407817",
            "CHANGE_DATE": "2017-03-07 08:54:56.407817",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3898F7620E0530100007F388100000000",
            "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156712",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.408198",
            "CHANGE_DATE": "2017-03-07 08:54:56.408198",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "7B238377335B7B48AFDC99851D8F4793",
        "WHENCHANGED": "20151229230305.0Z",
        "WHENCREATED": "20151229230305.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Annett Schwanke,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Annett Schwanke",
        "GIVENNAME": "Annett",
        "SN": "Schwanke",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156711",
        "SAMACCOUNTNAME": "AC275412",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC275412@acme.com",
        "MAIL": "Annett.Schwanke@acme.com",
        "TELEPHONENUMBER": "+49-33633-744751",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.408263",
        "CHANGE_DATE": "2017-03-10 08:40:35.742148",
        "SOURCE": "ad",
        "HAS_TM_THUMBNAILPHOTO": false,
```

```
"HAS_THUMBNAILPHOTO": false,
"TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
"contacts": [{
    "CONTACT_ID": "4A211BA389947620E0530100007F388100000000",
    "IDENTITY_ID": "7B238377335B7B48AFDC99851D8F4793",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac275412@acme.com",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.409108",
    "CHANGE_DATE": "2017-03-07 08:54:56.409108",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389927620E0530100007F388100000000",
    "IDENTITY_ID": "7B238377335B7B48AFDC99851D8F4793",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac275412@trafmon",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.408985",
    "CHANGE_DATE": "2017-03-07 08:54:56.408985",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389937620E0530100007F388100000000",
    "IDENTITY_ID": "7B238377335B7B48AFDC99851D8F4793",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac275412@trafmon.iw",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.409051",
    "CHANGE_DATE": "2017-03-07 08:54:56.409051",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389907620E0530100007F388100000000",
    "IDENTITY_ID": "7B238377335B7B48AFDC99851D8F4793",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "email",
    "VALUE": "annett.schwanke@acme.com",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.408589",
    "CHANGE_DATE": "2017-03-07 08:54:56.408589",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389917620E0530100007F388100000000",
    "IDENTITY_ID": "7B238377335B7B48AFDC99851D8F4793",
```

```
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "phone",
            "VALUE": "+49-33633-744751",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.408799",
            "CHANGE_DATE": "2017-03-07 08:54:56.408799",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389957620E0530100007F388100000000",
            "IDENTITY_ID": "7B238377335B7B48AFDC99851D8F4793",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156711",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.409176",
            "CHANGE_DATE": "2017-03-07 08:54:56.409176",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "E277B624BC988C4C96590A8435F634ED",
        "WHENCHANGED": "20151229230305.0Z",
        "WHENCREATED": "20151229230305.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Edgar Genz,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Edgar Genz",
        "GIVENNAME": "Edgar",
        "SN": "Genz",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156714",
        "SAMACCOUNTNAME": "AC938446",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC938446@acme.com",
        "MAIL": "Edgar.Genz@acme.com",
        "TELEPHONENUMBER": "+49-7191-731883",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.409247",
        "CHANGE_DATE": "2017-03-10 08:40:35.742335",
        "SOURCE": "ad",
        "HAS_TM_THUMBNAILPHOTO": false,
        "HAS_THUMBNAILPHOTO": false,
        "TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
        "contacts": [{
            "CONTACT_ID": "4A211BA3899A7620E0530100007F388100000000",
            "IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
```

```
        "VALUE": "ac938446@acme.com",
        "IS_PRIMARY": 0,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.410087",
        "CHANGE_DATE": "2017-03-07 08:54:56.410087",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA389987620E0530100007F388100000000",
        "IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "auth",
        "VALUE": "ac938446@trafmon",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.409967",
        "CHANGE_DATE": "2017-03-07 08:54:56.409967",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA389997620E0530100007F388100000000",
        "IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "auth",
        "VALUE": "ac938446@trafmon.iw",
        "IS_PRIMARY": 0,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.410030",
        "CHANGE_DATE": "2017-03-07 08:54:56.410030",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA389967620E0530100007F388100000000",
        "IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "email",
        "VALUE": "edgar.genz@acme.com",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.409572",
        "CHANGE_DATE": "2017-03-07 08:54:56.409572",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA389977620E0530100007F388100000000",
        "IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "phone",
        "VALUE": "+49-7191-731883",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.409781",
        "CHANGE_DATE": "2017-03-07 08:54:56.409781",
```

```
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3899B7620E0530100007F388100000000",
            "IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156714",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.410155",
            "CHANGE_DATE": "2017-03-07 08:54:56.410155",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",
        "WHENCHANGED": "20151229230305.0Z",
        "WHENCREATED": "20151229230305.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Velten SchÃ¶ning,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Velten SchÃ¶ning",
        "GIVENNAME": "Velten",
        "SN": "SchÃ¶ning",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156713",
        "SAMACCOUNTNAME": "AC617874",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC617874@acme.com",
        "MAIL": "Velten.SchÃ¶ning@acme.com",
        "TELEPHONENUMBER": "+49-9184-565860",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.410219",
        "CHANGE_DATE": "2017-03-10 08:40:35.742522",
        "SOURCE": "ad",
        "HAS_TM_THUMBNAILPHOTO": false,
        "HAS_THUMBNAILPHOTO": false,
        "TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
        "contacts": [{
            "CONTACT_ID": "4A211BA389A07620E0530100007F388100000000",
            "IDENTITY_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac617874@acme.com",
            "IS_PRIMARY": 0,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.411056",
            "CHANGE_DATE": "2017-03-07 08:54:56.411056",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3899E7620E0530100007F388100000000",
```

```
        "IDENTITY_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "auth",
        "VALUE": "ac617874@trafmon",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.410937",
        "CHANGE_DATE": "2017-03-07 08:54:56.410937",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA3899F7620E0530100007F388100000000",
        "IDENTITY_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "auth",
        "VALUE": "ac617874@trafmon.iw",
        "IS_PRIMARY": 0,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.411000",
        "CHANGE_DATE": "2017-03-07 08:54:56.411000",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA3899C7620E0530100007F388100000000",
        "IDENTITY_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "email",
        "VALUE": "velten.schÃ¶ning@acme.com",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.410541",
        "CHANGE_DATE": "2017-03-07 08:54:56.410541",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA3899D7620E0530100007F388100000000",
        "IDENTITY_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "phone",
        "VALUE": "+49-9184-565860",
        "IS_PRIMARY": 1,
        "IS_PERSONAL": 0,
        "CREATE_DATE": "2017-03-07 08:54:56.410750",
        "CHANGE_DATE": "2017-03-07 08:54:56.410750",
        "SOURCE": "ad"
    }, {
        "CONTACT_ID": "4A211BA389A17620E0530100007F388100000000",
        "IDENTITY_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",
        "IDENTITY_TYPE": "person",
        "IDENTITY_SOURCE": "ad",
        "CONTACT_TYPE": "sid",
        "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156713",
```

```
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.411129",
            "CHANGE_DATE": "2017-03-07 08:54:56.411129",
            "SOURCE": "ad"
        }],
        "status": []
    }],
    "meta": {
        "totalCount": 232525,
        "start": 0,
        "limit": 10
    }
}
```

Получение списка статусов персоны

Описание : Список всех возможных статусов персон

Метод: `GET`

Ресурс: `person/status`

Параметры:

- `start` — с какой позиции начать отдавать элементы, по-умолчанию 0
- `limit` — сколько элементов отдавать, по-умолчанию 100, не может быть больше 1000

Ответ: Список статусов, описанный следующей JSON-Schema:

JSON схема отдачи списка статусов

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "data": {
            "description": "Массив статусов",
            "type": "array",
            "items": {
                "description": "Статус персоны",
                "type": "object",
                "required": [
                    "IDENTITY_STATUS_ID",
                    "CREATE_DATE",
                    "CHANGE_DATE",
                    "ASSIGNMENT_DATE",
                    "DISPLAY_NAME",
                    "COLOR",
                    "EDITABLE"
                ],
                "properties": {
                    "IDENTITY_STATUS_ID": {
                        "description": "Уникальный идентификатор статуса",
                        "type": "string"
```

```
                    },
                    "CREATE_DATE": {
                        "description": "Дата создания статуса",
                        "type": "string"
                    },
                    "CHANGE_DATE": {
                        "description": "Дата изменения статуса",
                        "type": "string"
                    },
                    "ASSIGNMENT_DATE": {
                        "description": "Дата проставления статуса персоне",
                        "type": "string"
                    },
                    "DISPLAY_NAME": {
                        "description": "Название статуса",
                        "type": "string"
                    },
                    "COLOR": {
                        "description": "Цвет статуса (например #ff0000)",
                        "type": "string"
                    },
                    "EDITABLE": {
                        "description": "Признак, что статус не является системным",
                        "type": "integer"
                    }
                }
            }
        },
        "meta": {
            "description": "Дополнительные данные",
            "type": "object",
            "required": [
                "totalCount",
                "start",
                "limit"
            ],
            "properties": {
                "totalCount": {
                    "description": "Общее количество",
                    "type": "integer"
                },
                "start": {
                    "description": "С какой позиции началось отдаваться",
                    "type": "integer"
                },
                "limit": {
                    "description": "Сколько всего отдалось",
                    "type": "integer"
                }
            }
        }
    }
}
```

Пример:

Запрос

```
GET /xapi/person/status?start=0&limit=10
```

Получение списка с типами контактов персоны

Описание : Список всех возможных типов контактов персоны

Метод: `GET`

Ресурс: `person/contact_type`

Параметры:

- `start` — с какой позиции начать отдавать элементы, по-умолчанию 0
- `limit` — сколько элементов отдавать, по-умолчанию 100, не может быть больше 1000

Ответ: Список типов, описанный следующей JSON-Schema:

JSON схема отдачи списка контактов

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "data": {
            "description": "Массив типов контактов",
            "type": "array",
            "items": {
                "description": "Тип контакта",
                "type": "object",
                "properties": {
                    "CONTACT_TYPE_ID": {
                        "description": "Уникальный идентификатор типа контактов",
                        "type": "string"
                    },
                    "DISPLAY_NAME": {
                        "description": "Название типа контакта",
                        "type": "string"
                    },
                    "LANGUAGE": {
                        "description": "Язык перевода",
                        "type": "string"
                    },
                    "MNEMO": {
                        "description": "Мнемонический идентификатор (используется в
сущности контактов)",
                        "type": "string"
                    }
                }
            }
        },
        "meta": {
```

```
            "description": "Дополнительные данные",
            "type": "object",
            "required": [
                "totalCount",
                "start",
                "limit"
            ],
            "properties": {
                "totalCount": {
                    "description": "Общее количество",
                    "type": "integer"
                },
                "start": {
                    "description": "С какой позиции началось отдаваться",
                    "type": "integer"
                },
                "limit": {
                    "description": "Сколько всего отдалось",
                    "type": "integer"
                }
            }
        }
    }
}
```

Пример:

Запрос

```
GET /xapi/person/contact_type?start=0&limit=10
```

Получение статистики по персонам

Описание: Статистика по персонам

Метод: `GET`

Ресурс: `person/stat`

Параметры:

- `filter` — список фильтрации:
  - `change_date[from]` — от даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[to]`)
  - `change_date[to]` — до даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[from]`)

Ответ:

JSON схема отдачи

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "Статистика",
    "type": "object",
    "properties": {
```

```
        "data": {
            "description": "Не используется",
            "type": "array",
            "items": {}
        },
        "meta": {
            "description": "Статистика",
            "type": "object",
            "properties": {
                "CHANGE_DATE": {
                    "description": "Дата изменения",
                    "type": "object",
                    "properties": {
                        "from": {
                            "description": "Самая ранняя дата изменения",
                            "type": "string"
                        },
                        "to": {
                            "description": "Самая поздняя дата изменения",
                            "type": "string"
                        }
                    }
                },
                "totalCount": {
                    "description": "Общее количество элементов",
                    "type": "integer"
                }
            }
        }
    }
}
```

Пример:

```
GET /xapi/person/stat?filter[change_date][to]=1494951762&filter[change_date][from]=149
4951000
```

Ответ

```
{
  "data": [],
  "meta": {
    "CHANGE_DATE": {
      "from": "2017-03-24 10:40:03",
      "to": "2017-05-24 12:10:22"
    },
    "totalCount": 539693
  }
}
```

Получение информации по рабочим станциям

Получение списка рабочих станций

Описание : Список рабочих станций

Метод: `GET`

Ресурс: `workstation`

Параметры:

- `filter` — список фильтрации:
  - `change_date[from]` — от даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[to]`)
  - `change_date[to]` — до даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[from]`)
  - `create_date[from]` — от даты создания в формате UNIX-timestamp (можно использовать отдельно от `create_date[to]`)
  - `create_date[to]` — до даты создания в формате UNIX-timestamp (можно использовать отдельно от `create_date[from]`
  - is_deleted — флаг 1 или 0. Если параметр указан, то будут выдавать сущности с указанным значением
  - `disabled` — флаг 1 или 0, признак активности. Если выставлен, то сущность неактивна (признак берется из LDAP)

  - `contact_type[]` — фильтрация по типу контакта (можно передать несколько, например: `/xapi/workstation?filter[contact_type][]=webaccount&filter[contact_type][]=icq`)
  - `contact_value[]` — фильтрация по значению контакта (можно передавать несколько, например: `/xapi/workstation?filter[contact_value][]=user@ad&filter[contact_value][]=login@example.com`)
  - `group_id[]` — фильтрация по группам (можно передать несколько, например: `/xapi/workstation?filter[group_id][]=FCE1409680E5D6469E1F7998266CE9E9&filter[group_id][]=918462B6DC2C954BAAF8706F7DC53F14)`
- `with[]` — список дополнительных сущностей, которые должны быть добавлены к объекту:
  - `contacts` — контакты рабочей станции
  - `w2g` — связь рабочей станции с группами
  - `status` — статусы рабочей станции
- `sort` — параметр для сортировки
  - `change_date` — сортировать по дате изменения `asc` или `desc` (например: `/xapi/workstation?sort[change_date]=asc`) может замедлить выдачу
- `start` — с какой позиции начать отдавать элементы, по умолчанию 0
- `limit` — сколько элементов отдавать, по умолчанию 100, не может быть больше 1000

Ответ : Список рабочих станций, описанный в следующей JSON-Schema :

JSON схема отдачи

```
{
```

```
"$schema": "http://json-schema.org/draft-04/schema#",
"type": "object",
"required": [
    "data",
    "meta"
],
"properties": {
    "data": {
        "description": "Массив Рабочих станций",
        "type": "array",
        "items": {
            "description": "Рабочая станция",
            "type": "object",
            "required": [
                "WORKSTATION_ID",
                "CREATE_DATE",
                "CHANGE_DATE",
                "DISPLAY_NAME",
                "DISTINGUISHEDNAME",
                "DNSHOSTNAME",
                "DISABLED",
                "TYPE",
                "IS_DELETED"
            ],
            "properties": {
                "WORKSTATION_ID": {
                    "description": "Уникальный идентификатор рабочей станции",
                    "type": "string"
                },
                "CREATE_DATE": {
                    "description": "Дата создания записи",
                    "type": "string"
                },
                "CHANGE_DATE": {
                    "description": "Дата изменения записи",
                    "type": "string"
                },
                "DISPLAY_NAME": {
                    "description": "Название рабочей станции",
                    "type": "string"
                },
                "DISTINGUISHEDNAME": {
                    "description": "Отличительное имя",
                    "type": "string"
                },
                "DNSHOSTNAME": {
                    "description": "Полное имя",
                    "type": "string"
                },
                "DISABLED": {
                    "description": "Состояние рабочей станции",
                    "type": "integer"
                },
```

```
                    "TYPE": {
                        "description": "Тип: computer – Рабочая станция, mobile –
Мобильное устройство, terminal – Терминальный сервер",
                        "type": "string"
                    },
                    "IS_DELETED": {
                        "description": "Удалена ли рабочая станция из ТМ",
                        "type": "integer"
                    },
                    "contacts": {
                        "description": "Массив контактов",
                        "type": "array",
                        "items": {
                            "description": "Контакт",
                            "type": "object",
                            "required": [
                                "CONTACT_ID",
                                "CREATE_DATE",
                                "CHANGE_DATE",
                                "CONTACT_TYPE",
                                "VALUE",
                                "IS_PERSONAL",
                                "IS_PRIMARY",
                                "SOURCE"
                            ],
                            "properties": {
                                "CONTACT_ID": {
                                    "description": "Уникальный идентификатор объекта",
                                    "type": "string"
                                },
                                "CREATE_DATE": {
                                    "description": "Дата создания",
                                    "type": "string"
                                },
                                "CHANGE_DATE": {
                                    "description": "Дата изменения",
                                    "type": "string"
                                },
                                "CONTACT_TYPE": {
                                    "description": "Тип",
                                    "type": "string"
                                },
                                "VALUE": {
                                    "description": "Значение",
                                    "type": "string"
                                },
                                "IS_PERSONAL": {
                                    "description": "Принадлежность (Рабочий, Личный)",
                                    "type": "integer"
                                },
                                "IS_PRIMARY": {
                                    "description": "Признак основной",
                                    "type": "integer"
```

```
                                },
                                "SOURCE": {
                                    "description": "Тип",
                                    "type": "string"
                                }
                            }
                        }
                    },
                    "w2g": {
                        "description": "Массив связи с группами",
                        "type": "array",
                        "items": {
                            "description": "Связь с группой",
                            "type": "object",
                            "required": [
                                "WORKSTATION_ID",
                                "PARENT_GROUP_ID",
                                "RELATION_TYPE"
                            ],
                            "properties": {
                                "WORKSTATION_ID": {
                                    "description": "Уникальный идентификатор рабочей
станции",
                                    "type": "string"
                                },
                                "PARENT_GROUP_ID": {
                                    "description": "Уникальный идентификатор группы",
                                    "type": "string"
                                },
                                "RELATION_TYPE": {
                                    "description": "Тип связи: 0 - tm, 1 - member, 2 -
distinguishedname, 3 - primarygroupsid",
                                    "type": "integer"
                                }
                            }
                        }
                    },
                    "status": {
                        "description": "Массив статусов",
                        "type": "array",
                        "items": {
                            "description": "Статус персоны",
                            "type": "object",
                            "required": [
                                "IDENTITY_STATUS_ID",
                                "CREATE_DATE",
                                "CHANGE_DATE",
                                "ASSIGNMENT_DATE",
                                "DISPLAY_NAME",
                                "COLOR",
                                "EDITABLE"
                            ],
                            "properties": {
```

```
                                "IDENTITY_STATUS_ID": {
                                    "description": "Уникальный идентификатор статуса",
                                    "type": "string"
                                },
                                "CREATE_DATE": {
                                    "description": "Дата создания статуса",
                                    "type": "string"
                                },
                                "CHANGE_DATE": {
                                    "description": "Дата изменения статуса",
                                    "type": "string"
                                },
                                "ASSIGNMENT_DATE": {
                                    "description": "Дата проставления статуса персоне",
                                    "type": "string"
                                },
                                "DISPLAY_NAME": {
                                    "description": "Название статуса",
                                    "type": "string"
                                },
                                "COLOR": {
                                    "description": "Цвет статуса (например #ff0000)",
                                    "type": "string"
                                },
                                "EDITABLE": {
                                    "description": "Признак, что статус не является
системным",
                                    "type": "integer"
                                }
                            }
                        }
                    }
                }
            }
        },
        "meta": {
            "description": "Дополнительные данные",
            "type": "object",
            "required": [
                "totalCount",
                "start",
                "limit"
            ],
            "properties": {
                "totalCount": {
                    "type": "integer",
                    "description": "Общее количество"
                },
                "start": {
                    "type": "integer",
                    "description": "С какой позиции началось отдаваться"
                },
                "limit": {
```

```
                    "type": "integer",
                    "description": "Сколько всего отдалось"
                }
            }
        }
    }
}
```

Пример:

Запрос

```
/xapi/workstation?filter[change_date][from]=1&filter[change_date][to]=2&with[]=contact
s&with[]=w2g&limit=10&sort[change_date]=asc
```

Получение списка со статусами рабочих станций

Описание : Список всех возможных статусов рабочих станций

Метод: `GET`

Ресурс: `workstation/status`

- `start` — с какой позиции начать отдавать элементы, по-умолчанию 0
- `limit` — сколько элементов отдавать, по-умолчанию 100, не может быть больше 1000

Пример:

Запрос

```
GET /xapi/workstation/status?start=0&limit=10
```

Получение списка с типами контактов рабочих станций

Описание : Список всех возможных типов контактов рабочих станций

Метод: `GET`

Ресурс: `workstation/contact_type`

- `start` — с какой позиции начать отдавать элементы, по-умолчанию 0
- `limit` — сколько элементов отдавать, по-умолчанию 100, не может быть больше 1000

Пример:

Запрос

```
GET /xapi/workstation/contact_type?start=0&limit=10
```

Получение статистики по рабочим станциям

Описание: Статистика по рабочим станциям

Метод: `GET`

Ресурс: workstation `/stat`

Параметры:

- `filter` — список фильтрации:
  - `change_date[from]` — от даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[to]`)
  - `change_date[to]` — до даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[from]`)

Ответ:

JSON схема отдачи

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "Статистика",
    "type": "object",
    "properties": {
        "data": {
            "description": "Не используется",
            "type": "array",
            "items": {}
        },
        "meta": {
            "description": "Статистика",
            "type": "object",
            "properties": {
                "CHANGE_DATE": {
                    "description": "Дата изменения",
                    "type": "object",
                    "properties": {
                        "from": {
                            "description": "Самая ранняя дата изменения",
                            "type": "string"
                        },
                        "to": {
                            "description": "Самая поздняя дата изменения",
                            "type": "string"
                        }
                    }
                },
                "totalCount": {
                    "description": "Общее количество элементов",
                    "type": "integer"
                }
            }
        }
    }
}
```

Пример:

```
GET /xapi/workstation/stat?filter[CAPTURE_DATE][to]=1494951762&filter[CAPTURE_DATE]
[from]=1494951000
```

Ответ

```
{
  "data": [],
  "meta": {
    "CHANGE_DATE": {
      "from": "2017-03-24 10:40:03",
      "to": "2017-05-24 12:10:22"
    },
    "totalCount": 539693
  }
}
```

Получение событий

Получение извлеченного текста множества событий

Метод: `POST`

Ресурс: `event/texts`

Параметры (передаются в виде JSON):

- `response_type` — тип ответа:
  - `chunked` — используется `Transfer-Encoding: chunked`, где каждый текст отдаётся в своём блоке и в порядке, переданном в теле запроса, если текста нет, то вместо текста передаётся null символ "\0"
  - `multipart/mixed` — используется формат `multipart/mixed` rfc tech-ms, он не входит в стандарт HTTP, но используется в электронных письмах. Поэтому реализовать его не составляет труда. Также он очень похож на `multipart/form-data`. Каждый текст отдаётся в своём блоке и помечен отдельно заголовком X-Event-ID, к какому событию он принадлежит. Если текста нет, то он либо не передаётся, либо тело этого текста отсутствует
- `event` — массив объектов с полями:
  - `id` — ID события
  - `tbsId` — Табличное пространство, в котором событие

Примеры:

`chunked`:

Запрос

```
POST /xapi/event/texts
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(пустая строка)
{"response_type": "chunked", "events": [{"id": 2, "tbsId": 4}, {"id": 4, "tbsId": 4}]}
```

Ответ

```
HTTP/1.1 200 OK
Content-Type: text/plain
Transfer-Encoding: chunked
(пустая строка)
1
\0 (символ отсутствия текста для события 2)
48AC
(содержимое текста для события 4)
0
```

`multipart/mixed` :

Запрос

```
POST /xapi/event/texts
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(пустая строка)
{"response_type": "multipart/mixed", "events": [{"id": 2, "tbsId": 4}, {"id": 4, "tbsId": 4}]}
```

Ответ

```
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=Asrf456BGe4h
Content-Length: (суммарный объём, включая дочерние заголовки, может отсутствовать!)
(пустая строка)
(отсутствующая преамбула)
--Asrf456BGe4h
Content-Type: application/x-empty
Content-Disposition: attachment; filename="text-2.txt"
X-Event-ID: 2
(пустая строка и после нулевое содержимое)
--Asrf456BGe4h
Content-Disposition: attachment; filename="text-4.txt"
Content-Type: text/plain
X-Event-ID: 4
(пустая строка)
(содержимое текста для события 4)
--Asrf456BGe4h--
(отсутствующий эпилог - пустая строка)
```

Получение извлеченного текста события

Метод: `POST`

Ресурс: `event/texts`

Параметры (передаются в виде JSON):

- `response_type` — тип ответа:

    - `chunked` — используется `Transfer-Encoding: chunked`, где каждый текст отдаётся в своём блоке и в порядке, переданном в теле запроса, если текста нет, то вместо текста передаётся null символ "\0"
    - `multipart/mixed` — используется формат `multipart/mixed` rfc tech-ms, он не входит в стандарт HTTP, но используется в электронных письмах. Поэтому реализовать его не составляет труда. Так же он очень похож на `multipart/form-data`. Каждый текст отдаётся в своём блоке и помечен отдельно заголовком X-Event-ID, к какому событию он принадлежит. Если текста нет, то оно либо не передаётся, либо тело этого текста отсутствует
- `events` — массив объектов с полями:
    - `id` — ID события
    - `tbsId` — Табличное пространство события

Примеры:

`chunked`:

Запрос

```
 POST /xapi/event/texts
X-API-Version: 1.1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(пустая строка)
{"response_type": "chunked", "events": [{"id": 2, "tbsId": 4}, {"id": 4, "tbsId": 4}]}
```

Ответ

```
HTTP/1.1 200 OK
Content-Type: text/plain
Transfer-Encoding: chunked
(пустая строка)
1
\0 (символ отсутствия текста для события 2)
48AC
(содержимое текста для события 4)
0
```

`multipart/mixed`:

Запрос

```
POST /xapi/event/texts
X-API-Version: 1.1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(пустая строка)
{"response_type": "multipart/mixed", "events": [{"id": 2, "tbsId": 4}, {"id": 4, "tbsId": 4}]}
```

Ответ

```
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=Asrf456BGe4h
Content-Length: (суммарный объём, включая дочерние заголовки, может отсутствовать!)
(пустая строка)
(отсутствующая преамбула)
--Asrf456BGe4h
Content-Type: application/x-empty
Content-Disposition: attachment; filename="text-2.txt"
X-Event-ID: 2
(пустая строка и после нулевое содержимое)
--Asrf456BGe4h
Content-Disposition: attachment; filename="text-4.txt"
Content-Type: text/plain
X-Event-ID: 4
(пустая строка)
(содержимое текста для события 4)
--Asrf456BGe4h--
(отсутствующий эпилог - пустая строка)
```

Получение метаданных события

Метод: `GET`

Ресурс: `event/{id}`

Параметры:

- `{id}` - `OBJECT_ID` события
- `with[]` - список дополнительных сущностей, которые должны быть добавлены к объекту (например: отправители, получатели и т.д.)

Ответ: Событие

Пример:

Получить событие с OBJECT_ID "1" с отправителями, получателями:

```
GET /xapi/event/1?with[]=senders&with[]=recipients
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: aplab
X-API-Auth-Token: gla20dos1pcdbr69caxz
```

Получение множества контентов событий

Метод: `POST`

Ресурс: `event/raws`

Параметры (передаются в виде JSON):

- `response_type` — тип ответа:
  - `chunked` — используется `Transfer-Encoding: chunked`, где каждый контент отдаётся в своём блоке и в порядке, переданном в теле запроса, если контента нет, то вместо контента передаётся null символ "\0"

    - `multipart/mixed` — используется формат `multipart/mixed` rfc tech-ms, он не входит в стандарт HTTP, но используется в электронных письмах. Поэтому реализовать его не составляет труда. Так же он очень похож на `multipart/form-data`. Каждый контент отдаётся в своём блоке и помечен отдельно заголовками X-Event-ID и X-Content-ID, к какому событию и контенту он принадлежит. Если контента нет, то оно либо не передаётся, либо тело этого текста отсутствует
- `event` — массив объектов с полями:
    - `id` — ID события
    - `tbsId` — Табличное пространство, в котором событие
    - `contentId` — ID контента. Если не указан, то отдаётся главный узел

Примеры:

`chunked`:

Запрос

```
POST /xapi/event/raws
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(пустая строка)
{"response_type": "chunked", "events": [{"id": 2, "tbsId": 4}, {"id": 3, "tbsId": 4}, {"id": 4, "tbsId": 4,
"contentId": 8}]}
```

Ответ

```
HTTP/1.1 200 OK
Content-Type: text/plain
Transfer-Encoding: chunked
(пустая строка)
1
\0 (символ отсутствия контента для события 2)
3E4
(содержимое корневого контента для события 3)
48AC
(содержимое контента 8 для события 4)
0
```

`multipart/mixed`:

Запрос

```
POST /xapi/event/raws
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(пустая строка)
{"response_type": "multipart/mixed", "events": [{"id": 2, "tbsId": 4}, {"id": 3, "tbsId": 4}, {"id": 4,
"tbsId": 4, "contentId": 8}]}
```

Ответ

```
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=Asrf456BGe4h
Content-Length: (суммарный объём, включая дочерние заголовки, может отсутствовать!)
(пустая строка)
(отсутствующая преамбула)
--Asrf456BGe4h
Content-Type: application/x-empty
Content-Disposition: attachment; filename="content-2"
X-Event-ID: 2
(пустая строка и после нулевое содержимое)
--Asrf456BGe4h
Content-Disposition: attachment; filename="content-3-5.eml"
Content-Type: email
X-Event-ID: 3
X-Content-ID: 5
(пустая строка)
(содержимое корневого контента для события 3)
--Asrf456BGe4h
Content-Disposition: attachment; filename="my-project.jpg"
Content-Type: image/jpeg
X-Event-ID: 4
X-Content-ID: 8
(пустая строка)
(содержимое контента 8 для события 4)
--Asrf456BGe4h--

(отсутствующий эпилог - пустая строка)
```

Получение информации о контентах события

Метод: `POST`

Ресурс: `event/contents`

Параметры (передаются в виде JSON):

- `response_type` – тип ответа:
  - `chunked` – используется `Transfer-Encoding: chunked`, где каждый контент отдаётся в своём блоке и в порядке, переданном в теле запроса, если контента нет, то вместо контента передаётся null символ "\0"
  - `multipart/mixed` – используется формат `multipart/mixed` rfc tech-ms, он не входит в стандарт HTTP, но используется в электронных письмах. Поэтому реализовать его не составляет труда. Так же он очень похож на `multipart/form-data`. Каждый контент отдаётся в своём блоке и помечен отдельно заголовками X-Event-ID и X-Content-ID, к какому событию и контенту он принадлежит. Если контента нет, то оно либо не передаётся, либо тело этого текста отсутствует
- `events` – массив объектов с полями:
  - `id` – ID одного или нескольких событий, по которым нужно получить контенты;
  - `tbsId` – ID табличных пространств для каждого события;
- `filter[mime]`– типы контентов с возможностью указать несколько значений (по точному совпадению или по маске) или сразу все
- `start` - с какой позиции начать отдавать контенты, по-умолчанию 0

- `limit` - сколько контентов отдавать. По умолчанию 100, не может быть больше 1000 в одном запросе.

Пример:

Запрос

```
curl -X POST "https://qa-2e55.infowatch.ru/xapi/event/contents?filter[mime][]=image%2Fjpeg&filter[mime][]=text%2F%2A&start=1&limit=10" -H "accept: application/json" -H "Content-Type: application/json" -H "X-Enable-OpenAPI: 1" -d "{\"events\":[{\"id\":2,\"tbsId\":4}]}"
```

По запросу будут предоставлены ID контентов, соответствующих условиям запроса, с указанием следующих атрибутов:

Ответ

```
{
  "data": [
    {
      "OBJECT_CONTENT_ID": 15,
      "MIME": "string",
      "OBJECT_ID": 2,
      "TBS_ID": 4,
      "CONTENT_SIZE": 1500,
      "is_filename": 0           // признак, что контент является или не является именем файла
    }
  ],
  "meta": {
    "totalCount": 1,
    "start": 1,
    "limit": 10
  }
}
```

Получение перехваченного файла события

Метод: `GET`

Ресурс: `event/{id}/raw?download=0`

Параметры:

- `{id}` - OBJECT_ID события, обязательный
- `download` - добавляет HTTP-заголовки для скачивания файла (`Content-Disposition`, `Content-Transfer-Encoding` и прочие), обязательный, значения 1 или 0 (по умолчанию 0)

Поддерживается HTTP-заголовок `Range`

Ответ: Файл перехваченного события. Например: eml (для писем) или сам файл для событий копирования файла.

Получение списка ОЗ

Описание : Список ОЗ, отсортированный по имени.

Метод: `GET`

Ресурс: `event/protectedDocument`

Параметры:

- `start` - с какой позиции начать отдавать ОЗ, по-умолчанию 0
- `limit` - сколько ОЗ отдавать, по-умолчанию 100, не может быть больше 1000
- `with[]` — список дополнительных сущностей, которые должны быть добавлены к ОЗ:
    - `pd2pc` — связь объекта защиты к каталогу объекта защиты

Ответ: Список ОЗ, описанный следующей **JSON схемой отдачи списка ОЗ:**

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "data": {
            "type": "array",
            "description": "Массив ОЗ",
            "items": {
                "type": "object",
                "description": "ОЗ",
                "properties": {
                    "DOCUMENT_ID": {
                        "type": "string",
                        "description": "Идентификатор"
                    },
                    "DISPLAY_NAME": {
                        "type": "string",
                        "description": "Имя"
                    },
                    "CREATE_DATE": {
                        "type": "string",
                        "description": "Дата создания"
                    },
                    "IS_DELETED": {
                        "type": "integer",
                        "description": "Удален ли из активной конфигурации",
                        "enum": [
                            0,
                            1
                        ]
                    },
                    "pd2pc": {
                        "description": "Массив связи с каталогами ОЗ",
                        "type": "array",
                        "items": {
                            "description": "Связь с каталогом ОЗ",
                            "type": "object",
                            "required": [
                                "CATALOG_ID",
                                "DOCUMENT_ID"
                            ],
                            "properties": {
                                "CATALOG_ID": {
```

```
                                    "description": "Уникальный идентификатор каталога
ОЗ",
                                    "type": "string"
                                },
                                "DOCUMENT_ID": {
                                    "description": "Уникальный идентификатор ОЗ",
                                    "type": "string"
                                }
                            }
                        }
                    }
                },
                "required": [
                    "DOCUMENT_ID",
                    "DISPLAY_NAME",
                    "CREATE_DATE",
                    "IS_DELETED"
                ]
            }
        },
        "meta": {
            "description": "Дополнительные данные",
            "type": "object",
            "properties": {
                "totalCount": {
                    "type": "integer",
                    "description": "Общее количество ОЗ"
                },
                "start": {
                    "type": "integer",
                    "description": "С какой позиции началось отдаваться"
                },
                "limit": {
                    "type": "integer",
                    "description": "Сколько всего отдалось"
                }
            },
            "required": [
                "totalCount",
                "start",
                "limit"
            ]
        }
    },
    "required": [
        "data",
        "meta"
    ]
}
```


Пример:

Запрос

```
GET /xapi/event/protectedDocument?start=0&limit=10
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
```

Ответ

```
{
    "data": [{
        "DOCUMENT_ID": "C78E821A67C7F34A1B1D63A4151CEF6300000000",
        "DISPLAY_NAME": "Грифованная информация",
        "CREATE_DATE": "2016-05-12 10:24:41.342554",
        "IS_DELETED": 0
    }],
    "meta": {
        "totalCount": 1,
        "start": 0,
        "limit": 10
    }
}
```

Получение списка заголовков

Описание: Получение списка дополнительных полей события.

Метод: `GET`

Ресурс: `event/header`

Параметры:

- `start` - с какой позиции начать отдавать заголовки, по-умолчанию 0
- `limit` - сколько заголовков отдавать, по-умолчанию 100, не может быть больше 1000

Ответ: Список дополнительный полей, описанных следующей **JSON схемой отдачи:**

```
{
    "$id": "http://infowatch.com/iwtm/xapi/event/header.json",
    "type": "object",
    "$schema": "http://json-schema.org/draft-07/schema#",
    "properties": {
        "data": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "NAME": {
                        "type": "string",
```

```
                        "title": "Код атрибута, который будет использоваться в заголовках события и для идентификации в xAPI/PushAPI",
                        "examples": [
                            "mac_level"
                        ]
                    },
                    "NOTE": {
                        "type": "string",
                        "title": "Отображаемое в консоли ТМ название атрибута",
                        "examples": [
                            "Уровень мандатного доступа"
                        ]
                    },
                    "TYPE": {
                        "type": "string",
                        "title": "Одно из следующих значений: число (целое, дробное, строка), дата и время в UTC + указание смещение часового пояса, длительность, гиперссылка",
                        "enum": [
                            "integer",
                            "float",
                            "string",
                            "date",
                            "UTC",
                            "duration",
                            "link"
                        ]
                    },
                    "IS_DELETED": {
                        "type": "integer",
                        "title": "Является ли заголовок удаленным из системы (у старых объектов он остаётся)",
                        "default": 0,
                        "enum": [
                            0,
                            1
                        ]
                    },
```

```
                    "FORMAT": {
                        "type": "string",
                        "title": "Тип значения (с версии 1.2)",
                        "enum": [
                            "string",
                            "date",
                            "number"
                        ]
                    },
                    "IS_SYSTEM": {
                        "type": "integer",
                        "title": "Является ли заголовок системным (с версии
1.1)",
                        "default": 1,
                        "enum": [
                            0,
                            1
                        ]
                    },
                    "IS_MULTIPLE_VALUE": {
                        "type": "integer",
                        "title": "Может ли заголовок содержать несколько значений
(с версии 1.1)",
                        "default": 0,
                        "enum": [
                            0,
                            1
                        ]
                    },
                    "locale": {
                        "type": "array",
                        "title": "Локализация для заголовков (с версии 1.2)",
                        "items": {
                            "type": "object",
                            "title": "Локализация для заголовка",
                            "properties": {
                                "NAME": {
```

```
                                    "type": "string",
                                    "title": "Код атрибута, который будет
использоваться в заголовках события и для идентификации в xAPI/PushAPI"
                                },
                                "LANGUAGE": {
                                    "type": "string",
                                    "title": "Название языка",
                                    "enum": [
                                        "rus",
                                        "eng"
                                    ]
                                },
                                "DISPLAY_NAME": {
                                    "type": "string",
                                    "title": "Отображаемое в консоли ТМ
название атрибута"
                                }
                            }
                        }
                    }
                }
            },
            "meta": {
                "description": "Дополнительные данные",
                "type": "object",
                "properties": {
                    "totalCount": {
                        "type": "integer",
                        "description": "Общее количество ОЗ"
                    },
                    "start": {
                        "type": "integer",
                        "description": "С какой позиции началось отдаваться"
                    },
                    "limit": {
                        "type": "integer",
```

```
                    "description": "Сколько всего отдалось"
                }
            },
            "required": [
                "totalCount",
                "start",
                "limit"
            ]
        }
    },
    "required": [
        "data",
        "meta"
    ]
}
```

Пример:

Запрос

```
GET /xapi/event/header
X-API-Version: 1.1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
```

Ответ

```
{
    "data": [
        {
            "NAME": "device_type",
            "NOTE": "Тип устройства",
            "TYPE": "enum",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "destination_type",
            "NOTE": "Тип источника сканирования",
```

```
            "TYPE": "enum",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "destination_url",
            "NOTE": "URL",
            "TYPE": "string",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "print_server",
            "NOTE": "Имя сервера печати",
            "TYPE": "string",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "location",
            "NOTE": "Расположение принтера",
            "TYPE": "string",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "comment",
            "NOTE": "Комментарий к принтеру",
            "TYPE": "string",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
```

```
        {
            "NAME": "port_name",
            "NOTE": "Имя порта принтера",
            "TYPE": "string",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "copies",
            "NOTE": "Количество копий",
            "TYPE": "number",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "task_name",
            "NOTE": "Имя задания",
            "TYPE": "string",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "print_task_name",
            "NOTE": "Имя задания на печать",
            "TYPE": "string",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "task_run_date",
            "NOTE": "Дата запуска задания",
            "TYPE": "date",
            "IS_DELETED": 0,
```

```
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "create_date",
            "NOTE": "Дата создания файла",
            "TYPE": "date",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "modify_date",
            "NOTE": "Дата модификации файла",
            "TYPE": "date",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "source_ip",
            "NOTE": "IP адрес источника",
            "TYPE": "string",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "workstation_type",
            "NOTE": "Тип рабочей станции",
            "TYPE": "enum",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "application_path_from",
```

```
            "NOTE": "Путь к приложению из которого скопировано событие",
            "TYPE": "string",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "application_path_to",
            "NOTE": "Путь к приложению в которое скопировано событие",
            "TYPE": "string",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "terminal_session",
            "NOTE": "Признак терминальной сессии",
            "TYPE": "number",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "emitter",
            "NOTE": "Источник события",
            "TYPE": "enum",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
        },
        {
            "NAME": "charset",
            "NOTE": "Кодировка текста вложения",
            "TYPE": "string",
            "IS_DELETED": 0,
            "IS_SYSTEM": 1,
            "IS_MULTIPLE_VALUE": 0
```

```
        }
    ],
    "meta": {
        "totalCount": 20,
        "start": 0,
        "limit": 100
    }
}
```

Получение списка каталогов ОЗ

Описание : Список каталогов ОЗ, отсортированных по имени.

Метод: `GET`

Ресурс: `event/protectedCatalog`

Параметры:

- `start` - с какой позиции начать отдавать каталоги (по умолчанию 0)
- `limit` - сколько каталогов отдавать (по умолчанию 100, не может быть больше 1000)

Ответ: Список каталогов ОЗ, описанный следующей JSON-Schema:

JSON схема отдачи списка каталогов.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "data": {
            "type": "array",
            "description": "Массив каталогов ОЗ",
            "items": {
                "type": "object",
                "description": "Каталог ОЗ",
                "properties": {
                    "CATALOG_ID": {
                        "type": "string",
                        "description": "Идентификатор"
                    },
                    "DISPLAY_NAME": {
                        "type": "string",
                        "description": "Имя"
                    },
                    "CREATE_DATE": {
                        "type": "string",
                        "description": "Дата создания"
                    },
                    "IS_DELETED": {
                        "type": "integer",
                        "description": "Удален ли из активной конфигурации",
                        "enum": [
                            0,
```

```
                        1
                    ]
                }
            },
            "required": [
                "CATALOG_ID",
                "DISPLAY_NAME",
                "CREATE_DATE",
                "IS_DELETED"
            ]
        }
    },
    "meta": {
        "description": "Дополнительные данные",
        "type": "object",
        "properties": {
            "totalCount": {
                "type": "integer",
                "description": "Общее количество ОЗ"
            },
            "start": {
                "type": "integer",
                "description": "С какой позиции началось отдаваться"
            },
            "limit": {
                "type": "integer",
                "description": "Сколько всего отдалось"
            }
        },
        "required": [
            "totalCount",
            "start",
            "limit"
        ]
    }
},
"required": [
    "data",
    "meta"
]
}
```

Пример:

Запрос

```
GET /xapi/event/protectedCatalog?start=0&limit=10
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: aplab
X-API-Auth-Token: gla20dos1pcdbr69caxz
```

Ответ

```
{
    "data": [{
```

```
        "CATALOG_ID": "3D0F8CDAF9E74F82AC85EB32D12727A700000000",
        "DISPLAY_NAME": "Грифованная информация",
        "CREATE_DATE": "2016-05-12 10:24:41.342554",
        "IS_DELETED": 0
    }],
    "meta": {
        "totalCount": 1,
        "start": 0,
        "limit": 10
    }
}
```

Получение списка политик

Описание: Список политик отсортированный по имени.

Метод: `GET`

Ресурс: `event/policy`

Параметры:

- `start` — с какой позиции начать отдавать политики, по-умолчанию 0
- `limit` — сколько политик отдавать, по-умолчанию 100, не может быть больше 1000

Ответ: Список политик, описанный следующей **JSON схемой отдачи списка каталогов:**

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "data": {
            "type": "array",
            "description": "Массив политик",
            "items": {
                "type": "object",
                "description": "Политика",
                "properties": {
                    "POLICY_ID": {
                        "type": "string",
                        "description": "Идентификатор"
                    },
                    "DISPLAY_NAME": {
                        "type": "string",
                        "description": "Имя"
                    },
                    "CREATE_DATE": {
                        "type": "string",
                        "description": "Дата создания"
                    },
                    "IS_DELETED": {
                        "type": "integer",
                        "description": "Удалена ли из активной конфигурации",
                        "enum": [
                            0,
                            1
```

```
                        ]
                    }
                },
                "required": [
                    "POLICY_ID",
                    "DISPLAY_NAME",
                    "CREATE_DATE",
                    "IS_DELETED"
                ]
            }
        },
        "meta": {
            "description": "Дополнительные данные",
            "type": "object",
            "properties": {
                "totalCount": {
                    "type": "integer",
                    "description": "Общее количество политик"
                },
                "start": {
                    "type": "integer",
                    "description": "С какой позиции началось отдаваться"
                },
                "limit": {
                    "type": "integer",
                    "description": "Сколько всего отдалось"
                }
            },
            "required": [
                "totalCount",
                "start",
                "limit"
            ]
        }
    },
    "required": [
        "data",
        "meta"
    ]
}
```

Пример:

Запрос

```
GET /xapi/event/policy?start=0&limit=10
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
```

Ответ

```
{
    "data": [{
        "POLICY_ID": "3D0F8CDAF9E74F82AC85EB32D12727A700000000",
        "DISPLAY_NAME": "Грифованная информация",
        "CREATE_DATE": "2016-05-12 10:24:41.342554",
        "IS_DELETED": 0
    }],
    "meta": {
        "totalCount": 1,
        "start": 0,
        "limit": 10
    }
}
```


Получение списка событий

Описание: Список событий, отсортированных по времени перехвата (от новых к старым).

Метод: `GET`

Ресурс: `event`

Параметры:

- `filter` - список фильтрации:
  - `date[from]` — дата начала в формате UNIX-timestamp (можно использовать отдельно от `date[to]`)
  - `date[to]` — дата окончания в формате UNIX-timestamp (можно использовать отдельно от `date[from]`)
  - `insert_date[>]` — дата вставки начала в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][>]=1234.123312`
  - `insert_date[<]` — дата вставки окончания в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][<]=1234.123312`
  - `insert_date[>=]` — дата вставки начала в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][>=]=1234.123312` (включительно) (с версии API 1.1)
  - `insert_date[<=]` — дата вставки окончания в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][<=]=1234.123312` (включительно) (с версии API 1.1)
  - `rule_group_type[]` — группа правил (с версии API 1.1)
  - `violation_level[]` — уровень нарушения (с версии API 1.1)
  - `verdict[]` — вердикт (с версии API 1.1)
  - `object_type_code[]` — тип события (возможно несколько значений)
  - `protocol[]` — протокол (возможно несколько значений)
  - `user_decision[]` — решение пользователя (возможно несколько значений)
  - `document_id[]` — идентификатор документа идентификатор
  - `object_size[from]` — от скольки байт искать событие (можно использовать отдельно от `object_size [to]`)
  - `object_size[to]` — до скольки байт искать событие (можно использовать отдельно от `object_size [from]`)
  - `protected_catalog_id[]` — идентификатор каталога ОЗ (возможно несколько значений)
  - `tag_id[]` — идентификатор тега (возможно несколько значений)

- `with[]` - список дополнительных сущностей, которые должны быть добавлены к объекту (например: отправители, получатели и т.д.):
  - `protected_catalogs`
  - `protected_documents`
  - `policies`
  - `senders_keys`
  - `senders`
  - `recipients_keys`
  - `recipients`
  - `workstations_keys`
  - `workstations`
  - `tags`
  - `perimeters` (с версии API 1.1)
  - `attachments` (с версии API 1.1)
  - `files` (с версии API 1.1) — недокументированный параметр, отдающий все контенты, в которых непустое поле FILE_NAME
  - `contents_renderer` (с версии API 1.8) - прикрепить контенты рендерера различного типа
- `without[] – дополнительные флаги`
  - `count` — не считать `totalCount` (расчет `totalCount` может занимать больше половины времени запроса, поэтому лучше воспользоваться методом `event/stat`)
- `headers[]` — список дополнительных полей (заголовков) события (можно написать *, если нужны все). Список можно получить с помощью метода `event/header (с версии API 1.1)`
- `sort[]` — недокументированное поле, позволяющее сортировать результат (получение событий замедляется существенно, для получения последнего/первого события лучше использовать метод `event/stat`. Возможные ключи:
  - `capture_date — asc` (с версии API 1.1) `/desc`, например: `?sort[capture_date]=desc`
  - `insert_date` — `asc/desc`, например: `?sort[insert_date]=asc (с версии API 1.1)`
- `start` - с какой позиции начать отдавать события (по умолчанию 0)
- `limit` - сколько событий отдавать (по умолчанию 100, не может быть больше 1000)

Алгоритм при первом получении событий:

1. Определить, с какого времени нужно получать события. Например, получить из статистики событий (GET /xapi/event/stat) интервал INSERT_DATE.
2. Разделить интервал на меньшее из значений: количество потоков (но не более 32) **или** количество ядер на сервере.
3. В каждом потоке составить и запустить первый запрос на получение списка события вида:
   GET /xapi/event?without[]=count&filter[insert_date][>=]={минимум_интервала}&filter[insert_date][<=]={максимум_интервала}&sort[insert_date]=asc&limit=1000
4. В полученном списке запомнить все OBJECT_ID.
5. В полученном списке найти последнюю INSERT_DATE и использовать её в запросе из пункта 3, в качестве `минимум_интервала.`

6. В полученном новом списке исключить события, которые запомнили в предыдущей итерации.
7. Если не дошли до `максимум_интервала`, снова начать с пункта 4.
8. Сохранить в персистентное хранилище:
    a. максимальное значение INSERT_DATE, полученное из пункта 1.
    b. список OBJECT_ID из интервала, в который входит максимальное значение INSERT_DATE.

Примеры:

Получить первые 100 событий с за январь 2015 года:

```
GET /xapi/event?filter[date][from]=1420059600&filter[date][to]=1422737999&start=0&limit=100
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: aplab
X-API-Auth-Token: gla20dos1pcdbr69caxz
```

Получить 1000 событий с письмами, содержащими контент:

```
GET /xapi/event?with=contents_renderer&limit=1000
X-API-Version: 1.8
X-API-CompanyId: IW
X-API-productName: Vision
X-API-Auth-Token: 1eja07v23vs9k1s5ij5k
```

Ответ содержит список событий, отсортированный по времени перехвата от новых к старым. Для разных типов событий структура ответа будет отличаться.

Пример ответа

```
[
 OBJECT_ID: 1
 VIOLATION_LEVEL: 'High'
 CAPTURE_DATE: '12-03-2022 15:02:25'
 contents_renderer:
 [
    SUBJECT: [
       CONTENT_ID: 1
       OBJECT_ID: 1
       CONTENT_SIZE: 26
   ]
   BODY: [
      CONTENT_ID: 2
      OBJECT_ID: 1
      CONTENT_SIZE:5125
   ]
 ]
]
```

Получение списка тегов

Описание: Список тегов, отсортированных по имени.

Метод: `GET`

Ресурс: `event/tag`

Параметры:

- `start` - с какой позиции начать отдавать теги (по умолчанию 0)
- `limit` - сколько тегов отдавать (по умолчанию 100, не может быть больше 1000)

Ответ: Список тегов, описанный следующей JSON-Schema:

JSON схема отдачи списка тэгов

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {
       "data": {
           "type": "array",
           "description": "Массив тегов",
           "items": {
               "type": "object",
               "description": "Тег",
               "properties": {
                   "TAG_ID": {
                       "type": "string",
                       "description": "Идентификатор"
                   },
                   "DISPLAY_NAME": {
                       "type": "string",
                       "description": "Имя"
                   },
                   "COLOR": {
                       "type": "string",
                       "description": "Цвет тега в HEX RGP, например #ffffff"
                   },
                   "CREATE_DATE": {
                       "type": "string",
                       "description": "Дата создания"
                   },
                   "IS_DELETED": {
                       "type": "integer",
                       "description": "Удален ли из активной конфигурации"
                   },
                   "IS_SYSTEM": {
                       "type": "integer",
                       "description": "Является ли тег системным"
                   }
               },
               "required": [
                   "TAG_ID",
                   "DISPLAY_NAME",
                   "COLOR",
                   "CREATE_DATE",
```

```
                    "IS_DELETED",
                    "IS_SYSTEM"
                ]
            }
        },
        "meta": {
            "description": "Дополнительные данные",
            "type": "object",
            "properties": {
                "totalCount": {
                    "type": "integer",
                    "description": "Общее количество ОЗ"
                },
                "start": {
                    "type": "integer",
                    "description": "С какой позиции началось отдаваться"
                },
                "limit": {
                    "type": "integer",
                    "description": "Сколько всего отдалось"
                }
            },
            "required": [
                "totalCount",
                "start",
                "limit"
            ]
        }
    },
    "required": [
        "data",
        "meta"
    ]
}
```

Пример:

Запрос

```
GET /xapi/event/tag?start=0&limit=10
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: aplab
X-API-Auth-Token: gla20dos1pcdbr69caxz
```

Ответ

```
{
    "data": [{
        "TAG_ID": "0AF69DD5742E6977E0530100007F2BB700000000",
        "DISPLAY_NAME": "VIP",
        "COLOR": "#4e954e",
        "CREATE_DATE": "2016-05-12 10:24:40.002673",
        "IS_DELETED": 0,
        "IS_SYSTEM": 1
    }],
```

```
    "meta": {
        "totalCount": 1,
        "start": 0,
        "limit": 10
    }
}
```

Получение статистики по событиям

Описание: Недокументированный метод получения статистики по событиям, отдаёт только статистику по дате перехвата.

Метод: `GET`

Ресурс: `event/stat`

Параметры:

- fields — список полей, которые нужно выбрать: CAPTURE_DATE , INSERT_DATE , totalCount (с версии API 1.2)
- `filter` — список фильтрации:
  - `date[from]` — дата начала в формате UNIX-timestamp (можно использовать отдельно от `date[to]`)
  - `date[to]` — дата окончания в формате UNIX-timestamp (можно использовать отдельно от `date[from]`)
  - `insert_date[>]` — дата вставки начала в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][>]=1234.123312` (с версии API 1.1)
  - `insert_date[<]` — дата вставки окончания в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][<]=1234.123312` (с версии API 1.1)
  - `insert_date[>=]` — дата вставки начала в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][>=]=1234.123312` (включительно) (с версии API 1.1)
  - `insert_date[<=]` — дата вставки окончания в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][<=]=1234.123312` (включительно) (с версии API 1.1)
  - `object_type_code[]` — тип события (возможно несколько значений) (с версии API 1.1)
  - `violation_level[]` — уровень нарушения (с версии API 1.1)

Ответ: Статистика по событиям, описанная следующей JSON-Schema:

JSON схема отдачи:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "data": {
            "description": "Пустая секция",
            "items": {},
            "type": "array"
        },
        "meta": {
            "description": "Дополнительные данные",
```

```
            "type": "object",
            "properties": {
                "CAPTURE_DATE": {
                    "description": "Статистика по полю CAPTURE_DATE",
                    "type": "object",
                    "properties": {
                        "from": {
                            "type": "string",
                            "description": "Дата перехвата первого события в выборке"
                        },
                        "to": {
                            "type": "string",
                            "description": "Дата перехвата последнего события в
выборке"
                        }
                    }
                },
                "INSERT_DATE": {
                    "description": "Статистика по полю INSERT_DATE (с версии API 1.1)",
                    "type": "object",
                    "properties": {
                        "from": {
                            "type": "string",
                            "description": "Дата вставки первого события в выборке"
                        },
                        "to": {
                            "type": "string",
                            "description": "Дата вставки последнего события в выборке"
                        }
                    }
                },
                "totalCount": {
                    "type": "integer",
                    "description": "Общее количество событий"
                }
            }
        }
    }
}
```

Пример:

Запрос

```
GET /xapi/event/stat?filter[data][from]=0
X-API-Version: 1.3
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
```

Ответ

```
{
```

```
    "data": [],
    "meta": {
        "CAPTURE_DATE": {
            "from": "2017-04-16 00:00:04",
            "to": "2017-05-16 16:20:36"
        },
        "INSERT_DATE": {
            "from": "2017-04-16 00:01:07",
            "to": "2017-05-16 16:20:40"
        },
        "totalCount": 225075
    }
}
```


Изменение атрибутов событий

Назначение вердикта пользователя
Версия API: ≥ 1.2

Метод: PATCH

Ресурс: event/{id}/userDecision?tbsId={tbsId}

Описание:

Метод назначает решение пользователя. Поля:

- `{id}` — ID события (`OBJECT_ID`)
- `{tbsId}` — Табличное пространство события (`TBS_ID`)
- `USER_DECISION` — Решение пользователя. Возможные значения:
    - `NotProcessed - Решение не принято`
    - `Violation - Нарушение`
    - `NoViolation - Нет нарушения`
    - `AdditionalProcessingNeeded - Требует дополнительной обработки`
- `NOTE –` комментарий, который сохраняется в разделе Аудит. Может быть заполнено, например, именем пользователя сторонней системы, который изменил значение атрибута. Длина поля не должна превышать 1000 символов.

Вызов данного API записывается в раздел Аудит. При назначении решения пользователя в событии могут автоматически поменяться следующие поля:

- `TBS_ID`
- `VERDICT`
- `FORWARD_STATE_CODE`

Поэтому в ответе API отдаются значения, которые можно будет обновить у вашей копии события.

Тело запроса:JSON схема назначения вердикта пользователя


```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "title": "Set user decision schema",
  "required": [
```

```
    "USER_DECISION"
  ],
  "properties": {
    "USER_DECISION": {
      "$id": "#/properties/USER_DECISION",
      "type": "string",
      "title": "User decision",
      "enum": [
        "NotProcessed",
        "Violation",
        "NoViolation",
        "AdditionalProcessingNeeded"
      ]
    },
    "NOTE": {
      "$id": "#/properties/NOTE",
      "type": "string",
      "title": "Note",
      "maxLength": 1000
    }
  }
}
```

Тело ответа: JSON схема ответа на назначение вердикта пользователя

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "title": "Result setting user decision schema",
  "required": [
    "OBJECT_ID",
    "TBS_ID",
    "VERDICT",
    "USER_DECISION",
    "FORWARD_STATE_CODE"
  ],
  "properties": {
    "OBJECT_ID": {
      "$id": "#/properties/OBJECT_ID",
      "type": "integer",
      "title": "ID События"
    },
    "TBS_ID": {
      "$id": "#/properties/TBS_ID",
      "type": "integer",
      "title": "ID табличного пространства"
    },
    "VERDICT": {
      "$id": "#/properties/VERDICT",
      "type": "string",
      "title": "Вердикт",
      "enum": [
        "Allowed",
        "Quarantined",
        "Forbidden"
```

```
      ]
    },
    "USER_DECISION": {
      "$id": "#/properties/USER_DECISION",
      "type": "string",
      "title": "Проставленное решение пользователя",
      "enum": [
        "NotProcessed",
        "NoViolation",
        "Violation",
        "AdditionalProcessingNeeded"
      ]
    },
    "FORWARD_STATE_CODE": {
      "$id": "#/properties/FORWARD_STATE_CODE",
      "type": "string",
      "title": "Статус отправки",
      "examples": [
        "Sent",
        "NotSent",
        "Pending"
      ]
    }
  }
}
```

Пример: Назначение вердикта "Нарушение" с комментарием событию ID 194 в TBS 1 с помощью cUrl:

```
curl 'https://example.com/xapi/event/195/userDecision?tbsId=1' \
  -X PATCH \
  -H 'X-API-Auth-Token: 1792jmclf7fer1ikuhby' \
  -H 'X-API-Version: 1' \
  -H 'X-API-CompanyId: IW' \
  -H 'X-API-ImporterName: dome' \
  -H 'Content-Type: application/json' \
  --data-binary '{"USER_DECISION":"Violation","NOTE":"Назначение вердикта через Dome.
Сотрудник Иванов Виктор Павлович."}'
```

Назначение вердикта пользователя множеству событий
Версия API: ≥ 1.2

Метод: PATCH

Ресурс: event/userDecision

Описание:

Метод позволяет массово назначать решения пользователя (принимает не более **1000** решений пользователя, иначе возвращается сообщение об ошибке с кодом **413**). ID события и ID табличного пространства передаются в теле запроса с решением пользователя и комментарием.

Тело запроса: JSON схема для множественного назначения вердиктов

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "array",
```

```
  "title": "Set batch user decision schema",
  "items": {
    "$id": "#/items",
    "type": "object",
    "required": [
      "OBJECT_ID",
      "TBS_ID",
      "USER_DECISION"
    ],
    "properties": {
      "OBJECT_ID": {
        "$id": "#/items/properties/OBJECT_ID",
        "type": "integer",
        "title": "ID События"
      },
      "TBS_ID": {
        "$id": "#/items/properties/TBS_ID",
        "type": "integer",
        "title": "ID табличного пространства"
      },
      "USER_DECISION": {
        "$id": "#/items/properties/USER_DECISION",
        "type": "string",
        "title": "User decision",
        "enum": [
          "NotProcessed",
          "Violation",
          "NoViolation",
          "AdditionalProcessingNeeded"
        ]
      },
      "NOTE": {
        "$id": "#/items/properties/NOTE",
        "type": "string",
        "title": "Note",
        "maxLength": 1000
      }
    }
  }
}
```

Тело ответа: JSON схема для множественного назначения вердиктов

```
 {
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "array",
  "title": "Array result",
  "items": {
    "$id": "#/items",
    "type": "object",
    "title": "Result setting user decision schema",
    "required": [
      "OBJECT_ID",
      "TBS_ID",
      "VERDICT",
```

```
      "USER_DECISION",
      "FORWARD_STATE_CODE"
    ],
    "properties": {
      "OBJECT_ID": {
        "$id": "#/items/properties/OBJECT_ID",
        "type": "integer",
        "title": "ID События"
      },
      "TBS_ID": {
        "$id": "#/items/properties/TBS_ID",
        "type": "integer",
        "title": "ID табличного пространства"
      },
      "VERDICT": {
        "$id": "#/items/properties/VERDICT",
        "type": "string",
        "title": "Вердикт",
        "enum": [
          "Allowed",
          "Quarantined",
          "Forbidden"
        ]
      },
      "USER_DECISION": {
        "$id": "#/items/properties/USER_DECISION",
        "type": "string",
        "title": "Проставленное решение пользователя",
        "enum": [
          "NotProcessed",
          "NoViolation",
          "Violation",
          "AdditionalProcessingNeeded"
        ]
      },
      "FORWARD_STATE_CODE": {
        "$id": "#/items/properties/FORWARD_STATE_CODE",
        "type": "string",
        "title": "Статус отправки",
        "examples": [
          "Sent",
          "NotSent",
          "Pending"
        ]
      }
    }
  }
}
```

Пример:

```
curl 'https://example.com/xapi/event/userDecision' \
  -X PATCH \
  -H 'X-API-Auth-Token: 1792jmclf7fer1ikuhby' \
  -H 'X-API-Version: 1' \
```

```
  -H 'X-API-CompanyId: IW' \
  -H 'X-API-ImporterName: dome' \
  -H 'Content-Type: application/json' \
  --data-binary '[{"OBJECT_ID": 1, "TBS_ID": 1,
"USER_DECISION":"Violation","NOTE":"Проставление вердикта через Dome. Сотрудник Иванов
Виктор Павлович."},{"OBJECT_ID": 24, "TBS_ID": 2,
"USER_DECISION":"AdditionalProcessingNeeded"}]'
```

Получение списка пользователей консоли

Описание: Список пользователей консоли Traffic Monitor (включая бывших сотрудников)

Метод: `GET`

Ресурс: `user`

Параметры (передаются в виде JSON):

- `start` - с какой позиции начать отдавать данные пользователей (по умолчанию 0)
- `limit` - сколько комплектов данных пользователей отдавать (по умолчанию 100, не может быть больше 1000)

JSON-схема отдачи списка пользователей

```yaml
schemas:
    user:
      type: object
      additionalProperties: false
      properties:
        USER_ID:
          type: integer
          description: User ID
        USERNAME:
          type: string
          description: Login
        DISPLAY_NAME:
          type: string
          description: Full name
        STATUS:
          type: integer
          description: Status (0 - active, 1 - disabled, 2 - deleted)
          enum: [0, 1, 2]
        CREATE_DATE:
          type: string
          description: Create Date with format "Y-m-d H:i:s.u" in UTC
        EMAIL:
          type: string
          description: E-mail
        NOTE:
          type: string
          description: Note
```

Пример:

Запрос

```
curl -X GET "https://qa-b2a1.infowatch.ru/xapi/user?start=1&limit=100" -H "accept: application/json" -H "X-Enable-OpenAPI: 1"
```

Ответ


```
{
  "data": [
    {
      "USER_ID": 0,
      "USERNAME": "string",
      "DISPLAY_NAME": "string",
      "STATUS": 0,
      "CREATE_DATE": "string",
      "EMAIL": "string",
      "NOTE": "string"
    }
  ],
  "meta": {
    "totalCount": 0,
    "start": 0,
    "limit": 0
  }
}
```


Получение результатов выполнения запроса Traffic Monitor

Получение списка запросов

Описание: Список запросов Traffic Monitor

Метод: `GET`

Ресурс: `query`

Параметры (передаются в виде JSON):

- `userId` - id пользователя, список запросов которого требуется получить
- `start` - с какой позиции начать отдавать запросы (по умолчанию 0)
- `limit` - сколько запросов отдавать (по умолчанию 100, не может быть больше 1000)

По запросу будет предоставлен список уже существующих в TM запросов с указанием ID запроса и его текстовым именем.
JSON-схема отдачи списка запросов:


```
 "data": [
  {
    "QUERY_ID": 0,
    "DISPLAY_NAME": "string"
  }
],
"meta": {
  "totalCount": 0,
  "start": 0,
```

```
    "limit": 0
  }
```

Пример запроса:

```
curl -X GET "https://server.company.ru/xapi/query?userId=5&start=0&limit=10" -H "accept: application/json"
 -H "X-Enable-OpenAPI: 1"
```

Запуск запроса на выполнение

Описание: Запуск запроса на выполнение в Traffic Monitor

Метод: `POST`

Ресурс: `query/{queryId}/exec`

Параметры:

- `queryId` - id запроса, который требуется запустить

JSON-схема ответа на запуск запроса:

```
{
  "data": {
    "QUERY_ID": 0,
    "QUERY_TYPE": "query",
    "DISPLAY_NAME": "string",
    "CREATE_DATE": "string",
    "CHANGE_DATE": "string"
  }
}
```

Пример запроса:

```
curl -X POST "https://server.conpany.ru/xapi/query/0/exec" -H "accept: application/json" -H "X-Enable-
OpenAPI: 1"
```

Получение статуса запроса

Описание: Получение статуса запроса

Метод: `GET`

Ресурс: `query/{queryId}/status`

Параметры:

- `queryId` - id запроса, статус которого требуется получить

JSON-схема ответа:

```
{
  "data": {
    "QUERY_ID": 0,
    "USER_ID": 0,
    "STATUS": 0,
    "RUN_DATE": "string",
    "COMPLETE_DATE": "string",
    "PERCENT_COMPLETE": 0,
    "OBJECT_COUNT": 0
  }
}
```

Пример запроса:

```
curl -X GET "https://server.company.ru/xapi/query/0/status" -H "accept: application/json" -H "X-Enable-OpenAPI: 1"
```

Получение результатов запроса

Описание: Список с результатами запросов Traffic Monitor

Метод: `GET`

Ресурс: `query/{queryId}/result`

Параметры (передаются в виде JSON):

- `queryId` - id запроса;
- `start` - с какой позиции начать отдавать запросы (по умолчанию 0)
- `limit` - сколько запросов отдавать (по умолчанию 100, не может быть больше 1000)
- `fields:`
  - `OBJECT_ID –` идентификатор события в ТМ (положительное целое число);
  - `TBS_ID –` идентификатор табличного пространства (положительное целое число).

В ответ на ранее инициированный запрос (см. "Запуск запроса на выполнение") будет предоставлен список событий с указанием их ID.

JSON-схема ответа:

```
{
  "data": [
```

```
    {
      "OBJECT_ID": 0,
      "TBS_ID": 0
    }
  ],
  "meta": {
    "totalCount": 0,
    "start": 0,
    "limit": 0
  }
}
```


Пример запроса:

```
curl -X GET "https://server.company.ru/xapi/query/5/result?start=0&limit=10" -H "accept: application/json"
 -H "X-Enable-OpenAPI: 1"
```

10 Приложение A. Рекомендации по составлению имен и паролей

Требования к именам пользователей

- Длина имени пользователя должна составлять от 1 до 20 символов.
- Имя пользователя должно состоять из букв латинского алфавита, цифр и символа подчеркивания «_».
- Имя пользователя должно начинаться с буквы.

Требования к паролям пользователей

- Длина пароля может составлять от 8 до 128 символов.
- Пароль пользователя должен состоять только из букв латинского алфавита, цифр и символов: «#», «$», «!» или «%».
- Пароль чувствителен к регистру символов.

Рекомендации по составлению надежных паролей

- Рекомендуемая длина пароля: от 10 до 30 символов.
- Рекомендуемый пароль должен представлять собой смешанный набор букв, цифр и символов.
- Не рекомендуется:
    - включать в состав пароля слова и словосочетания;
    - включать в состав пароля несколько идущих подряд одинаковых символов;
    - начинать и заканчивать пароль одним и тем же символом;
    - создавать новый пароль путем добавления символов к текущему паролю.

11 Приложение B. Индикаторы мониторинга

В приведенной ниже таблице перечислены все индикаторы, используемые подсистемой мониторинга для проверки состояния компонентов, установленных и работающих на серверах Системы в режиме установки "Все-в-одном". Под проверкой подразумевается периодическое получение значения индикатора и сравнение значения индикатора с пороговым значением. Если все индикаторы показывают критические значения, это может свидетельствовать о физической недоступности проверяемых серверов. Следует проверить их работоспособность.

Важно!

Если производились изменения конфигурационных файлов, то пороговые значения и период проверки могут отличаться от указанных.

Важно!

Если действия, рекомендованные в таблице, не привели к решению проблемы, обратитесь в службу технической поддержки InfoWatch по адресу support@infowatch.com .

Название индикатора	**Что проверяется и где проверяется**	**Периодичность опроса**	**Значения индикатора**	Подробности и рекомендации
Общая нагрузка системы	Загрузка серверов Linux-серверы, TMcap-серверы, серверы БД за последние 15, 5, 1 минуту	30 мин.	🔴 – хотя бы одна цифра превышает соответствующее значение 20.0,16.0,12.0 за 15, 5, 1 🟡 – хотя бы одна цифра превышает соответствующее значение 10.0,8.0,6.0 за 15, 5, 1 🟢 – ни одна цифра не превышает соответствующее значение 10.0,8.0,6.0 за 15, 5, 1	Проверьте загруженность серверов перехвата и анализа, серверов БД: 1. Выполните: `top` 2. Отсортируйте результаты: a. Shift+m - по загрузке ОЗУ; b. Shift+p - по загрузке ЦП.

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации
Состояние компонентов Системы	Статусы служб системы	30 мин.	🔴 – хотя бы один сервис не запущен 🟢 – все сервисы запущены	Состав и количество служб Системы в режиме установки "Все-в-одном" известен и постоянен. Чтобы вывести на экран подробности о состоянии служб, введите: `iwtm status`
Состояние службы syslog	Запущен или нет сервис syslog	30 мин.	🔴 – не запущен 🟢 – запущен	Проверьте запуск службы rsyslog: `systemctl status rsyslog` Дополнительная информация в ст. "Использование демона rsyslogd для логирования сообщений от служб iwtm"
Количество активных пользователей	Количество пользователей (Офицеров безопасности), зарегистрированных в Системе (Linux-серверы)	30 мин.	🔴 – > 50 пользователей 🟡 – от 20 до 50 🟢 – < 20 пользователей	Необходимо уменьшить количество ОБ до 50 человек
Ошибки в журнале предупреждений БД	Наличие ошибок в журнале предупреждений БД (Серверы БД)	5 мин.	🔴 – есть ошибки 🟢 – нет ошибок	Необходимо снизить количество ошибок с SEVERITY=HIGH. **Поиск в pg_log должен осуществляться по наличию слов:** • FATAL • PANIC • ERROR
Доступность сервера	Время ответа на посылаемые пакеты и количество потерянных	5 мин.	🔴 – > 500 мс 🟡 – от 100 до 500 мс	Пинг заданных серверов с параметрами `100.0,20%! 500.0,60%` означает,

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации
	пакетов (Linux-серверы)		🟢 – < 100 мс	что ответ на посылаемые пакеты составляет больше 100 миллисекунд и теряется больше чем 20% пакетов, более 500 миллисекунд и теряется 60% пакетов.
Доступность встроенного агента передачи почты	Доступность сервера Exim	30 мин.	🔴 – не доступен 🟢 – доступен	1. Проверьте состояние сервера Exim: `systemctl status exim4` 2. Запустите проверку конфигурационного файла */etc/exim4/exim4.conf* на правильность: `exim4 check`
Доступность сервера по SSH	Время ответа на запрос	10 мин.	🔴 – >= 10 с 🟢 – < 10 с	Проверьте состояние демона sshd: `systemctl status sshd`
Доступность сервера Device Monitor	Ответа на запрос	30 мин.	🔴 – нет соединения 🟢 – есть соединение	Должно отображаться, если в Системе установлена активная лицензия на Device Monitor. Проверьте наличие лицензии в разделе **Управление->Лицензии** в консоли управления Traffic Monitor
Использование файла подкачки	Количество свободного места в swap (виртуальная	5 мин.	🔴 – свободно < 10%	Подробнее см. "Рекомендации по разбиению дискового пространства

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации
	память). (Linux-серверы)		🟡 – свободно от 50% до 10% 🟢 – свободно > 50%	серверов при установке в разных режимах"
Свободное место в корневой партиции	Количество свободного места в корневой файловой системе. (Linux-серверы)	30 мин.	🔴 – < 10 000 Мб 🟡 – от 10 000 МБ до 20 000 Мб 🟢 – > 20 000 Мб	
Свободное место в партиции /var	Количество свободного места в файловой системе `/var.(` Linux-серверы)	30 мин.	🔴 – < 10 000 Мб 🟡 – от 10 000 Мб до 20 000 Мб 🟢 – > 20 000 Мб	
Состояние службы синхронизации времени	Статус сервиса ntpd. (NTPD)	30 мин.	🔴 – NTP сервер не доступен 🟢 – NTP сервер доступен	Проверьте состояние демона ntpd: `systemctl status ntpd`
Отклонение системного времени	Лаг времени на серверах Системы и сервере NTP. (Linux-серверы)	360 мин.	🔴 – > 40 с 🟡 – от 20 до 40 с 🟢 – < 20 с	Данная ошибка возникает, если Системе не удалось установить соединение с NTP-серверами или в конфигурации Системы не указано ни одного NTP-сервера. В этом случае появится сообщение об ошибке "*Can't create socket connection*". Убедитесь, что в конфигурационных файлах *ntp.conf* и *iwmon-services-ntp.cfg* указан один и тот же корректный NTP-сервер.

Название индикатора	**Что проверяется и где проверяется**	**Периодичность опроса**	**Значения индикатора**	Подробности и рекомендации
Наличие дампов памяти	Количество файлов в каталоге /opt/iw/tm5, начинающихся с core_.... (Linux-серверы)	30 мин.	🔴 – есть дамп 🟢 – нет дампов	Обратитесь в службу технической поддержки InfoWatch по адресу support@infowatch.com

Очередь ошибок обработки событий	Количество объектов в очереди **queue/final-errors** (iw_rammer)	1 мин.	🔴 – > 1
			🟢 – 0
				Вариант 1: Очистите очереди, используя команды:
				`su - iwtm`
				`/opt/iw/tm5/bin/iw_qtool erase /opt/iw/tm5/queue/final-errors`
				`exit`
				Вариант 2:
				1. Откройте на редактирование файл `/etc/cron.d/iwtm_cleanup.`
				2. Уменьшите период очистки файлов в папках файловой очереди в штатном задании cron (значение по умолчанию - 30 дней).
				3. Сохраните изменения и закройте файл.
				
				Для получения дополнительной информации см. "Каким образом можно дослать письма, если они попали в очереди ошибок?"

Название индикатора	**Что проверяется и где проверяется**	**Периодичность опроса**	**Значения индикатора**	Подробности и рекомендации
Состояние базы данных	Статус базы данных (в зависимости от используемой в системе СУБД)	10 мин.	🔴 – <> ok 🟢 – ok	Проверьте, доступна ли БД: 1. Зайдите в БД: `sqlplus iwtm@iwtm` 2. Введите пароль: xxXX1234 3. Наблюдайте: `SQL>`
Свободное место в основном каталоге БД	Свободное место на партиции хранения основной информации (По умолчанию: `/u01`)	30 мин.	🔴 – < 10 000 Мб 🟡 – от 10000 до 20 000 Мб 🟢 – > 20 000 Мб	Выделите больше места на диске для `/u01` (подробнее см. "Рекомендации по разбиению дискового пространства серверов при установке в разных режимах")
Свободное место в каталогах событий БД	Свободное место на партициях хранения событий (По умолчанию: `/u02`)	30 мин.	🔴 – < 7% 🟡 – от 7 до 15% (если для хранения ежедневных ТП используются несколько разделов, то при значении < 7% хотя бы в одном разделе) 🟢 – > 15%	Если для хранения ежедневных табличных пространств используется несколько дисковых разделов, то свободное место рассчитывается как сумма всех разделов, выделенных под ежедневные ТП на этапе установки Подробнее см. "Рекомендации по разбиению дискового пространства серверов при установке в разных режимах"
Свободное место в каталоге ежедневных	Свободное место в партиции	30 мин.	🔴 – < 7% 🟡 – от 7 до 15%	Индикатор добавляется, если при установке был

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации
табличных пространств на быстром диске	хранения ежедневных табличных пространств на быстром диске.		🟢 – > 15%	выбран режим хранения данных *Быстрые и медленные диски* Подробнее см. "Рекомендации по разбиению дискового пространства серверов при установке в разных режимах"
Размер журнала предупреждений БД	Размер журнала предупреждений для СУБД PostgreSQL: System Log	5 мин.	🔴 – > 3 Гб 🟡 – от 1 Гб до 3 Гб 🟢 – < 1 Гб	Обратитесь в службу технической поддержки InfoWatch по адресу support@infowatch.com
Наличие ошибок в журнале базы данных	Наличие ошибок	30 мин.	🔴 – > 1 🟡 – 1 🟢 – нет ошибок	Необходимо снизить количество ошибок с SEVERITY=Error
Очередь обработки почтового трафика	Количество объектов в очереди **queue/smtp** (iw_messed)	30 мин.	🔴 – > 10000 объектов 🟡 – от 1000 до 10000 объектов 🟢 – < 1000 объектов	1. Проверьте статус службы: `systemctl status iw_messed` 2. Просмотрите логи службы в `opt/iw/tm5/log/messed_app.log` 3. Проверьте нагрузку на оперативную память и swap: a. для краткого вывода выполните: `free -m` b. для расширенного вывода

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации
				выполните: `cat /proc/meminfo`
Очередь на контентный анализ для xAPI/pushAPI-объектов	Количество объектов в очереди **queue/analysis** (iw_analysis)	30 мин.	🔴 – > 10000 объектов 🟡 – от 1000 до 10000 объектов 🟢 – < 1000 объектов	1. Проверьте статус службы: `systemctl status iw_analysis` 2. Просмотрите логи службы в `opt/iw/tm5/log/analysis.log` 3. Проверьте нагрузку на оперативную память и swap: a. для краткого вывода выполните: `free -m` b. для расширенного вывода выполните: `cat /proc/meminfo`
Очередь загрузки объектов от перехватчиков в сервис **iw_x2x**	Количество объектов от перехватчиков для **iw_x2x** в очереди **queue/db** (iw_x2x)	30 мин.	🔴 – > 10000 объектов 🟡 – от 1000 до 10000 объектов 🟢 – < 1000 объектов	1. Проверьте статус службы: `systemctl status iw_x2x` 2. Просмотрите логи службы в `opt/iw/tm5/log/x2x.log` 3. Проверьте нагрузку на оперативную память и swap: a. для краткого вывода

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации
				выполните: `free -m` b. для расширенного вывода выполните: `cat /proc/meminfo`
Очередь загрузки в хранилище	Количество объектов в очереди **queue/x2x** на загрузку в хранилище сервисом **iw_x2db** (TMcap-серверы)	30 мин.	🔴 – > 10000 объектов 🟡 – от 1000 до 10000 объектов 🟢 – < 1000 объектов	1. Проверьте статус службы: `systemctl status iw_x2db` 2. Просмотрите логи службы в `opt/iw/tm5/log/x2db.log` 3. Проверьте нагрузку на оперативную память и swap: a. для краткого вывода выполните: `free -m` b. для расширенного вывода выполните: `cat /proc/meminfo`
Очередь обработки действий от примененных политик	Количество объектов в очереди **queue/blackboard** (Linux-серверы)	30 мин.	🔴 – > 1000 объектов 🟡 – от 200 до 1000 объектов 🟢 – < 200 объектов	1. Проверьте статус службы: `systemctl status iw_analysis` 2. Просмотрите логи службы в `opt/iw/tm5/log/analysis.log` 3. Проверьте нагрузку на

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации
				оперативную память и swap: a. для краткого вывода выполните: `free -m` b. для расширенного вывода выполните: `cat /proc/meminfo`
Очередь объектов на индексацию в Sphinx для полнотекстового поиска	Количество объектов в очереди на индексацию в Sphinx для полнотекстового поиска	30 мин.	🔴 – > 10000 объектов 🟡 – от 1000 до 10000 объектов 🟢 – < 1000 объектов	1. Проверьте статус службы: `searchd --status -c /etc/infowatch/sphinx.conf` 2. Просмотрите логи службы в `opt/iw/tm5/log/sphinx/` 3. Проверьте нагрузку на оперативную память и swap: a. для краткого вывода выполните: `free -m` b. для расширенного вывода выполните: `cat /proc/meminfo` 4. Вычислите количество запущенных экземпляров `searchd`:

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации
				a. если на ноде запущен iw_is, то: 1 (сам iw_is)+ 4 (grep max_children /opt/iw/tm5/etc/is.conf)=5 b. если на ноде запущен iw_indexer, то: 1 (сам iw_ndexer)+ 8 (grep max_children /opt/iw/tm5/etc/sphinx.conf)=9 c. Итого: a+b=14 экземпляров searchd. Чем больше процессов, тем меньше вероятность длительного выполнения поискового запроса пользователя.
Очередь объектов на индексацию в Sphinx для поиска по метаинформации	Количество объектов в очереди на индексацию в Sphinx для поиска по метаинформации	30 мин.	🔴 – > 10000 объектов 🟡 – от 1000 до 10000 объектов 🟢 – < 1000 объектов	1. Проверьте статус службы: searchd --status -c /etc/infowatch/sphinx.conf 2. Просмотрите логи службы в

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации
				`opt/iw/tm5/log/sphinx/` 3. Проверьте нагрузку на оперативную память и swap: a. для краткого вывода выполните: `free -m` b. для расширенного вывода выполните: `cat /proc/meminfo`
Количество ошибок индексации событий службой **iw_indexer**	Количество ошибок индексации (iw_indexer)	30 мин.	🔴 – > 100 ошибок 🟡 – от 10 до 100 🟢 – 10	1. Проверьте статус службы: `systemctl status iw_indexer` 2. Просмотрите логи службы в `opt/iw/tm5/log/indexer.log` 3. Проверьте нагрузку на оперативную память и swap: a. для краткого вывода выполните: `free -m` b. для расширенного вывода выполните: `cat /proc/meminfo`
Количество ошибок индексации	Количество ошибок индексации	30 мин.	🔴 – > 100 ошибок	1. Проверьте статус службы:

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации
метаинформации службой **iw_is**	метаинформации службой **iw_is**		🟡 – от 10 до 100 ошибок 🟢 – 10 ошибок	`systemctl status iw_is` 2. Просмотрите логи службы в `opt/iw/tm5/log/is.log` 3. Проверьте нагрузку на оперативную память и swap: a. для краткого вывода выполните: `free -m` b. для расширенного вывода выполните: `cat /proc/meminfo`
Очередь команд **iw_is**	Количество команд в очереди службы **iw_is**	30 мин.	🔴 – > 100 команд 🟡 – от 50 до 100 команд 🟢 – < 50 команд	
Очереди выборки данных	Количество объектов в очереди службы **iw_is**	30 мин.	🔴 – > 100 объектов 🟡 – от 50 до 100 объектов 🟢 – < 50 объектов	
Очередь индексации **iw_is**	Количество объектов в очереди службы **iw_is**	30 мин.	🔴 – > 100 объектов 🟡 – от 50 до 100 объектов 🟢 – < 50 объектов	
Очередь ошибок **iw_is**	Наличие ошибок сервиса **iw_is**	5 мин.	🔴 – > 500 🟡 от 50 до 500 🟢 – < 50	
Очередь команд для табличного пространства текста	Количество команд в очереди табличного пространства текста	30 мин.	🔴 – > 50 команд 🟡 – от 20 до 50 команд 🟢 – < 20 команд	1. Включите логирование работы службы **iw_indexer** на `debug` (см. "Логирование работы Системы") 2. Просмотрите лог-файл **iw_indexer** на ошибки в `opt/iw/tm5/log/indexer.log`

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации
Очередь команд для табличного пространства метаинформации	Количество команд в очереди табличного пространства метаинформации	30 мин.	🔴 – > 50 команд 🟡 – от 20 до 50 команд 🟢 – < 20 команд	3. Обратитесь в службу технической поддержки InfoWatch

Очередь ошибок обработки событий	Количество объектов в очереди ошибок **queue/errors** (Extractors Framework, TMcap-серверы: iw_luaengined, iw_warpd, iw_cas)	30 мин.	🔴 – > 500 объектов
			🟡 – от 50 до 500 объектов
			🟢 – < 50 объектов
				Вариант 1: Очистите очередь, используя команды:
				`su - iwtm`
				`/opt/iw/tm5/bin/iw_qtool erase /opt/iw/tm5/queue/errors`
				`exit`
				Вариант 2:
				1. Откройте на редактирование файл `/etc/cron.d/iwtm_cleanup.`
				2. Уменьшите период очистки файлов в папках файловой очереди в штатном задании cron (значение по умолчанию - 30 дней).
				3. Сохраните изменения и закройте файл.

Очередь ошибок обработки событий сервисом **iw_x2x**	Количество объектов в очереди **queue/x2x-errors** (iw_x2x)	30 мин.	🔴 – > 500 объектов
			🟡 – от 50 до 500 объектов
			🟢 – 50
				Вариант 1: Очистите очередь, используя команды:
				`su - iwtm`
				`/opt/iw/tm5/bin/iw_qtool erase /`

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации

				`opt/iw/tm5/queue/x2x-errors`
				`exit`
				Вариант 2:
				1. Откройте на редактирование файл `/etc/cron.d/iwtm_cleanup.`
				2. Уменьшите период очистки файлов в папках файловой очереди в штатном задании cron (значение по умолчанию - 30 дней).
				3. Сохраните изменения и закройте файл.

Очередь ошибок обработки событий сервисом **iw_x2db**	Количество объектов в очереди **queue/x2db-errors** (iw_x2db)	30 мин.	🔴 – > 500 объектов
			🟡 – от 50 до 500 объектов
			🟢 – 50
				Вариант 1: Очистите очередь, используя команды:
				`su - iwtm`
				`/opt/iw/tm5/bin/iw_qtool erase /opt/iw/tm5/queue/x2db-errors`
				`exit`
				Вариант 2:
				1. Откройте на редактирование файл `/etc/cron.d/iwtm_cleanup.`
				2. Уменьшите период очистки файлов в папках файловой очереди в штатном

Название индикатора	Что проверяется и где проверяется	Периодичность опроса	Значения индикатора	Подробности и рекомендации
				задании cron (значение по умолчанию - 30 дней). 3. Сохраните изменения и закройте файл.

Очередь ошибок обработки действий примененных политик	Количество объектов в очереди **queue/blackboard_errors** (Linux-серверы)	30 мин.	🔴 – > 500 объектов
			🟡 – от 50 до 500 объектов
			🟢 – 50
				Вариант 1: Очистите очередь, используя команды:
				`su - iwtm`
				`/opt/iw/tm5/bin/iw_qtool erase /opt/iw/tm5/queue/blackboard-errors`
				`exit`
				Вариант 2:
				1. Откройте на редактирование файл `/etc/cron.d/iwtm_cleanup.`
				2. Уменьшите период очистки файлов в папках файловой очереди в штатном задании cron (значение по умолчанию - 30 дней).
				3. Сохраните изменения и закройте файл.