Мессенджер и ВКС

Настройка интеграции с DLP-системами (для версии 26.1)

Портал с документацией: https://biz.mail.ru/docs/on-premises/

Оглавление

Назначение документа

В документе описан механизм отправки запросов в DLP-системы, а также процесс настройки отправки данных в DLP-систему для версии Мессенджер и ВКС 26.1.

Документ предназначен для использования системными администраторами.

Дополнительная документация

Архитектура и описание системы — в документе описаны сервисы, обеспечивающие отправку данных в DLP-системы. Не является частью публичной документации, обратитесь к представителю VK Tech, чтобы ознакомиться с документом.

Инструкция по настройке интеграции с DLP-системами SearchInform и InfoWatch — в документе описан механизм отправки запросов в DLP-системы SearchInform и InfoWatch, актуальный для версии Мессенджер и ВКС 24.9 и ниже. Документ предназначен для использования системными администраторами.

Общее описание

DLP-система — специализированное программное обеспечение, предназначенное для защиты компании от утечек информации.

Мессенджер и ВКС поддерживает интеграцию со следующими поставщиками DLP-систем:

1. Solar Dozor
2. InfoWatch
3. Любая DLP-система, работающая по протоколу ICAP (базовый адаптер)

DLP-система	**Отправка данных**	**Получение результата проверки**
Solar Dozor	Используется протокол ICAP. Метаинформация передается в заголовках, а не в теле запроса. В теле запроса передаётся только проверяемый контент. Содержимое заголовков закодировано Base64 (RFC 2045)	Результат проверки возвращается в ответе на отправку данных для проверки
InfoWatch		

DLP-система	Отправка данных	Получение результата проверки
	Передача данных осуществляется через InfoWatch Traffic Monitor SDK методом pushAPI SDK. Данные, отправляемые сервисом Vahter в DLP-систему InfoWatch, представлены в документации к продукту https://kb.infowatch.com/pages/viewpage.action?pageId=165545261	Результат проверки возвращается в ответе на HTTP-запрос в проверяющую систему, запрос содержит идентификатор отправленного события
Другая DLP-система (базовый адаптер)	Для файлов и текстовых сообщений: используется протокол ICAP. Отправляемые данные соответствуют стандарту RFC 3507	Результата проверки возвращается в ответе на отправку данных для проверки

Режимы проверки отправляемых сообщений:

- Синхронный – адресаты не увидят сообщение, пока не пройдет проверка DLP-системой. В случае негативного результата проверки сообщение не будет показано адресатам.
- Асинхронный — в случае негативного результата проверки в DLP-системе автоматический отзыв сообщения не происходит.

Сообщения проверяются в следующих чатах:

- Личный чат.
- Групповой чат.
- Канал.
- Чаты звонков и чаты обсуждения задач.

Есть два типа запросов, отправляемых в DLP-систему:

- Текстовое сообщение — текстовое сообщение и текстовое сообщение с прикреплённым файлом без проверки файла.
- Файл — фото, видео, аудио, стикер, голосовое сообщение, текстовый и табличный документ, архивы. Ограничений на расширение файла нет. Размер — не более 4 Гб.

Если отправленное сообщение содержит и текст, и файл, контент проверяется на стороне DLP-системы независимо. Проверка файла не блокирует отправку сообщения.

Проверка текстовых сообщений

При отправке пользователем текстового сообщения Мессенджер и ВКС отправляет данные на проверку в DLP-систему. Вместе с текстом сообщения передается информация об отправителе сообщения (user, login, user-agent, IP-адрес), эти параметры учитывается DLP-системой при принятии решения о

блокировке сообщения. Также отправляется информация о количестве участников чата. При синхронном режиме проверки текстовое сообщение не отображается у адресатов и на других устройствах отправителя до получения ответа от DLP-системы.

При получении положительного ответа от DLP-системы текстовое сообщение становится видимым для адресатов. Отправитель сообщения видит, что сообщение получено адресатами.

При получении отрицательного ответа от DLP-системы Мессенджер и ВКС блокирует текстовое сообщение. Заблокированные сообщения отображаются в клиентском приложении отправителя в статусе часов и не синхронизируются с остальными устройствами этого пользователя. Адресаты не видят заблокированное сообщение.

Если ответ DLP-системы сопровождает статус «невозможно провести анализ», это дополнительно фиксируется в журнале на сервере Мессенджер и ВКС, передается извещение в систему мониторинга и Мессенджер и ВКС разрешает или запрещает отправку сообщения в зависимости от настроек интеграции.

При превышении времени ожидания ответа от DLP-системы Мессенджер и ВКС дополнительно отображает предупреждение в логах и передает извещение в систему мониторинга.

Если DLP-система недоступна, Мессенджер и ВКС отображает сбой в логах и передает извещение во внешнюю систему мониторинга по стандартному протоколу. Далее, в зависимости от настроек, Мессенджер и ВКС отправляет сообщения без проверки в DLP-системе либо останавливает отправку сообщений до восстановления доступа к DLP-системе в тех чатах и для тех пользователей, где согласно настройкам проверка необходима.

Мессенджер и ВКС не хранит заблокированные текстовые сообщения. При сигнале от DLP-системы о блокировке, мессенджер не отправляет текстовое сообщение, оно остается (хранится) на устройстве пользователя.

Проверка файлов

При отправке пользователем файла Мессенджер и ВКС отправляет данные на проверку в DLP-систему. При этом ссылка на файл всегда отправляется адресатам, но файл недоступен для просмотра или скачивания, пока не получен ответ от DLP-системы. Для файлов типа картинок также отключена возможность превью, до получения положительного ответа от DLP-системы адресаты видят только «заглушку» файла.

При получении положительного ответа от DLP-системы файл становится доступным для просмотра и скачивания в зависимости от уровня доступа файла:

- External – файл доступен вне корпоративной системы передачи данных (интернет/внешняя среда).
- Internal – файл доступен только в корпоративной системе передачи данных.

Примечание

Правила определения уровня доступа настраиваются администратором организации. По умолчанию в инсталляции не предоставляется никакой конфигурации правил, связывающих эти уровни с конкретными подсетями/клиентами. Настройка описана ниже.

Уровень доступа на скачивание файла определяется по IP-адресу получателя и устройству, с которого он зашёл в Мессенджер и ВКС.

При получении отрицательного ответа от DLP-системы файл недоступен для просмотра и скачивания, но адресатам доступны метаданные файла (название, тип расширения, размер и т. д.).

Если DLP-система недоступна

При ожидании ответа от DLP-системы или если DLP-система недоступна, Мессенджер и ВКС отражает сбой в логах и передает извещение во внешнюю систему мониторинга по стандартному протоколу. Далее, в зависимости от настроек параметра stretigy_on_fail, Мессенджер и ВКС отправляет файл/текстовое сообщение без проверки в DLP-системе либо останавливает отправку файлов/текстовых сообщений до восстановления доступа к DLP-системе.

Если ответ DLP-системы сопровождает статус «невозможно провести анализ», это дополнительно фиксируется в журнале на сервере Мессенджер и ВКС, и извещение передается в систему мониторинга.

При ожидании ответа от DLP-системы или если DLP-система недоступна, файл недоступен для просмотра и скачивания, но адресатам доступны метаданные файла. Если сообщение с заблокированным файлом не было отправлено, оно хранится на устройстве отправителя. Если сообщение было отправлено, оно хранится на устройствах отправителя и получателя без возможности открыть или скачать файл.

Настройка отправки текстовых сообщений в DLP-систему

> **Внимание**
>
> Все команды в консоли выполняются под пользователем root.

Шаг 1. Активируйте сервис Vahter

Сервис находится в инсталляции в неактивном состоянии. Чтобы активировать сервис:

1. Перейдите в конфигурационный файл **/usr/local/etc/k8s/helmwave/projects.yml** и удалите из секции **disabled** сервис Vahter и строку **strimzi-kafka-operator**.
2. Запустите деплой сервиса.

   Для инсталляции на одну виртуальную машину выполните команду:

   ```
   HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
   ```

   Для кластерной инсталляции:

   ```
   HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
   ```

3. Проверьте, что сервис запустился:

   ```
   kubectl -n vkteams get po | grep vahter
   ```

   В выводе консоли pod должен находиться в статусе Running.

Шаг 2. Настройте конфигурацию адаптера к DLP-системе

В конфигурационном файле **/usr/local/etc/k8s/helmwave/store/dlp.yml** в секции **adapters** содержатся общие и исключительные параметры для проверки текста и файлов. Секция **adapter_manager** управляет всеми адаптерами, описанными в секции **adapters**, и определяет, какой адаптер будет использован. Базовый адаптер может быть использован для интеграции с любой DLP-системой, работающей по протоколу ICAP.

1. Укажите настройки подключения к DLP-системе:

Базовый адаптер

```
adapters:
    - provider: default_adapter
      name: my_icap_adapter_1
      enabled: true
      url: <your_default_adapter_host.example>
      icap_client_timeout: 10s
      topic_url: http://VkTeams
      check_text_warn_timeout: 1s
      check_text_disconnect_timeout: 3s
      check_file_warn_timeout: 1m #используется для отправки файлов в DLP-систему
      check_file_disconnect_timeout: 10m #используется для отправки файлов в DLP-систему
      strategy_on_fail: block
      debug_mode: false
      access_levels_recheck: false
      default_access_level: Internal #используется для отправки файлов в DLP-систему
      access_levels: #используется для отправки файлов в DLP-систему
        - Internal
        - External
adapters_manager:
    icap_debug: false
    default_adapter_name: my_icap_adapter_1
```

где:

- name — имя адаптера.
- enabled — если значение true – включена отправка данных в DLP-систему, если false – выключена.
- url — адрес ICAP-сервера.
- icap_client_timeout — таймаут для соединения с ICAP-сервером.
- topic_url — топик сообщения.
- check_text_warn_timeout — таймаут, при достижении которого пишется лог о превышении времени проверки текста и увеличивается соответствующая метрика, которая подсчитывает количество превышений этого порога. Необходим для оценки времени реакции DLP-системы.
- check_text_disconnect_timeout — таймаут, при достижении которого обрывается запрос проверки текста, создается запись в логах и увеличивается соответствующая метрика.

> **Внимание**
>
> Не устанавливайте таймаут более семи секунд, это может привести к проблемам отправки сообщений в Сумраппе

- check_file_warn_timeout — используется для отправки файлов в DLP-систему, не изменяйте параметр.
- check_file_disconnect_timeout — используется для отправки файлов в DLP-систему, не изменяйте параметр.

- strategy_on_fail — используется для формирования результата проверки текста, когда DLP-система долго не отвечает или недоступна. Параметр может принимать значения:
    - ok — при превышении порога check_text_disconnect_timeout или недоступности DLP-системы в сервис Vahter будет возвращаться успешный ответ, как если бы был получен положительный результат проверки сообщения.
    - block — в случае возникновения ошибки в работе DLP-системы и невозможности проверить сообщение Мессенджер и ВКС будет считать сообщение заблокированным. Пользователю отобразится статус «Сообщение заблокировано».
    - fail — ошибка в работе DLP-системы будет считаться внутренней ошибкой отправки сообщения. Клиентское приложение попытается повторить отправку (что приведет к повторным запросам в DLP-систему) и будет отображать статус «Ожидание отправки» или «Ошибка».
- debug_mode — активация режима отладки, который рекомендуется включать при первичной настройке системы, чтобы в случае проблемы в интеграции с DLP-системой работа мессенджера не была затронута. Режим отладки отличается от реального лишь тем, что на проверку текста будет сразу отдан успешный ответ, как если бы DLP-система провела проверку и не нашла бы ничего подозрительного в тексте. Значение true включает режим отладки, значение false выключает.
- access_levels_recheck — используется для отправки файлов в DLP-систему, не изменяйте параметр.
- default_access_level — используется для отправки файлов в DLP-систему, не изменяйте параметр.
- access_levels — используется для отправки файлов в DLP-систему, не изменяйте параметр.

В секции **adapters_manager** укажите настройки управления адаптерами:

- icap_debug — если true, то включено подробное логирование при общении по протоколу ICAP.
- default_adapter_name — укажите адаптер, используемый по умолчанию.

Solar dozor

```
adapters:
    - provider: solar_dozor
      name: solar_dozor
      enabled: true
      url: <your_solar_dozor_host.example>
      icap_client_timeout: 10s
      topic_url: http://VkTeams
      check_text_warn_timeout: 1s
      check_text_disconnect_timeout: 3s
      check_file_warn_timeout: 1m #используется для отправки файлов в DLP-систему
      check_file_disconnect_timeout: 10m #используется для отправки файлов в DLP-систему
      strategy_on_fail: fail
      debug_mode: false
      access_levels_recheck: false
      default_access_level: Internal #используется для отправки файлов в DLP-систему
      access_levels: #используется для отправки файлов в DLP-систему
        - Internal
        - External
adapters_manager:
    icap_debug: false
    default_adapter_name: solar_dozor
```

где:

- name — имя адаптера.
- enabled — если значение true – включена отправка данных в DLP-систему, если false – выключена.
- url — адрес ICAP-сервера.
- icap_client_timeout — таймаут для соединения с ICAP-сервером.
- topic_url — топик сообщения.
- check_text_warn_timeout — таймаут, при достижении которого пишется лог о превышении времени проверки текста и увеличивается соответствующая метрика, которая подсчитывает количество превышений этого порога. Необходим для оценки времени реакции DLP-системы.
- check_text_disconnect_timeout — таймаут, при достижении которого обрывается запрос проверки текста, создается запись в логах и увеличивается соответствующая метрика.

> **Внимание**
>
> Не устанавливайте таймаут более семи секунд, это может привести к проблемам отправки сообщений в Супераппе

- check_file_warn_timeout — используется для отправки файлов в DLP-систему, не изменяйте параметр.
- check_file_disconnect_timeout — используется для отправки файлов в DLP-систему, не изменяйте параметр.

- strategy_on_fail — используется для формирования результата проверки текста, когда DLP-система долго не отвечает или недоступна. Параметр может принимать значения:
  - ok — при превышении порога check_text_disconnect_timeout или недоступности DLP-системы в сервис Vahter будет возвращаться успешный ответ, как если бы был получен положительный результат проверки сообщения.
  - block — в случае возникновения ошибки в работе DLP-системы и невозможности проверить сообщение Мессенджер и ВКС будет считать сообщение заблокированным. Пользователю отобразится статус «Сообщение заблокировано».
  - fail — ошибка в работе DLP-системы будет считаться внутренней ошибкой отправки сообщения. Клиентское приложение попытается повторить отправку (что приведет к повторным запросам в DLP-систему) и будет отображать статус «Ожидание отправки» или «Ошибка».
- debug_mode — активация режима отладки, который рекомендуется включать при первичной настройке системы, чтобы в случае проблемы в интеграции с DLP-системой работа мессенджера не была затронута. Режим отладки отличается от реального лишь тем, что на проверку текста будет сразу отдан успешный ответ, как если бы DLP-система провела проверку и не нашла бы ничего подозрительного в тексте. Значение true включает режим отладки, значение false выключает.
- access_levels_recheck — используется для отправки файлов в DLP-систему, не изменяйте параметр.
- default_access_level — используется для отправки файлов в DLP-систему, не изменяйте параметр.
- access_levels — используется для отправки файлов в DLP-систему, не изменяйте параметр.

В секции **adapters_manager** укажите настройки управления адаптерами:

- icap_debug — если true, то включено подробное логирование при общении по протоколу ICAP.
- default_adapter_name — укажите адаптер, используемый по умолчанию.

InfoWatch

```
adapters:
    - provider: info_watch
      name: info_watch
      enabled: true
      address: <your_infowatch_host.example>
      token: <your_token>
      company: VKteams # не меняйте параметр при использовании предоставленного
manifest.json, значения должны совпадать с указанными внутри manifest.json
      imservice: im_VKteams # не меняйте параметр при использовании предоставленного
manifest.json, значения должны совпадать с указанными внутри manifest.json
      capture_server_fqdn: vahter
      push_api_version: 1.9
      push_api_address: <your_infowatch_host.example>:1234/verdict
      push_api_retry_count: 5
      check_text_warn_timeout: 1s
      check_text_disconnect_timeout: 3s
      check_file_warn_timeout: 1m #используется для отправки файлов в DLP-систему
      check_file_disconnect_timeout: 10m #используется для отправки файлов в DLP-систему
      strategy_on_fail: block
      debug_mode: false
      access_levels_recheck: false
      default_access_level: Internal #используется для отправки файлов в DLP-систему
      access_levels: #используется для отправки файлов в DLP-систему
        - Internal
        - External
      check_file_chunk_bytes: 100480 #используется для отправки файлов в DLP-систему
adapters_manager:
    default_adapter_name: solar_dozor
```

где:

- name — имя адаптера.
- enabled — если значение true – включена отправка данных в DLP-систему, если false – выключена.
- address — адрес проверяющего сервера.
- token — токен доступа.
- company — наименования компании отправителя. Не меняйте параметр при использовании предоставленного manifest.json, значения должны совпадать с указанными внутри manifest.json.
- imservice — наименование сервиса отправителя. Не меняйте параметр при использовании предоставленного manifest.json, значения должны совпадать с указанными внутри manifest.json.
- capture_server_fqdn — наименование сервера отправителя.
- push_api_version — версия push_api.
- push_api_address — адрес HTTP-сервера, возвращающего результат проверки.
- push_api_retry_count — количество попыток получения результата.
- check_text_warn_timeout — таймаут, при достижении которого пишется лог о превышении времени проверки текста и увеличивается соответствующая метрика, которая подсчитывает количество превышений этого порога. Необходим для оценки времени реакции DLP-системы.

- check_text_disconnect_timeout — таймаут, при достижении которого обрывается запрос проверки текста, создается запись в логах и увеличивается соответствующая метрика.

> **Внимание**
>
> Не устанавливайте таймаут более семи секунд, это может привести к проблемам отправки сообщений в Супераппе

- check_file_warn_timeout — используется для отправки файлов в DLP-систему, не изменяйте параметр.
- check_file_disconnect_timeout — используется для отправки файлов в DLP-систему, не изменяйте параметр.
- strategy_on_fail — используется для формирования результата проверки текста, когда DLP-система долго не отвечает или недоступна. Параметр может принимать значения:
  - ok — при превышении порога check_text_disconnect_timeout или недоступности DLP-системы в сервис Vahter будет возвращаться успешный ответ, как если бы был получен положительный результат проверки сообщения.
  - block — в случае возникновения ошибки в работе DLP-системы и невозможности проверить сообщение Мессенджер и ВКС будет считать сообщение заблокированным. Пользователю отобразится статус «Сообщение заблокировано».
  - fail — ошибка в работе DLP-системы будет считаться внутренней ошибкой отправки сообщения. Клиентское приложение попытается повторить отправку (что приведет к повторным запросам в DLP-систему) и будет отображать статус «Ожидание отправки» или «Ошибка».
- debug_mode — активация режима отладки, который рекомендуется включать при первичной настройке системы, чтобы в случае проблемы в интеграции с DLP-системой работа мессенджера не была затронута. Режим отладки отличается от реального лишь тем, что на проверку текста будет сразу отдан успешный ответ, как если бы DLP-система провела проверку и не нашла бы ничего подозрительного в тексте. Значение true включает режим отладки, значение false выключает.
- access_levels_recheck— используется для отправки файлов в DLP-систему, не изменяйте параметр.
- default_access_level— используется для отправки файлов в DLP-систему, не изменяйте параметр.
- access_levels — используется для отправки файлов в DLP-систему, не изменяйте параметр.
- check_file_chunk_bytes — используется для отправки файлов в DLP-систему, не изменяйте параметр.

В секции **adapters_manager** укажите настройки управления адаптерами:

- default_adapter_name — укажите адаптер, используемый по умолчанию.

2. Примените изменения.

Для инсталляции на одну виртуальную машину выполните команду:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
```

Для кластерной инсталляции:

```
HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
```

Шаг 3. Создайте плагин для подключения провайдера

Данный шаг актуален только для DLP-системы InfoWatch. Если вы используете другую DLP-систему, перейдите к следующему шагу.

Подключение провайдера данных осуществляется через подключение плагина. В плагине содержится информация о поставщике данных, типы обрабатываемых событий и пользовательские заголовки. Подробнее про регистрацию сторонних компонентов можно посмотреть в документации InfoWatch — https://kb.infowatch.com/pages/viewpage.action?pageId=217787144

Для создания плагина создайте файл **manifest.json**. Пример manifest.json-файла плагина:

```
{
    "PLUGIN_ID": "346227C2657C4701B86892CAE732805D",
    "DISPLAY_NAME": "Плагин для события мессенджера",
    "DESCRIPTION": {
        "eng": "IM events reception",
        "rus": "Прием событий менеджера"
    },
    "VERSION": "0.0.1",
    "VENDOR": "VKteams",
    "LICENSE": [
        {
            "PATH": "licenses/tm_license.license"
        }
    ],
    "PATTERN_SEARCH_LICENSE": {
        "operator": "and",
        "conditions": [
            {
                "common_name": "VKteams"
            },
            {
                "object_type": "im_VKteams"
            },
            {
                "protocol": "NONE"
            }
        ]
    },
    "ADDS_SERVICES": {
        "SERVICE_TYPE": [
            {
                "SERVICE_MNEMO": "im_VKteams",
                "DATA_CLASS": [
                    "kChat",
                    "kFileExchange"
                ],
                "ICON": "icon/acme_messenger.png",
```

```
                "LOCALE": {
                    "rus": "Мессенджер VKteams",
                    "eng": "VKteams messenger"
                },
                "CONTACT_TYPE": [
                    {
                        "MNEMO": "im_VKteams",
                        "SCOPE": [
                            "person"
                        ],
                        "ICON": "icon/acme_messenger.png",
                        "LOCALE": {
                            "rus": "Аккаунт VKteams",
                            "eng": "VKteams account"
                        }
                    }
                ]
            }
        ]
    },
    "OBJECT_HEADER": [
        {
            "NAME": "VKteams_file_hash_header",
            "NOTE": {
                "rus": "Хеш файла",
                "eng": "File hash"
            },
            "DATA_CLASS": [
                "kChat",
                "kFileExchange"
            ],
            "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": "VKteams_text_chat_name_header",
            "NOTE": {
                "rus": "Название чата",
                "eng": "Chat name"
            },
            "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",
            "TYPE": "string",
            "FORMAT": "string",
            "IS_MULTIPLE_VALUE": "1"
        },
        {
            "NAME": "VKteams_text_chat_id_header",
            "NOTE": {
```

```
                "rus": "ID чата",
                "eng": "Chat ID"
            },
            "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",
            "TYPE": "string",
            "FORMAT": "string",
            "IS_MULTIPLE_VALUE": "1"
        },
        {
            "NAME": "VKteams_text_chat_participants_header",
            "NOTE": {
                "rus": "Количество участников чата",
                "eng": "Number of chat participants"
            },
            "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",
            "TYPE": "number",
            "FORMAT": "integer",
            "IS_MULTIPLE_VALUE": "1"
        },
        {
            "NAME": "VKteams_message_type_header",
            "NOTE": {
                "rus": "Тип отправленного сообщения",
                "eng": "Message type"
            },
            "DATA_CLASS": [
                "kChat",
                "kFileExchange"
            ],
            "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": "VKteams_sender_ip_header",
            "NOTE": {
                "rus": "IP отправителя",
                "eng": "Sender IP"
            },
            "DATA_CLASS": [
                "kChat",
```

```
                "kFileExchange"
            ],
            "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": "VKteams_sender_ua_header",
            "NOTE": {
                "rus": "UA отправителя",
                "eng": "Sender UA"
            },
            "DATA_CLASS": [
                "kChat",
                "kFileExchange"
            ],
            "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": "VKteams_access_level_header",
            "NOTE": {
                "rus": "Уровень доступа к файлу",
                "eng": "File access_level"
            },
            "DATA_CLASS": [
                "kFileExchange"
            ],
            "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"
        }
    ]
}
```


Шаг 4. Активируйте envoy-плагин Tourniquet

1. В конфигурационном файле **/usr/local/etc/k8s/helmwave/store/dlp.yml** установите в поле **tourniquetEnabled** значение **true**.

2. Примените изменения.

   Для инсталляции на одну виртуальную машину выполните команды:

   ```
   HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress
   ```

   Для кластерной инсталляции:

   ```
   HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress
   ```

3. Проверьте, что плагин запустился:

   ```
   kubectl -n istio-ingress get po | grep istio-ingress
   ```

   В выводе консоли pod должен находиться в статусе Running.

4. Проверьте логи сервиса на возможные проблемы:

   ```
   sudo kubectl logs -n istio-ingress $(sudo kubectl get pod -n istio-ingress | grep istio-
   ingress | awk '{print $1}')
   ```

   Искать необходимо по названию модуля Tourniquet. Пример проблем в логах:

   ```
   2024-11-21T13:52:28.946719Z error   envoy golang external/envoy/contrib/golang/common/log/
   cgo.cc:24 failed to parse golang plugin config: dlp_resp_blocked_code: expect float64 while
   got string    thread=14
   2024-11-21T13:52:28.946961Z warning envoy config external/envoy/source/extensions/
   config_subscription/grpc/delta_subscription_state.cc:269  delta config for
   type.googleapis.com/envoy.config.listener.v3.Listener rejected: Error adding/updating
   listener(s) 0.0.0.0_80: golang filter failed to parse plugin config: tourniquet /var/lib/
   im/modules/tourniquet.so
   ```

   Подобная запись в логах говорит о проблемах в конфигурации — необходимо перепроверить конфигурацию или вернуть конфиг к значению по умолчанию.

Шаг 5. Отключите Zstandard-сжатие

Для правильной работы модуля Tourniquet необходимо отключить Zstd-сжатие. Для этого:

1. В конфигурационном файле **/usr/local/nginx-im/html/myteam/myteam-config.json** установите для полей **zstd-requests-enabled** и **zstd-responses-enabled** значение false:

   ```
   {
      // ...
        "zstd-requests-enabled": false,
        "zstd-responses-enabled": false,
      // ...
   }
   ```

2. Для инсталляции на одну виртуальную машину выполните команду:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t godmod
```

Для кластерной инсталляции:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true HELMWAVE_ENV_NAME=cluster hwup -t godmod
```

3. Перезапустите под в технологическое окно (может приводить к сбою в новых подключениях):

```
kubectl delete pods -n vkteams -l app=myteam-admin
```

Шаг 6. Настройте маршрутизацию по доменам

Данный шаг является опциональным. Вы можете настроить выбор адаптера в зависимости от домена отправителя сообщения.

Перейдите в конфигурационный файл **/usr/local/etc/k8s/helmwave/store/dlp.yml** и заполните секцию **adapter_chooser**:

```
adapter_chooser:
  enabled: false
  adapters_rules:
  - target: "solar_dozor" # адаптер, в который будут направляться сообщения от domain_list
    domain_list:
    - "domain_1"
    - "domain_2"
  - target: "info_watch"
    domain_list:
    - "domain_3"
    - "domain_4"
```

1. Установите для параметра **enabled** значение true. Если false (конфигурация по умолчанию) — для всех доменов будет использоваться адаптер из параметра **default_adapter_name** конфигурационного файла **/usr/local/etc/k8s/helmwave/store/dlp.yml**.
2. В секции **adapters_rules** указываются правила сопоставления домена и адаптера. Укажите для параметров **target** нужный адаптер и список доменов, для которых этот адаптер будет использоваться.
3. Примените изменения.

   Для инсталляции на одну виртуальную машину выполните команды:

   ```
   HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress
   ```

   Для кластерной инсталляции:

   ```
   HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress
   ```

Шаг 7. Настройте исключения из проверок

Данный шаг является опциональным. Вы можете настроить список пользователей, исключенных из проверок. Сообщения, отправляемые такими пользователями, не будут попадать в DLP-систему и не будут блокироваться.

Если вы настроили маршрутизацию по доменам:

Перейдите в конфигурационный файл **/usr/local/etc/k8s/helmwave/store/dlp.yml** и укажите email-адреса пользователей в секции **adapters_rules.excluded_users**:

```
adapter_chooser:
  enabled: false
  adapters_rules:
  - target: "solar_dozor" # адаптер, в который будут направляться сообщения от domain_list
    domain_list:
    - "domain_1"
    - "domain_2"
    excluded_users: # исключаем пользователей из проверок
    - "i.ivanov@company_domain_1"
    - "p.petrov@company_domain_2"
  - target: "info_watch"
    domain_list:
    - "domain_3"
    - "domain_4"
```

Примените изменения.

Для инсталляции на одну виртуальную машину выполните команды:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress
```

Для кластерной инсталляции:

```
HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress
```

Если вы не настраивали маршрутизацию по доменам:

Перейдите в конфигурационный файл **/usr/local/etc/k8s/helmwave/store/dlp.yml** и укажите email-адреса пользователей в секции **adapters_manager.excluded_users**:

```
adapters_manager:
  icap_debug: false
  default_adapter_name: search_inform
  excluded_users: # исключаем пользователей из проверок
    - "i.ivanov@company_domain_1"
    - "p.petrov@company_domain_2"
```

Примените изменения.

Для инсталляции на одну виртуальную машину выполните команду:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
```

Для кластерной инсталляции:

```
HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
```

Настройка отправки файлов в DLP-систему

> **Внимание**
>
> Все команды в консоли выполняются под пользователем root.

Шаг 1. Активируйте Kafka и сервис Vahter

Сервис Vahter находится в инсталляции в неактивном состоянии. Чтобы активировать сервисы:

1. В конфигурационном файле **/usr/local/etc/k8s/helmwave/projects.yml** удалите из секции **envs: all: projects: disabled:**
   - строку **strimzi-kafka-operator**.
   - сервис Vahter.
2. Запустите деплой сервиса Vahter.

   Для инсталляции на одну виртуальную машину выполните команду:

   ```
   HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
   ```

   Для кластерной инсталляции:

   ```
   HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
   ```
3. Проверьте, что сервис запустился:

   ```
   kubectl -n vkteams get po | grep vahter
   ```

   В выводе консоли pod должен находиться в статусе Running.

Шаг 2. Активируйте сервис Multifora

1. В конфигурационном файле **/usr/local/etc/k8s/helmwave/projects.yml** удалите из секции **disabled** сервис Multifora.
2. Запустите деплой сервиса.

   Для инсталляции на одну виртуальную машину выполните команду:

   ```
   HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t multifora
   ```

Для кластерной инсталляции:

```
HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t multifora
```

3. Проверьте, что сервис запустился:

```
kubectl -n vkteams get po | grep multifora
```

В выводе консоли pod должен находиться в статусе Running.

Шаг 3. Настройте конфигурацию адаптера к DLP-системе

В конфигурационном файле **/usr/local/etc/k8s/helmwave/store/dlp.yml** в секции **adapters** содержатся общие параметры для проверки текста/файлов и исключительные для текста и файлов. Секция **adapter_manager** управляет всеми адаптерами, описанными в секции **adapters**, и определяет, какой адаптер будет использован.

1. Укажите настройки подключения к DLP-системе:

Базовый адаптер

```
adapters:
    - provider: default_adapter
      name: default_adapter
      enabled: true
      url: <your_solar_dozor_host.example>
      icap_client_timeout: 10s
      topic_url: http://VkTeams
      check_text_warn_timeout: 1s #используется для отправки текстовых сообщений в DLP-
систему
      check_text_disconnect_timeout: 3s #используется для отправки текстовых сообщений в
DLP-систему
      check_file_warn_timeout: 1m
      check_file_disconnect_timeout: 10m
      strategy_on_fail: block
      debug_mode: false
      access_levels_recheck: false
      default_access_level: Internal
      access_levels:
        - Internal
        - External
adapters_manager:
    icap_debug: false
    default_adapter_name: default_adapter
```

где:

- name — имя адаптера.

- enabled — если значение true – включена отправка данных в DLP-систему, если false – выключена.
- url — адрес для интеграции с DLP-системой.
- icap_client_timeout — таймаут на соединение с DLP-системой.
- topic_url — тема сообщения в URL-формате.
- check_text_warn_timeout — используется для отправки текстовых сообщений в DLP-систему, не изменяйте параметр.
- check_text_disconnect_timeout — используется для отправки текстовых сообщений в DLP-систему, не изменяйте параметр.

> **Внимание**
>
> Не устанавливайте таймаут более семи секунд, это может привести к проблемам отправки сообщений в Супераппе

- check_file_warn_timeout — таймаут, при достижении которого пишется лог о превышении времени проверки файлов и увеличивается соответствующая метрика, которая подсчитывает количество превышений этого порога. Необходим для оценки времени реакции DLP-системы.
- check_file_disconnect_timeout — таймаут, при достижении которого обрывается запрос проверки файлов, создается запись в логах и увеличивается соответствующая метрика. Параметр check_file_disconnect_timeout должен быть больше параметра max_timeout
- strategy_on_fail — используется для формирования результата проверки текста, когда DLP-система долго не отвечает или недоступна. Параметр может принимать значения:
  - ok — при превышении порога check_text_disconnect_timeout или недоступности DLP-системы в сервис Vahter будет возвращаться успешный ответ, как если бы был получен положительный результат проверки сообщения.
  - block — в случае возникновения ошибки в работе DLP-системы и невозможности проверить сообщение Мессенджер и ВКС будет считать сообщение заблокированным. Пользователю отобразится статус «Сообщение заблокировано».
  - fail — ошибка в работе DLP-системы будет считаться внутренней ошибкой отправки сообщения. Клиентское приложение попытается повторить отправку (что приведет к повторным запросам в DLP-систему) и будет отображать статус «Ожидание отправки» или «Ошибка».
- debug_mode — активация режима отладки, который рекомендуется включать при первичной настройке системы, чтобы в случае проблемы в интеграции с DLP-системой работа мессенджера не была затронута. Режим отладки отличается от реального лишь тем, что на проверку файлов будет отдан успешный ответ, как если бы DLP-система провела проверку и не нашла бы ничего подозрительного в файле. Значение true включает режим отладки, значение false выключает.
- access_levels_recheck — если значение true, включается логика «перезапросов» доступа к файлу из разных контуров. Пример: пользователь ранее запросил доступ к файлу из корпоративной сети, и DLP-система вернула позитивный результат проверки. Далее пользователь запросил

доступ к тому же файлу из внешней сети. Если для параметра установлено значение true, в DLP-систему будет направлен «перезапрос» доступа. Перезапросы выполняются последовательно - от самого доверенного до менее доверенного. Результатом будет минимальный уровень доступа, которым должен обладать пользователь, чтобы просмотреть/скачать файл.

- default_access_level— уровень доступа к файлам по умолчанию, используется при выключеннном параметре access_levels_recheck.
- access_levels — уровни доступа к файлу:
  - External — файл доступен вне корпоративной системы передачи данных (интернет/внешняя среда).
  - Internal — файл доступен только в корпоративной системе передачи данных. Обязательное требование к порядку уровней – от самого доверенного к менее доверенному уровню доступа, иначе будут получены некорректные результаты проверки и «перепроверки».

В секции **adapters_manager** укажите настройки управления адаптерами:

- icap_debug — если true, то включено подробное логирование при общении по протоколу ICAP.
- default_adapter_name — укажите адаптер, используемый по умолчанию.

Solar dozor

```
adapters:
    - provider: solar_dozor
      name: solar_dozor
      enabled: true
      url: <your_solar_dozor_host.example>
      icap_client_timeout: 10s
      topic_url: http://VkTeams
      check_text_warn_timeout: 1s #используется для отправки текстовых сообщений в DLP-
систему
      check_text_disconnect_timeout: 3s #используется для отправки текстовых сообщений в
DLP-систему
      check_file_warn_timeout: 1m
      check_file_disconnect_timeout: 10m
      strategy_on_fail: fail
      debug_mode: false
      access_levels_recheck: false
      default_access_level: Internal
      access_levels:
        - Internal
        - External
adapters_manager:
    icap_debug: false
    default_adapter_name: solar_dozor
```

где:

- name — имя адаптера.
- enabled — если значение true – включена отправка данных в DLP-систему, если false – выключена.
- url — адрес для интеграции с DLP-системой.
- icap_client_timeout — таймаут на соединение с DLP-системой.

- topic_url — тема сообщения в URL-формате.
- check_text_warn_timeout — используется для отправки текстовых сообщений в DLP-систему, не изменяйте параметр.
- check_text_disconnect_timeout — используется для отправки текстовых сообщений в DLP-систему, не изменяйте параметр.

> **Внимание**
>
> Не устанавливайте таймаут более семи секунд, это может привести к проблемам отправки сообщений в Супераппе

- check_file_warn_timeout — таймаут, при достижении которого пишется лог о превышении времени проверки файлов и увеличивается соответствующая метрика, которая подсчитывает количество превышений этого порога. Необходим для оценки времени реакции DLP-системы.
- check_file_disconnect_timeout — таймаут, при достижении которого обрывается запрос проверки файлов, создается запись в логах и увеличивается соответствующая метрика. Параметр check_file_disconnect_timeout должен быть больше параметра max_timeout
- strategy_on_fail — используется для формирования результата проверки текста, когда DLP-система долго не отвечает или недоступна. Параметр может принимать значения:
    - ok — при превышении порога check_text_disconnect_timeout или недоступности DLP-системы в сервис Vahter будет возвращаться успешный ответ, как если бы был получен положительный результат проверки сообщения.
    - block — в случае возникновения ошибки в работе DLP-системы и невозможности проверить сообщение Мессенджер и ВКС будет считать сообщение заблокированным. Пользователю отобразится статус «Сообщение заблокировано».
    - fail — ошибка в работе DLP-системы будет считаться внутренней ошибкой отправки сообщения. Клиентское приложение попытается повторить отправку (что приведет к повторным запросам в DLP-систему) и будет отображать статус «Ожидание отправки» или «Ошибка».
- debug_mode — активация режима отладки, который рекомендуется включать при первичной настройке системы, чтобы в случае проблемы в интеграции с DLP-системой работа мессенджера не была затронута. Режим отладки отличается от реального лишь тем, что на проверку текста будет сразу отдан успешный ответ, как если бы DLP-система провела проверку и не нашла бы ничего подозрительного в тексте. Значение true включает режим отладки, значение false выключает.
- access_levels_recheck — если значение true, включается логика «перезапросов» доступа к файлу из разных контуров. Пример: пользователь ранее запросил доступ к файлу из корпоративной сети, и DLP-система вернула позитивный результат проверки. Далее пользователь запросил доступ к тому же файлу из внешней сети. Если для параметра установлено значение true, в DLP-систему будет направлен «перезапрос» доступа. Перезапросы выполняются последовательно - от самого доверенного до менее доверенного. Результатом будет минимальный уровень доступа, которым должен обладать пользователь, чтобы просмотреть/скачать файл.

- default_access_level— уровень доступа к файлам по умолчанию, используется при выключеннном параметре access_levels_recheck.
- access_levels — уровни доступа к файлу:
  - External — файл доступен вне корпоративной системы передачи данных (интернет/внешняя среда).
  - Internal — файл доступен только в корпоративной системе передачи данных. Обязательное требование к порядку уровней – от самого доверенного к менее доверенному уровню доступа, иначе будут получены некорректные результаты проверки и «перепроверки».

В секции **adapters_manager** укажите настройки управления адаптерами:

- icap_debug — если true, то включено подробное логирование при общении по протоколу ICAP.
- default_adapter_name — укажите адаптер, используемый по умолчанию.

InfoWatch

```
adapters:
    - provider: info_watch
      name: info_watch
      enabled: true
      address: <your_infowatch_host.example>
      token: <your_token>
      company: VKteams # не меняйте параметр при использовании предоставленного
manifest.json, значения должны совпадать с указанными внутри manifest.json
      imservice: im_VKteams # не меняйте параметр при использовании предоставленного
manifest.json, значения должны совпадать с указанными внутри manifest.json
      capture_server_fqdn: vahter
      push_api_version: 1.9
      push_api_address: <your_infowatch_host.example>:1234/verdict
      push_api_retry_count: 5
      check_text_warn_timeout: 1s #используется для отправки текстовых сообщений в DLP-
систему
      check_text_disconnect_timeout: 3s #используется для отправки текстовых сообщений в
DLP-систему
      check_file_warn_timeout: 1m
      check_file_disconnect_timeout: 10m
      strategy_on_fail: block
      debug_mode: false
      access_levels_recheck: false
      default_access_level: Internal
      access_levels:
        - Internal
        - External
      check_file_chunk_bytes: 100480
adapters_manager:
    default_adapter_name: info_watch
```

где:

- name — имя адаптера.
- enabled — если значение true – включена отправка данных в DLP-систему, если false – выключена.
- address — адрес проверяющего сервера.

- token — токен доступа.
- company — наименования компании отправителя. Не меняйте параметр при использовании предоставленного manifest.json, значения должны совпадать с указанными внутри manifest.json.
- imservice — наименование сервиса отправителя. Не меняйте параметр при использовании предоставленного manifest.json, значения должны совпадать с указанными внутри manifest.json.
- capture_server_fqdn — наименование сервера отправителя.
- push_api_version — версия push_api.
- push_api_address — адрес HTTP-сервера, возвращающего результат проверки.
- push_api_retry_count — количество попыток получения результата.
- check_text_warn_timeout — используется для текстовых сообщений файлов в DLP-систему, не изменяйте параметр.
- check_text_disconnect_timeout — используется для отправки тектовых сообщений в DLP-систему, не изменяйте параметр.

> **Внимание**
>
> Не устанавливайте таймаут более семи секунд, это может привести к проблемам отправки сообщений в Супераппе

- check_file_warn_timeout — таймаут, при достижении которого пишется лог о превышении времени проверки файлов и увеличивается соответствующая метрика, которая подсчитывает количество превышений этого порога. Необходим для оценки времени реакции DLP-системы.
- check_file_disconnect_timeout — таймаут, при достижении которого обрывается запрос проверки файлов, создается запись в логах и увеличивается соответствующая метрика. Параметр check_file_disconnect_timeout должен быть больше параметра max_timeout.
- strategy_on_fail — используется для формирования результата проверки текста, когда DLP-система долго не отвечает или недоступна. Параметр может принимать значения:
    - ok — при превышении порога check_text_disconnect_timeout или недоступности DLP-системы в сервис Vahter будет возвращаться успешный ответ, как если бы был получен положительный результат проверки сообщения.
    - block — в случае возникновения ошибки в работе DLP-системы и невозможности проверить сообщение Мессенджер и ВКС будет считать сообщение заблокированным. Пользователю отобразится статус «Сообщение заблокировано».
    - fail — ошибка в работе DLP-системы будет считаться внутренней ошибкой отправки сообщения. Клиентское приложение попытается повторить отправку (что приведет к повторным запросам в DLP-систему) и будет отображать статус «Ожидание отправки» или «Ошибка».
- debug_mode — активация режима отладки, который рекомендуется включать при первичной настройке системы, чтобы в случае проблемы в интеграции с DLP-системой работа мессенджера не была затронута. Режим отладки отличается от реального лишь тем, что на проверку текста будет сразу отдан успешный ответ, как если бы DLP-система провела проверку

и не нашла бы ничего подозрительного в тексте. Значение true включает режим отладки, значение false выключает.

- access_levels_recheck — если значение true, включается логика «перезапросов» доступа к файлу из разных контуров. Пример: пользователь ранее запросил доступ к файлу из корпоративной сети, и DLP-система вернула позитивный результат проверки. Далее пользователь запросил доступ к тому же файлу из внешней сети. Если для параметра установлено значение true, в DLP-систему будет направлен «перезапрос» доступа. Перезапросы выполняются последовательно - от самого доверенного до менее доверенного. Результатом будет минимальный уровень доступа, которым должен обладать пользователь, чтобы просмотреть/скачать файл..
- default_access_level— уровень доступа к файлам по умолчанию, используется при выключеннном параметре access_levels_recheck.
- access_levels — уровни доступа к файлу:
  - External — файл доступен вне корпоративной системы передачи данных (интернет/внешняя среда).
  - Internal — файл доступен только в корпоративной системе передачи данных. Обязательное требование к порядку уровней – от самого доверенного к менее доверенному уровню доступа, иначе будут получены некорректные результаты проверки и «перепроверки».
- check_file_chunk_bytes — размер передаваемых чанков.

В секции **adapters_manager** укажите настройки управления адаптерами:

- default_adapter_name — укажите адаптер, используемый по умолчанию.

2. Примените изменения.

   Для инсталляции на одну виртуальную машину выполните команду:

   ```
   HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
   ```

   Для кластерной инсталляции:

   ```
   HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
   ```

Шаг 4. Создайте плагин для подключения провайдера

Данный шаг актуален только для DLP-системы InfoWatch. Если вы используете другую DLP-систему, перейдите к следующему шагу.

Подключение провайдера данных осуществляется через подключение плагина. В плагине содержится информация о поставщике данных, типы обрабатываемых событий и пользовательские заголовки. Подробнее про регистрацию сторонних компонентов можно посмотреть в документации InfoWatch — https://kb.infowatch.com/pages/viewpage.action?pageId=217787144

Для создания плагина создайте файл **manifest.json**. Пример manifest.json-файла плагина:

```
{
    "PLUGIN_ID": "346227C2657C4701B86892CAE732805D",
    "DISPLAY_NAME": "Плагин для события мессенджера",
    "DESCRIPTION": {
        "eng": "IM events reception",
        "rus": "Прием событий менеджера"
    },
    "VERSION": "0.0.1",
    "VENDOR": "VKteams",
    "LICENSE": [
        {
            "PATH": "licenses/tm_license.license"
        }
    ],
    "PATTERN_SEARCH_LICENSE": {
        "operator": "and",
        "conditions": [
            {
                "common_name": "VKteams"
            },
            {
                "object_type": "im_VKteams"
            },
            {
                "protocol": "NONE"
            }
        ]
    },
    "ADDS_SERVICES": {
        "SERVICE_TYPE": [
            {
                "SERVICE_MNEMO": "im_VKteams",
                "DATA_CLASS": [
                    "kChat",
                    "kFileExchange"
                ],
                "ICON": "icon/acme_messenger.png",
                "LOCALE": {
                    "rus": "Мессенджер VKteams",
                    "eng": "VKteams messenger"
                },
                "CONTACT_TYPE": [
                    {
                        "MNEMO": "im_VKteams",
                        "SCOPE": [
                            "person"
                        ],
                        "ICON": "icon/acme_messenger.png",
                        "LOCALE": {
                            "rus": "Аккаунт VKteams",
                            "eng": "VKteams account"
                        }
                    }
                ]
            }
        ]
    },
    "OBJECT_HEADER": [
        {
            "NAME": "VKteams_file_hash_header",
            "NOTE": {
                "rus": "Хеш файла",
```

```
                "eng": "File hash"
            },
            "DATA_CLASS": [
                "kChat",
                "kFileExchange"
            ],
            "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": "VKteams_text_chat_name_header",
            "NOTE": {
                "rus": "Название чата",
                "eng": "Chat name"
            },
            "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",
            "TYPE": "string",
            "FORMAT": "string",
            "IS_MULTIPLE_VALUE": "1"
        },
        {
            "NAME": "VKteams_text_chat_id_header",
            "NOTE": {
                "rus": "ID чата",
                "eng": "Chat ID"
            },
            "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",
            "TYPE": "string",
            "FORMAT": "string",
            "IS_MULTIPLE_VALUE": "1"
        },
        {
            "NAME": "VKteams_text_chat_participants_header",
            "NOTE": {
                "rus": "Количество участников чата",
                "eng": "Number of chat participants"
            },
            "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",
            "TYPE": "number",
            "FORMAT": "integer",
            "IS_MULTIPLE_VALUE": "1"
        },
        {
            "NAME": "VKteams_message_type_header",
            "NOTE": {
                "rus": "Тип отправленного сообщения",
                "eng": "Message type"
            },
            "DATA_CLASS": [
                "kChat",
                "kFileExchange"
            ],
            "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": "VKteams_sender_ip_header",
            "NOTE": {
                "rus": "IP отправителя",
                "eng": "Sender IP"
            },
            "DATA_CLASS": [
                "kChat",
                "kFileExchange"
            ],
            "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": "VKteams_sender_ua_header",
            "NOTE": {
                "rus": "UA отправителя",
                "eng": "Sender UA"
            },
            "DATA_CLASS": [
                "kChat",
                "kFileExchange"
            ],
            "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": "VKteams_access_level_header",
            "NOTE": {
                "rus": "Уровень доступа к файлу",
                "eng": "File access_level"
            },
            "DATA_CLASS": [
                "kFileExchange"
            ],
            "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"
        }
    ]
}
```

Шаг 5. Активируйте envoy-плагин Tourniquet

Включение модуля Tourniquet опционально. Он используется для проверки текстов сообщений, включающих в себя файлы. Отправка файлов сама по себе может производиться изолировано.

Чтобы активировать плагин:

1. В конфигурационном файле **/usr/local/etc/k8s/helmwave/store/dlp.yml** установить в поле **tourniquetEnabled** значение true.
2. Примените изменения.

   Для инсталляции на одну виртуальную машину выполните команду:

   ```
   HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress
   ```

   Для кластерной инсталляции:

   ```
   HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress
   ```

3. Проверьте, что плагин запустился:

   ```
   kubectl -n istio-ingress get po | grep istio-ingress
   ```

   В выводе консоли pod должен находиться в статусе Running.

4. Проверьте логи сервиса на возможные проблемы:

```
sudo kubectl logs -n istio-ingress $(sudo kubectl get pod -n istio-ingress | grep istio-
ingress | awk '{print $1}')
```

Искать необходимо по названию модуля Tourniquet. Пример проблем в логах:

```
2024-11-21T13:52:28.946719Z error   envoy golang external/envoy/contrib/golang/common/log/
cgo.cc:24 failed to parse golang plugin config: dlp_resp_blocked_code: expect float64 while
got string    thread=14
2024-11-21T13:52:28.946961Z warning envoy config external/envoy/source/extensions/
config_subscription/grpc/delta_subscription_state.cc:269  delta config for
type.googleapis.com/envoy.config.listener.v3.Listener rejected: Error adding/updating
listener(s) 0.0.0.0_80: golang filter failed to parse plugin config: tourniquet /var/lib/
im/modules/tourniquet.so
```

Подобная запись в логах говорит о проблемах в конфигурации — необходимо перепроверить конфигурацию или вернуть конфиг к значению по умолчанию.

Шаг 6. Активируйте сервис Watchman

1. В конфигурационном файле **/usr/local/etc/k8s/helmwave/store/dlp.yml** установите в поле **watchmanEnabled** значение **true**.
2. Примените изменения.

   Для инсталляции на одну виртуальную машину выполните команду:

   ```
   HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress
   ```

   Для кластерной инсталляции:

   ```
   HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress
   ```

3. Проверьте, что сервис запустился:

   ```
   kubectl -n istio-ingress get po | grep istio-ingress
   ```

   В выводе консоли pod должен находиться в статусе Running.
4. Проверьте логи сервиса на возможные проблемы:

   ```
   sudo kubectl logs -n istio-ingress $(sudo kubectl get pod -n istio-ingress | grep istio-
   ingress | awk '{print $1}')
   ```

   Искать необходимо по названию модуля Watchman. Пример проблем в логах:

   ```
   2024-11-21T13:52:28.946719Z error   envoy golang external/envoy/contrib/golang/common/log/
   cgo.cc:24 failed to parse golang plugin config: ...
   2024-11-21T13:52:28.946961Z warning envoy config external/envoy/source/extensions/
   config_subscription/grpc/delta_subscription_state.cc:269  delta config for
   type.googleapis.com/envoy.config.listener.v3.Listener rejected: Error adding/updating
   ```

```
listener(s) 0.0.0.0_80: golang filter failed to parse plugin config: watchman /var/lib/im/
modules/watchman.so
```

Подобная запись в логах говорит о проблемах в конфигурации — необходимо перепроверить конфигурацию или вернуть конфиг к значению по умолчанию.

Шаг 7. Настройте правила доступа к файлам

Модуль Watchman рассматривает запросы на получение доступа к содержимому файла (просмотр и скачивание) и проставляет к пользовательскому запросу уровень доступа на основе конфигурируемых правил. На данный момент есть два уровня доступа: External или Internal.

Уровень доступа для пользователя определяется по IP-адресу и устройству, с которого был отправлен запрос на доступ к файлу.

Механизм работы правил доступа пользователей к файлу

Правила настраиваются в конфигурационном файле **/usr/local/etc/k8s/helmwave/store/dlp.yml** в секции **watchmanAccessLevels.** Пример конфигурации с правилами:

```
tourniquetEnabled: true
watchmanEnabled: true
watchmanAccessLevels:
  - access_level: internal
    subnets:
      - 192.168.0.1/24
    Device: []
    OS:
      - IOS
    Browser: []
  - access_level: internal
    subnets: []
    Device: []
    OS: []
    Browser:
      - Opera
```

Каждое правило (access_level) состоит из секций селекторов:

- subnets.
- Device.
- OS.
- Browser.

Если секция отсутствует или пуста, проверка на соответствующий признак не будет осуществляться в рамках правила (пример: пустая секция subnets в правиле означает, что под правило попадает любой IP-адрес).

Данные пользователя проверяются на соответствие перечисленным правилам последовательно сверху вниз до первого подошедшего правила. Когда данные пользователя совпадают с данными из селекторов, запросу пользователя присваивается уровень доступа к файлу - External или Internal.

Чтобы настроить правила для определения уровня доступа к файлам:

1. Перейдите в конфигурационный файл **/usr/local/etc/k8s/helmwave/store/dlp.yml** и укажите правила в секции **watchmanAccessLevels**.

   По умолчанию секция **watchmanAccessLevels** содержит пустые элементы с ключами External и Internal. Если их оставить как есть, то к каждому запросу пользователя будет применятся уровень доступа External, так как он расположен первым.

   Значения в секциях селекторов не чувствительны к регистру. Возможные значения селекторов:

   Device:

   - mobile
   - desktop
   - web

   Browser (настраивается для веб-приложений):

   - chrome
   - firefox
   - opera
   - ie
   - safari
   - edge
   - yandex

   OS (настраивается для веб-приложений):

   - android
   - windows
   - macos
   - ios
   - linux

   Для декстоп-устройств не поддерживается определение операционных систем.

   Для мобильных устройств поддерживается определение iOS и Android.

2. Примените изменения.

   Для инсталляции на одну виртуальную машину выполните команду:

   ```
   HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress -t apigw
   ```

   Для кластерной инсталляции:

   ```
   HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress -t apigw
   ```

Примечание

Если вы используете правила Watchman для определения уровня доступа, вы можете изменить HTTP-заголовок, который используется для получения IP-адреса клиентского приложения, сделавшего запрос в DLP-систему:

1. Перейдите в конфигурационный файл **/usr/local/etc/k8s/helmwave/projects/istio/values/istio-ingress.yml** и в секции **watchman** добавьте следующие поля:

```
dlp_ip_source_header: "x-forwarded-for" # заголовок, который будет использоваться для
извлечения IP-адреса клиента, сделавшего запрос
default_access_level: "external"  # уровень доступа к файлу, который устанавливается по
умолчанию (если ни одно из правил не подошло)
```

   Пример конфигурационного файла:

```
plugins:
  # ...
  watchman:
      defined: {{.Release.Store.watchmanEnabled}}
      plugin_config: &watchman_plugin_config
          dlp_access_levels_enable: true          # deprecated
          dlp_access_levels_path_suffixes: []     # deprecated
          dlp_ip_source_header: "x-forwarded-for"
          default_access_level: "external"
          access_levels:
  {{.Release.Store.watchmanAccessLevels | toYaml | indent 8}}
        routes:
          watchman:
              plugin_config: *watchman_plugin_config
```

   Все значения кроме **default_access_level** являются техническими, изменять их не рекомендуется.

2. Примените изменения. Для инсталляции на одну виртуальную машину выполните команду:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress -t apigw
```

   Для кластерной инсталлции:

```
HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress -t apigw
```

Шаг 8. Настройте сервис Go-files

Перейдите в конфигурационный файл сервиса Go-files **/usr/local/go.files.icq.com/files.icq.com.config.yaml** и укажите настройки отправки файлов в DLP-систему:

1. В секции **DLPv2** укажите настройки для проверки файлов:

```
DLPv2:
    dlp_checks_timeout: 20m
    dlp_access_levels_checks_enable: false
    dlp_checks_enable: true
```

```
    dlp_status_on_timeout: "Outdated"
    dlp_access_level_on_timeout: "External"
```

где:

- dlp_checks_timeout — таймаут, при достижении которого статус проверки файла переходит в «Outdated» или «Ok», если DLP-система не вернула результат проверки.
- dlp_access_levels_checks_enable — если true, будет проверяться уровень доступа к файлу с устройства пользователя.
- dlp_checks_enable — если false, отключается отправка файла на проверку в сервис Vahter и отключается проверка уровня доступа пользователя к файлу.
- dlp_status_on_timeout — статус, выставляемый при достижении таймаута dlp_checks_timeout, если DLP-система не вернула результат проверки файла.

  Доступные значения: «Outdated», «Ok».
- dlp_access_level_on_timeout — уровень доступа, выставляемый при достижении таймаута dlp_checks_timeout, если DLP-система не вернула результат проверки файла. Доступные значения: Internal, External.

2. В секции **mysql_cluster_optimize** включите миграцию статусов проверки файлов:
3. `yaml`
   `mysql_cluster_optimize:`
   `login_admin_env_key: FILES_ADMIN_LOGIN`
   `password_admin_env_key: FILES_ADMIN_PASSWORD`
   `migrations_enabled: true`
   `migrations_path: file:////usr/local/go.files.icq.com/migrations`

   где migrations_enabled — если true, миграция статусов проверки файлов включена.
4. Перезапустите сервис Go-files командой:

```
sudo systemctl restart gofiles_httpd
```

5. Для включения автоудаления файлов при блокировке DLP-системой выполните настройки.

Шаг 9. Настройте маршрутизацию по доменам

Данный шаг является опциональным. Вы можете настроить выбор адаптера в зависимости от домена отправителя сообщения. Перейдите в конфигурационный файл **/usr/local/etc/k8s/helmwave/store/dlp.yml** и заполните секцию **adapter_chooser**:

```
adapter_chooser:
  enabled: false
  adapters_rules:
  - target: "solar_dozor" # адаптер, в который будут направляться сообщения от domain_list
    domain_list:
    - "domain_1"
    - "domain_2"
  - target: "info_watch"
```

```
    domain_list:
    - "domain_3"
    - "domain_4"
```

1. Установите для параметра **enabled** значение true. Если false (конфигурация по умолчанию) — для всех доменов будет использоваться адаптер из параметра **default_adapter_name** конфигурационного файла **/usr/local/etc/k8s/helmwave/store/dlp.yml**.
2. В секции **adapters_rules** указываются правила сопоставления домена и адаптера. Укажите для параметров **target** нужный адаптер и список доменов, для которых этот адаптер будет использоваться.

Шаг 10. Настройте исключения из проверок

Данный шаг является опциональным. Вы можете настроить список пользователей, исключенных из проверок. Файлы, загружаемые такими пользователями, не будут попадать в DLP-систему и не будут блокироваться. С точки зрения системы таким файлам будет установлен положительный результат проверки, уровень доступа будет соответствовать параметру **default_access_level** выбранного адаптера.

Если вы настроили маршрутизацию по доменам:

Перейдите в конфигурационный файл **/usr/local/etc/k8s/helmwave/store/dlp.yml** и укажите email-адреса пользователей в секции **adapters_rules.excluded_users**:

```
adapter_chooser:
  enabled: false
  adapters_rules:
  - target: "solar_dozor" # адаптер, в который будут направляться сообщения от domain_list
    domain_list:
    - "domain_1"
    - "domain_2"
    excluded_users: # исключаем пользователей из проверок
    - "i.ivanov@company_domain_1"
    - "p.petrov@company_domain_2"
  - target: "info_watch"
    domain_list:
    - "domain_3"
    - "domain_4"
```

Примените изменения.

Для инсталляции на одну виртуальную машину выполните команды:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress
```

Для кластерной инсталляции:

```
HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t istio-ingress
```

Если вы не настраивали маршрутизацию по доменам:

Перейдите в конфигурационный файл **/usr/local/etc/k8s/helmwave/store/dlp.yml** и укажите email-адреса пользователей в секции **adapters_manager.excluded_users**:

```
adapters_manager:
  icap_debug: false
  default_adapter_name: search_inform
  excluded_users: # исключаем пользователей из проверок
    - "i.ivanov@company_domain_1"
    - "p.petrov@company_domain_2"
```

Примените изменения.

Для инсталляции на одну виртуальную машину выполните команду:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
```

Для кластерной инсталляции:

```
HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
```

Настройка видимости статуса проверки сообщений

Вы можете настроить для пользователей видимость статусов проверки для файлов и текстовых сообщений. В таком случае до получения ответа от DLP-системы отправитель будет видеть иконку  у сообщений, содержащих файл или текст. После получения от DLP-системы положительного результата проверки отправитель увидит статус «Отправлено»  . При получении отрицательного результата проверки для сообщения будет отображаться иконка 

Чтобы настроить видимость статуса проверки для файлов:

1. Перейдите в конфигурационный файл сервиса Go-files **/usr/local/go.files.icq.com/files.icq.com.config.yaml** и укажите в секции **scribl** таймаут 5 секунд:

```
{
    scribl:
       timeout: 5s
}
```

2. Перезапустите сервис Go-files командой:

```
sudo systemctl restart gofiles_httpd
```

3. В конфигурационном файле **/usr/local/nginx-im/html/myteam/myteam-config.json** расширьте секцию **dlp**:

```
{
    "dlp": {
       enabled: true,
       config: {
           "ui-statuses-v1-enabled": true,
           "ui-file-statuses-v1-enabled": true,
           "ui-file-statuses-resub-delay": 300
       }
    }
}
```

4. Проверьте, что в конфигурационном файле **/usr/local/nginx-im/html/myteam/myteam-config.json** указана версия API >= 132.
5. Для инсталляции на одну виртуальную машину выполните команду:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t godmod
```

   Для кластерной инсталляции:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true HELMWAVE_ENV_NAME=cluster hwup -t godmod
```

6. Перезапустите под в технологическое окно (может приводить к сбою в новых подключениях):

```
kubectl delete pods -n vkteams -l app=myteam-admin
```

7. Проверьте, что в файле **/usr/local/etc/k8s/helmwave/projects/apigw/values/apigwv2.yml** в разделе **[[tpl (.Release.Store.istioIngress.cors | toYaml) . | indent 10]]** есть секция:

```
- ignoreUriCase: true
            uri:
              regex: '/api/v\d\d\d/files/info/.*'
```

Раздел должен выглядеть так:

```
[[ tpl (.Release.Store.istioIngress.cors | toYaml) . | indent 10 ]]
        match:
          - ignoreUriCase: true
            uri:
              regex: '/api/v\d\d\d/files/get/.*'
          - ignoreUriCase: true
            uri:
              regex: '/api/v\d\d\d/files/preview/.*'
          - ignoreUriCase: true
            uri:
              regex: '/api/v\d\d\d/files/info/.*'
```

Не вносите изменения самостоятельно. Если конфигурация отличается, обратитесь в техническую поддержку.

Чтобы настроить видимость статуса проверки для текстовых сообщений:

1. В конфигурационном файле **/usr/local/nginx-im/html/myteam/myteam-config.json** расширьте секцию **dlp**:

```
 {
     "dlp": {
        enabled: true,
        config: {
            "ui-statuses-v1-enabled": true,
        }
     }
  }
```

2. Для инсталляции на одну виртуальную машину выполните команду:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t godmod
```

Для кластерной инсталляции:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true HELMWAVE_ENV_NAME=cluster hwup -t godmod
```

3. Перезапустите под в технологическое окно (может приводить к сбою в новых подключениях):

```
kubectl delete pods -n vkteams -l app=myteam-admin
```

Решение проблем

«Грязная» база в сервисе Go-files

Ошибка: Dirty database version 1. Fix and force version

1. По логам сервиса Go-files определите шард (базу), которая стала «грязной» в результате миграций. Например, такое может происходить в случае, когда неправильно написан запрос в файле миграций.
2. Укажите верные настройки миграции в конфигурационном файле сервиса Go-files **/usr/local/go.files.icq.com/files.icq.com.config.yaml**.
3. Удалите таблицу schema_migrations из шарда.
4. Перезапустить сервис Go-files командой:

```
systemctl restart gofiles_httpd
```

Бесконечные ретраи метода отправки сообщений sendIM

Проблема: Статус-код 500 при ответе на sendIM на устройствах iOS и Android вызывает ретрай. Запрос повторяется до момента, пока сервер не вернет статус-код, отличный от 500. Таким образом сервис Vahter может стать недоступным.

Данная проблема может быть связана с недоступностью DLP-системы.

Чтобы решить эту проблему, в конфигурационном файле **/usr/local/etc/k8s/helmwave/store/dlp.yml**.

1. Укажите для параметра strategy_on_fail значение true. Тогда результат проверки будет положительным при недоступности или ошибке запроса в DLP-систему.
2. Укажите для параметра debug_mode значение true. Результат проверки будет положительным без ожидания ответа на запрос в DLP-системы.
3. Примените изменения.

   Для инсталляции на одну виртуальную машину выполните команду:

   ```
   HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
   ```

   Для кластерной инсталляции:

   ```
   HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
   ```

Таймауты при проверке текста/файлов

При неправильно конфигурации сервиса Vahter запрос в DLP-систему обрывается по таймауту.

Причины:

1. Низкие значения параметров в конфигурационном файле **/usr/local/etc/k8s/helmwave/store/dlp.yml**:
   - check_text_disconnect_timeout.
   - check_file_disconnect_timeout.
   - icap_client_timeout — сервис не успел установить соединение с DLP-системой по ICAP.
2. Низкие значения параметра max_timeout в конфигурационном файле сервиса Go-files **/usr/local/go.files.icq.com/files.icq.com.config.yaml** — не успел сформироваться id файла в облаке.

Решение:

1. Увеличьте таймауты.

   При выставлении таймаутов при проверке файлов нужно учитывать, что параметр check_file_disconnect_timeout должен быть больше max_timeout, т.к. при проверки файлов используется контекст с таймаутом, который формируется на основе check_file_disconnect_timeout.
2. Примените изменения командой:

```
HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup -t vahter
```

Файлы недоступны для проверки/недоступно файловое хранилище

Получение файлов осуществляется через сервис Go-files, используется механизм ретраев.

Если файлы недоступны для проверки или недоступно файловое хранилище, вы можете изменить настройки взаимодействия сервисов Go-files и Multifora в конфигурационном файле **/usr/local/etc/k8s/helmwave/projects/vahter/values/vahter.yml**:

```
files:
  host: "{{ .Release.Store.filesComHost }}" #хост Go-files
  timeout: 5m                               #таймаут подключения
  retries:                                  #ретраи при получении файла
    max_retry: 5                            #количество ретраев
    max_timeout: 9m                         #временной промежуток, в который выполняются
ретраи

callback_handler:
  address: multifora.vkteams.svc.cluster.local:44438 #хост Multifora, не изменяйте этот
параметр
  timeout: 3s                                        #таймаут подключения
```

Технический писатель: Белова Ирина

25 марта 2026 г.