Платформа Радар

Руководство по подключению источников

Версия 3.1.0

ООО Пангео Радар

Оглавление

1. Общее описание процесса подключения источников

Руководство по подключению источников содержит рекомендации и инструкции для настройки Платформы для приема событий в пассивном и активном режимах, настройки источников, настройки лог-коллектора, а также обработки событий, включая фильтрацию и обогащение.

1.1. Пассивный сбор

В Платформе присутствует возможность приема событий от источников в пассивном режиме. Для этого необходимо в веб-интерфейсе Платформы настроить прием событий: включить поддерживаемые источники или создать и настроить новые источники, которые смогут самостоятельно отправлять данные. Подробное описание включения и создание источников

дано в разделе «Работа с пассивными источниками событий»

1.2. Активный сбор

Для организации активного сбора необходимо использовать лог-коллектор. Он предназначен для организации сбора событий от активов, не имеющих возможности самостоятельной отправки данных в сторонние системы. Подробное описание настройки лог-коллектора дано в разделе «Руководство по настройке лог-коллектора. Активные источники событий».

1.3. Процесс подключения типового источника

Подключение типового источника осуществляется в три этапа:

1. Настройка Платформы на прием событий путем включения необходимого источника в веб-интерфейсе Платформы. После включения источника Платформа готова к приему событий в пассивном режиме. Подробнее о включении типовых источников в разделе «Работа с пассивными источниками событий».
2. Настройка лог-коллектора, если необходимо организовать активный сбор или реализовать цепочку по пересылке событий между двумя и более лог-коллекторами. Подробная настройка лог-коллектора описана в «Руководство по настройке лог-коллектора. Активные источники событий»
3. Настройка источника. Настройка производится согласно рекомендациям производителя или в соответствии с разделом с описанием поддерживаемых источников их настройки.

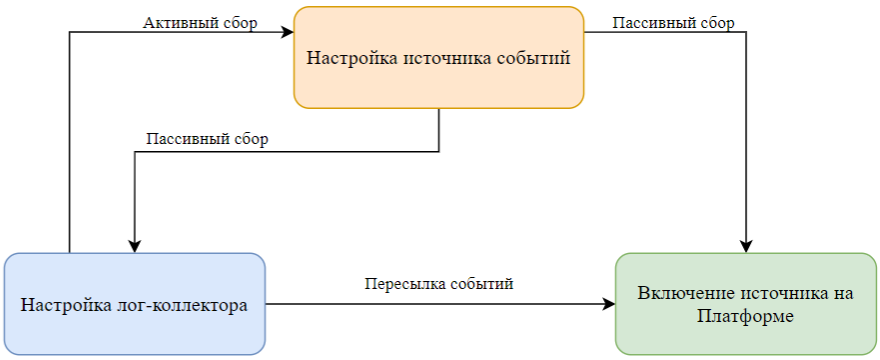

Рисунок 1

Рисунок 1. Добавление типового источника

1.4. Процесс подключения нетипового источника

Подключение нетипового источника осуществляется в пять этапов:

1. Настройка Платформы на прием событий путем создания нового пассивного источника событий в веб-интерфейсе Платформы. После включения источника Платформа готова к приему событий в пассивном режиме. Подробнее о создании новых источников в разделе «Работа с пассивными источниками событий»
2. Настройка Лог-коллектора, если необходимо организовать активный сбор или реализовать цепочку по пересылке событий между двумя и более Лог-коллекторами. Подробная настройка Лог-коллектора описана в "Руководство по настройке лог-коллектора. Активные источники событий"
3. Настройка источника. Настройка производится согласно рекомендациям производителя или в соответствии с разделом с описанием поддерживаемых источников и их настройки.

4. Создание правил разбора для событий с нового источника. Подробнее в разделе про форматы правил разбора
5. Создание правил нормализации для событий с нового источника. Подробнее в разделе «Разработка правил разбора и нормализации»

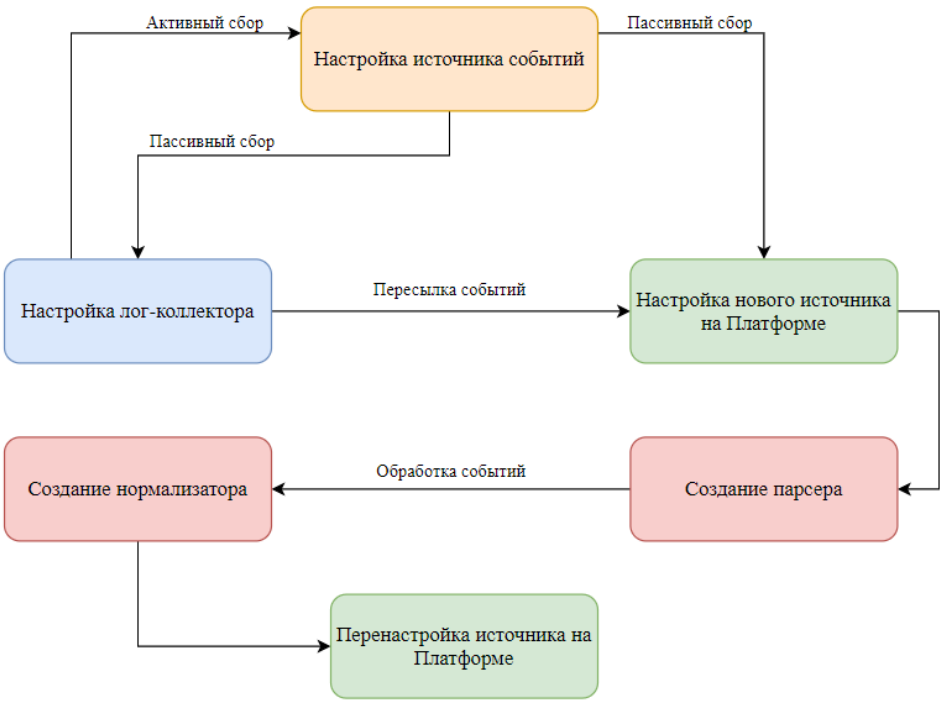

Рисунок 2

Рисунок 2. Добавление нетипового источника

1.5. Проверка получения данных от источников

Выполнить проверку поступающих данных можно в веб-интерфейсе Платформы в разделе «**Инциденты**» — «**Просмотр событий**», выставив необходимые временные фильтры.

2. Работа с пассивными источниками событий

Все действия по управлению пассивными источниками в веб-интерфейсе Платформы выполняются в разделе «Администрирование»-> «Источники»->«Управление источниками» (Рисунок 1).

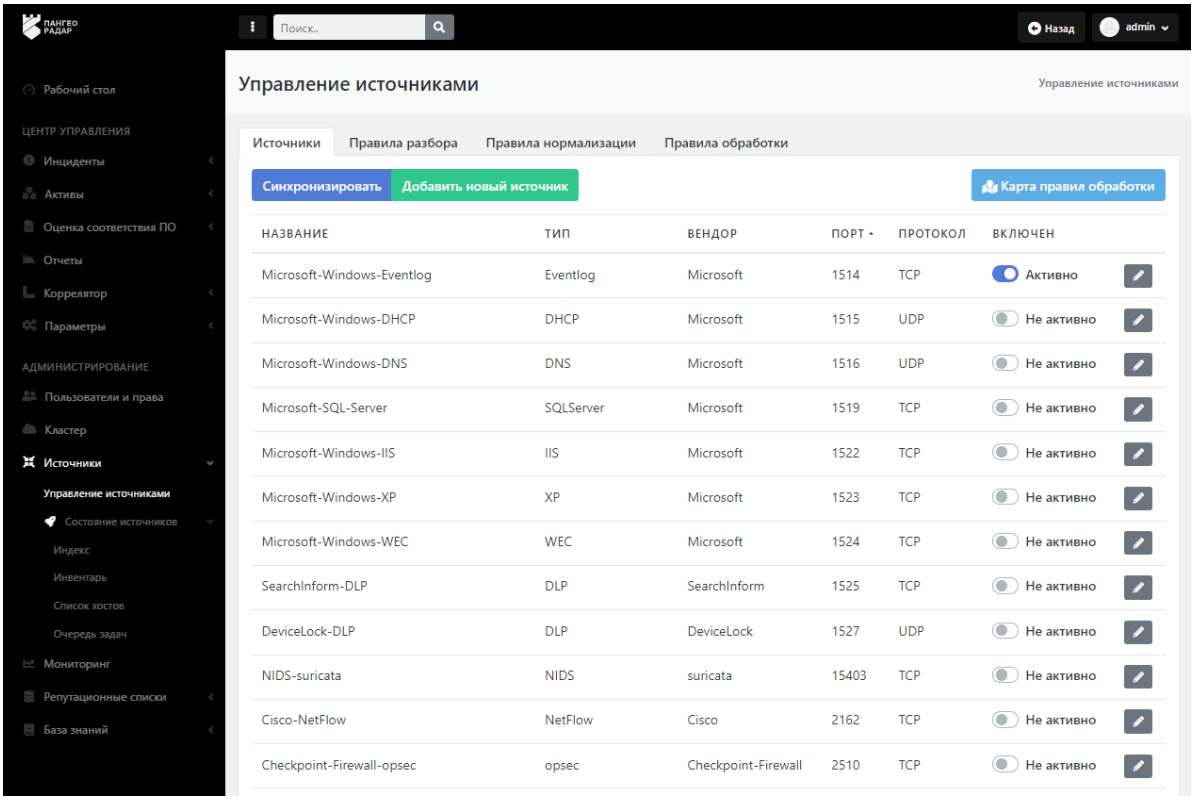

Рисунок 3

Рисунок 1. Управление источниками

2.1. Включение/выключение пассивных источников и их синхронизация

Для включения/выключения источников необходимо выполнить следующие действия:

1. Выбрать источник и проверить его текущий статус работы в столбце **ВКЛЮЧЕН**: **синий фон** кнопки-переключателя и надпись **Активно** показывают, что источник включен, **белый фон** и надпись **Не активно**, что выключен (Рисунок 2).

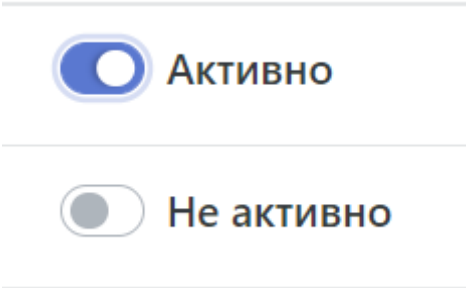

Рисунок 4

Рисунок 2. Статус работы источника

2. Включить/выключить все необходимые источники. Включение и выключение источника осуществляется нажатием на кнопку-переключатель.
3. Выполнить синхронизацию источников, чтобы внесенные изменения вступили в силу, нажав кнопку **Синхронизировать** (Рисунок 3).

> **Важно!** Необходимо синхронизировать источники после каждого изменения

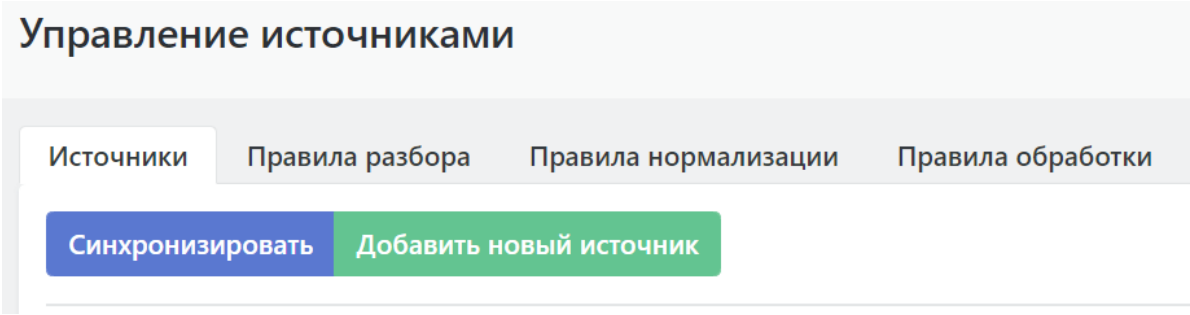

Рисунок 5

Рисунок 3. Кнопки для синхронизации и добавления нового источника

После выполнения вышеперечисленных действий Платформа готова к приему событий от включенных источников в пассивном режиме.

2.2. Заведение нового пассивного источника

Для заведения нового пассивного источника:

1. Нажать кнопку **Добавить новый источник** в верхней части страницы «Управление источниками» (Рисунок 1,3).
2. Заполнить форму (Рисунок 4). Расшифровка полей формы дана в следующем разделе *«Описание полей формы»*.
3. При необходимости проверить работу выбранных парсера и нормализатора в правой части формы, подставив сырое событие от источника и запустив проверку. Результаты проверки появятся во всплывающем окне.
4. Сохранить данные, нажав кнопку **Сохранить** в нижней левой части формы. Для выхода из формы без сохранения данных нажмите кнопку **Отменить**.
5. Включить новый источник.
6. Синхронизировать источники.

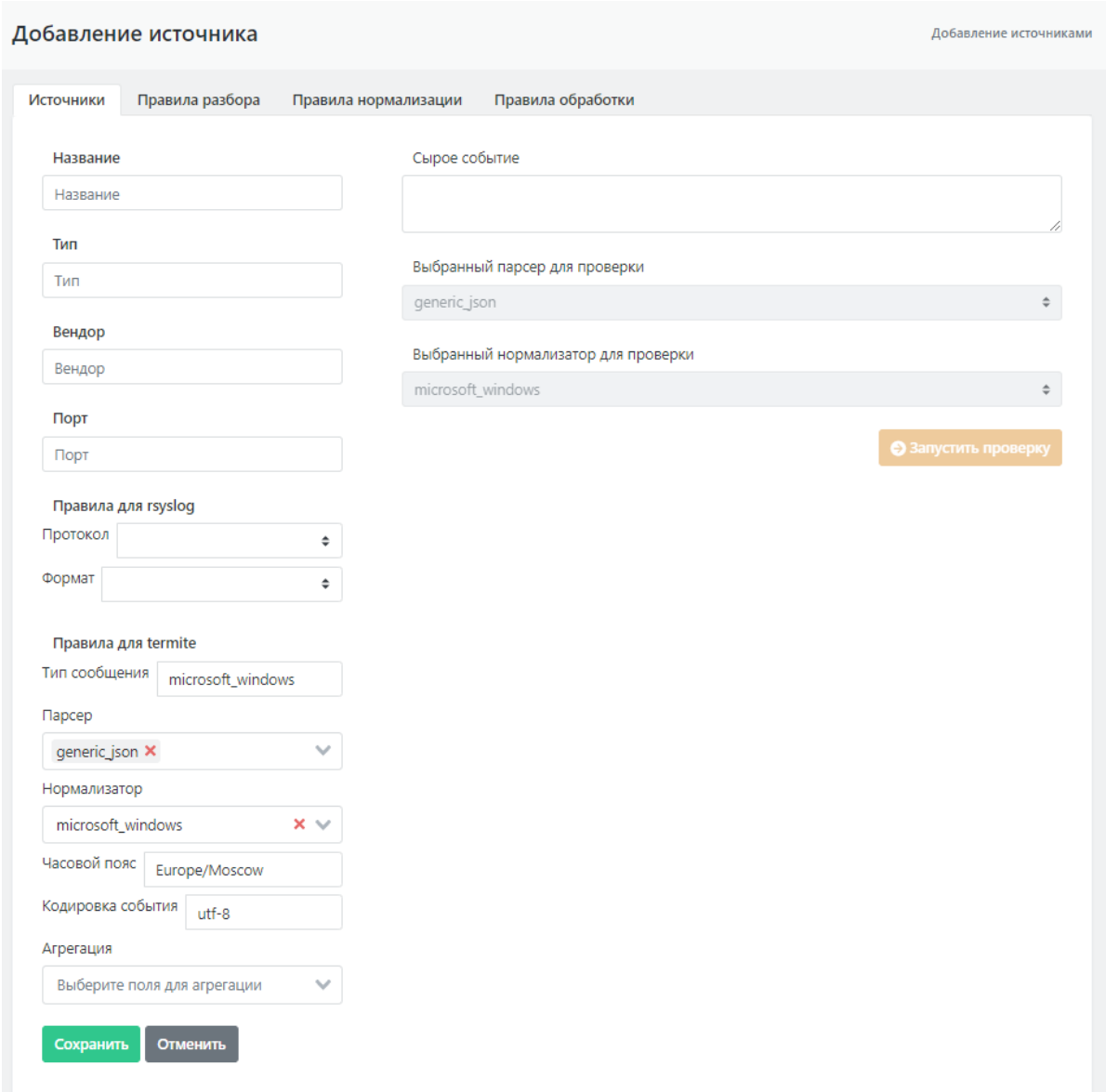

Рисунок 6

Рисунок 4. Форма добавления нового типа источника

2.3. Описание полей формы создания/ редактирования пассивного источника

Название — наименование типа источника (пример: «Linux Debian»)
Тип — тип источника (пример: «Linux»)
Вендор — производитель системы, которая выступает в качестве источника (пример: «Debian»)
Порт — необходимо указать один из свободных портов, который будет использоваться для отправки события с нового источника (+- диапазон 6000-8000)

Область **Правила для rsyslog**:

- Поле **Протокол** — протокол, по которому будут приниматься события. Возможные форматы:
  - **TCP**,
  - **PTCP** (Plain TCP),
  - **UDP**.
- Поле **Формат** — правила приема и обработки события. Возможные форматы:
  - **RAW** — не изменять входящее событие

- **RAW-JSON** — обогатить сообщение дополнительной технической информацией и упаковать в пакет json
- **JSON-JSON** — обогатить существующую структуру json дополнительными полями с технической информацией

Область **Правила для termite**:

- **Тип сообщения** — указывается тип события из правила нормализации.
- **Парсер** — указывается правило разбора данного типа событий.
- **Нормализатор** — указывается правило нормализации.
- **Часовой пояс** — указывается необходимая временная зона (пример: «Europe/Moscow»).
- **Кодировка событий** — указывается необходимая кодировка (пример: «utf-8»).
- **Агрегация** — позволяет выполнить агрегацию однотипных событий. Необходимо указать поля, которые могут меняться. Расшифровка полей для агрегации дана в разделе «Описание полей нормализации».

2.4. Изменение параметров пассивного источника

Для изменения данных об источнике необходимо выполнить следующие действия:

1. Нажать кнопку редактирования  в строке выбранного источника.
2. Внести необходимые изменения в форму редактирования источника (Рисунок 5).
3. Сохранить данные, нажав кнопку **Сохранить** в нижней левой части формы. Для выхода из формы без сохранения данных нажать кнопку **Отменить**.
4. Синхронизировать источники нажав кнопку **Синхронизировать** на вкладке "Источники".

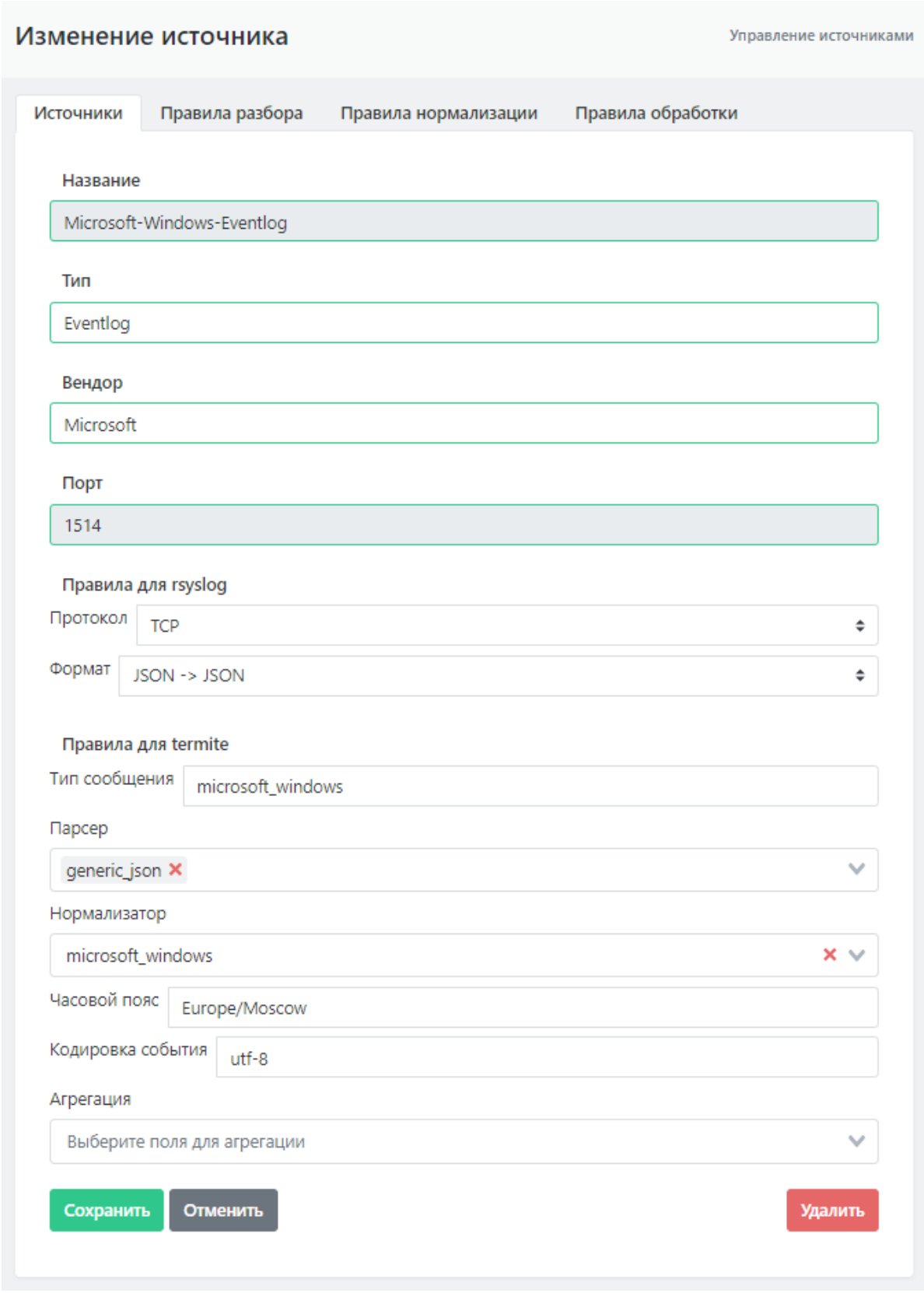

Рисунок 7

Рисунок 5. Форма редактирования типа источника

2.5. Удаление пассивного источника

Для удаления пассивного источника:

1. На вкладке "Источники" нажать кнопку редактирования  в строке выбранного для удаления источника.

2. В форме редактирования параметров источника нажать кнопку **Удалить** в нижней правой части формы (Рисунок 5). Подтвердить действие во всплывающем окне.
3. Синхронизировать источники после удаления одного из них нажав кнопку **Синхронизировать** на вкладке "Источники".

3. Список поддерживаемых источников

Данный раздел содержит перечень систем, которые могут быть подключены к Платформе Радар в качестве источников событий

3.0.1. Операционные системы

Наименование	Версия	Примечание
Microsoft Windows	XP, 7+	
Microsoft Windows Server	2003, 2008+	
Red Hat Enterprise Linux (RHEL)	6, 7, 8	
Debian Linux	8, 9, 10	
Ubuntu Linux	16.04+	
SUSE Linux Enterprise	11.3, 12	
Fedora Linux	30, 31	
CentOS Linux	7, 8	
IBM AIX	7.1, 7.2	
Oracle Solaris	10, 11	
Astra Linux		

3.0.2. Решения Endpoint Security

Наименование	Версия	Примечание
Kaspersky Security Center	10, 11	
McAfee ePolicy Orchestrator	5.9, 5.10	
PaloAlto Traps		
Symantec Endpoint Protection	14	
FireEye HX		

3.0.3. Решения Network Security

Наименование	Версия	Примечание
Barracuda Firewall		
Bluecoat Proxysg	6, 7	
Checkpoint NGFW	77, 80	log export(syslog)
OPSEC LEA		
Cisco ASA		
Cisco Firepower		
Trend Micro TippingPoint		
Fortinet Fortigate	5, 6	
McAfee Web Gateway		
PaloAlto NGFW	7, 8	
Suricata IDS		
SecurityCode Continent	3.7, 3.9	
SecurityCode Continent IDS		
Radware DefencePro		

3.0.4. Решения Application Security

Наименование	Версия	Примечание
F5 BIG-IP	15	

3.0.5. Сетевые устройства

Наименование	Версия	Примечание
Cisco IOS		
Cisco Netflow		
Infoblox Trinzic		

3.0.6. Системы управления базами данных

Наименование	Версия	Примечание
Microsoft SQL Server	2014+	
Oracle Database		

Наименование	Версия	Примечание
PostgreSQL	9+	
Oracle MySQL		

3.0.7. Системы защиты электронной почты

Наименование	Версия	Примечание
SEPPmail Secure Email	9	

3.0.8. Системы контроля привилегированного доступа

Наименование	Версия	Примечание
CyberArk PAM		
RSA SecurID		

3.0.9. Инфраструктурные системы

Наименование	Версия	Примечание
Microsoft DNS	2008+	
ISC Bind DNS	9	
Microsoft DHCP	2008+	
HAProxy		
VMware ESXi		
VMware vCenter		

3.0.10. Системы предотвращения утечек информации

Наименование	Версия	Примечание
SearchInform DLP		
SmartLine DeviceLock DLP	8x	

3.0.11. Web-серверы

Наименование	Версия	Примечание
Microsoft IIS		
Apache		

Наименование	Версия	Примечание
Nginx		
Lighttpd		

3.0.12. Proxy-серверы

Наименование	Версия	Примечание
Squid	3.5+	

3.0.13. Другое

Наименование	Версия	Примечание
Microsoft Sysmon		
Linux Auditd		

4. Операционные системы

4.1. Microsoft Windows 7+/2008+

4.1.1. Настройка источника

1. Создание учетной записи для сбора событий.
    - Если источник находится в домене, то на контроллере домена необходимо создать учетную запись и добавить ее в группу Event Log Readers.
    - Если источник не находится в домене, то необходимо создать локальную учетную запись с аналогичным набором прав.
      Процесс создания учетной записи приведен в **Приложении 1. Создание учетной записи Microsoft Windows**.
2. При использовании межсетевого экрана на узле, необходимо сделать правило для входящих соединений.
   Настройка расширенного аудита представлена в **Приложении 2. Настройка расширенных политик аудита Windows**.

4.1.2. Включение источника на Платформе

Для информации! Включение источника в Платформе подробно представлено в разделе **Управление источниками в Платформе — Включение/выключение источников и их синхронизация**.

1. Зайдите в веб-консоль Платформы, перейти в раздел «Источники» — «Управление источниками».
2. Найдите в списке доступных источников «Microsoft-Windows-Eventlog» и включить его.
3. Нажмите на кнопку «Синхронизировать».

4.1.3. Настройка коллектора событий

Для информации! Пример конфигурационного файла с настройкой данного источника представлен в разделе **Пример конфигурационного файла лог-коллектора**. Более подробная информация о настройках лог-коллектора представлена в разделе **Руководство по настройке лог-коллектора**.

1. В конфигурационный файл лог-коллектора (config.yaml) необходимо добавить input компонента Eventlog. Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора — Компонент Eventlog**.
   Основные параметры, которые необходимо указать:

```
channel: ['<название журнала, который нужно собирать>']
```

Например:

```
channel: ['Security', 'System']
```

Заполнить вкладку remote, по следующему принципу:

```
enabled: true (включение удаленного сбора)
user: <"username в открытом или зашифрованном виде"> (имя пользователя с правами
на чтение журнала событий)
password: <"password в открытом или зашифрованном виде"> (пароль пользователя)
domain: <"домен пользователя"> (если машина не в домене - ".")
remote_servers: [<"ip-адрес удаленного узла">] (адрес/список адресов серверов для
сбора событий)
auth_method: <"метод авторизации"> (выбрать один из доступных методов
авторизации: Negotiate, Kerberos, NTLM)
```

2. После настройки компонента сбора событий (input) - необходимо настроить компонент отправки событий (output). В качестве компонента отправки событий для данного источника предусмотрено использование отправки по протоколу TCP. Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Компонент отправки событий по протоколу TCP.**
   Основные параметры, которые необходимо указать:

```
target_host: <"ip адрес или имя удаленного узла"> (адрес Платформы)
port: <"порт"> (стандартный порт для данного источника 1514)
```

3. Далее необходимо включить компоненты сбора (collectors) и отправки (senders). Пример настройки по умолчанию можно найти в **Руководство по настройке лог-коллектора — Включение компонентов**.
   Основные параметры, которые нужно указать при включении компонентов сбора:

```
collectors:
event_log:
- <<: *<"id компонента сбора"> (ID компонента сбора, который указывали при
объявлении компонента сбора)
```

Основные параметры, которые нужно указать при включении компонентов отправки:

```
senders:
tcp:
- <<: *<"id компонента отправки"> (ID компонента отправки, который указывали при
объявлении компонента отправки)
```

4. После чего необходимо произвести настройку маршрутизации событий.
   Пример настройки по умолчанию можно найти в **Руководство по настройке лог-коллектора — Маршрутизация событий**.
   Основные параметры, которые нужно указать при настройке маршрута:

```
route_1: &route_1

collector_id:

- <"id компонента сбора">

sender_id:

- <"id компонента отправки">
```

5. После нужно включить маршрут в разделе routers. Пример включения маршрута:

```
routers:

- <<: *<название маршрута> (например - <<: *route_1)
```

4.2. Приложение 1. Создание учетной записи Microsoft Windows. {: annex1win>}

4.2.1. Создание учетной записи

Для создания учетной записи необходимо выполнить следующие действия:

1. В панели управления Windows открыть консоль Computer Management (Управление компьютером).
2. В консоли открыть раздел:

   System Tools (Служебные программы) → Local Users and Groups (Локальные пользователи и группы) → Users (Пользователи).
3. В контекстном меню раздела Users (Пользователи) выбрать функцию New User (Новый пользователь) для создания нового пользователя (см. Рисунок 8).

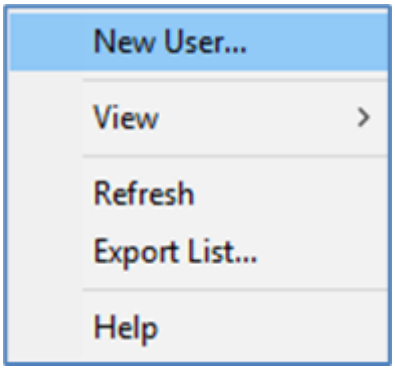

Рисунок 8 - Выбор функции создания нового пользователя.

4. В открывшемся окне New User (Новый пользователь) ввести следующие денные (см. Рисунок 9):
    - В поле Name (Имя) ввести имя нового пользователя.
    - Установить пароль в поле Password (Пароль) и подтвердить его в поле Confirm Password (Подтвердить).
    - При необходимости выставить настройки в пунктах:
    - User cannot change password (Запретить смену пароля пользователем).
    - Password never expires (Срок действия пароля неограничен).
5. Для создания пользователя с заданными параметрами нажать кнопку Create (Создать) (см. Рисунок 9).

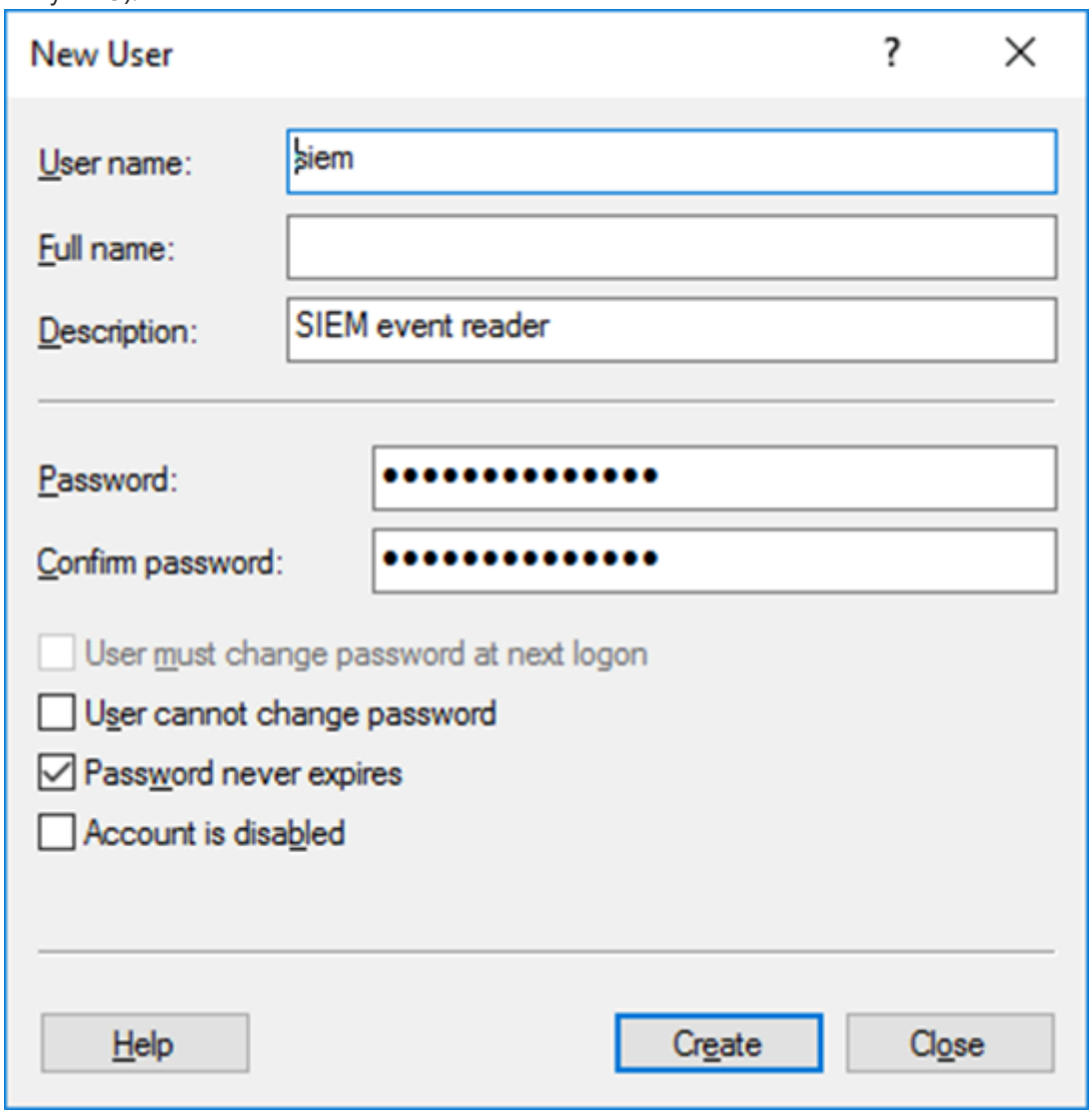

Рисунок 9 - Ввод данных нового пользователя.

4.2.2. Предоставление пользователю прав доступа к журналу событий

Для добавления пользователя в группу Event Log Readers (с правом доступа к журналам событий) необходимо выполнить следующие действия:

1. В консоли Computer Management (Управление компьютером) открыть раздел:

    System Tools (Служебные программы) → Local Users and Groups (Локальные пользователи и группы) → Groups (Группы)

2. Выбрать в списке группу Event Log Readers (Читатели журнала событий) (см. Рисунок 10).

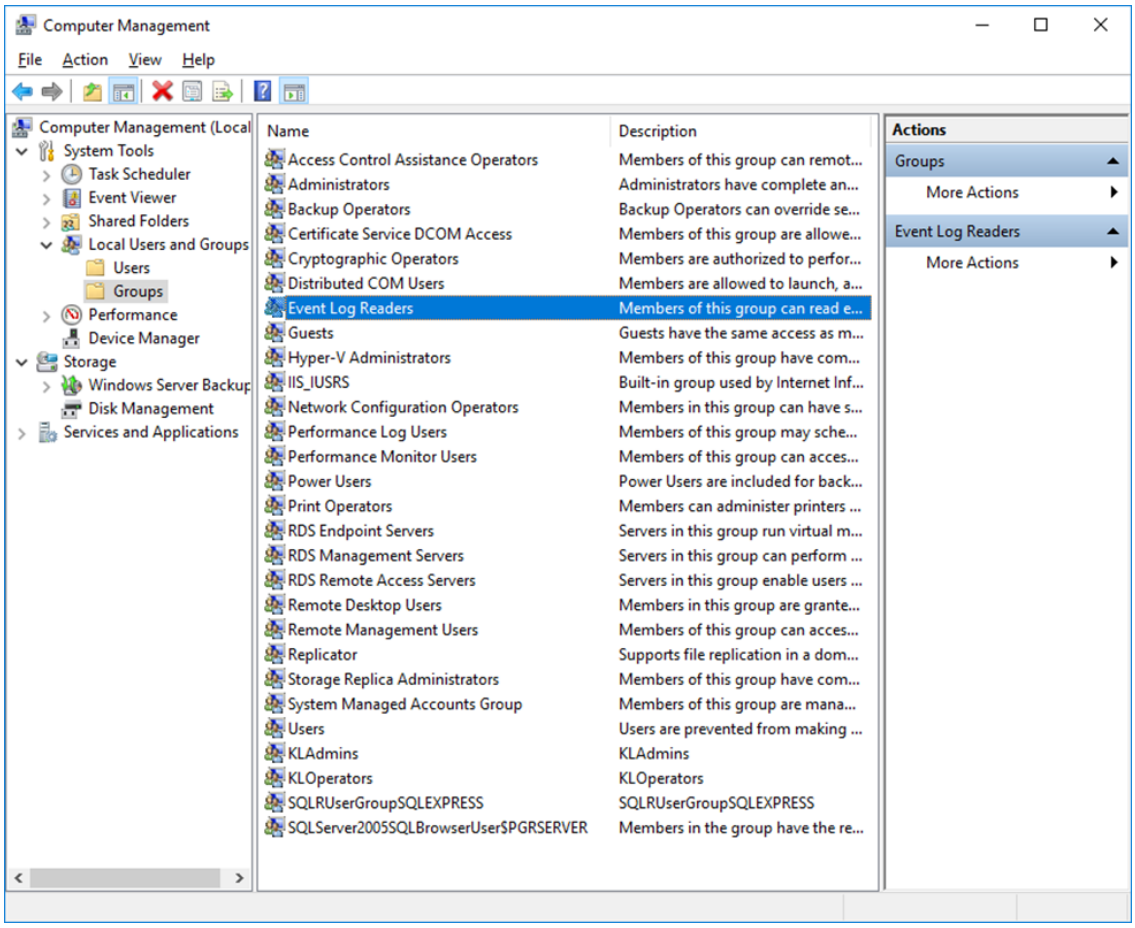

Рисунок 10 - Выбор группы Event Log Readers для включения учетной записи.

3. Открыть правой кнопкой мыши контекстное меню группы Event Log Readers (Читатели журнала событий) и выбрать пункт Add To Group (Добавить в группу). Откроется окно Event Log Readers Properties (Свойства: Читатели журнала событий) (см. Рисунок 11).

4. Для добавления пользователя в группу:

   - Нажать кнопку Add (Добавить).
   - В открывшемся окне Select Users (Выбор: Пользователи) выбрать в списке пользователя, созданного в п.3.1.1, и добавить его в группу, нажав кнопку ОК.

5. Для сохранения введенных настроек в окне Event Log Readers Properties (Свойства: Читатели журнала событий) нажать кнопку OK (см. Рисунок 11).

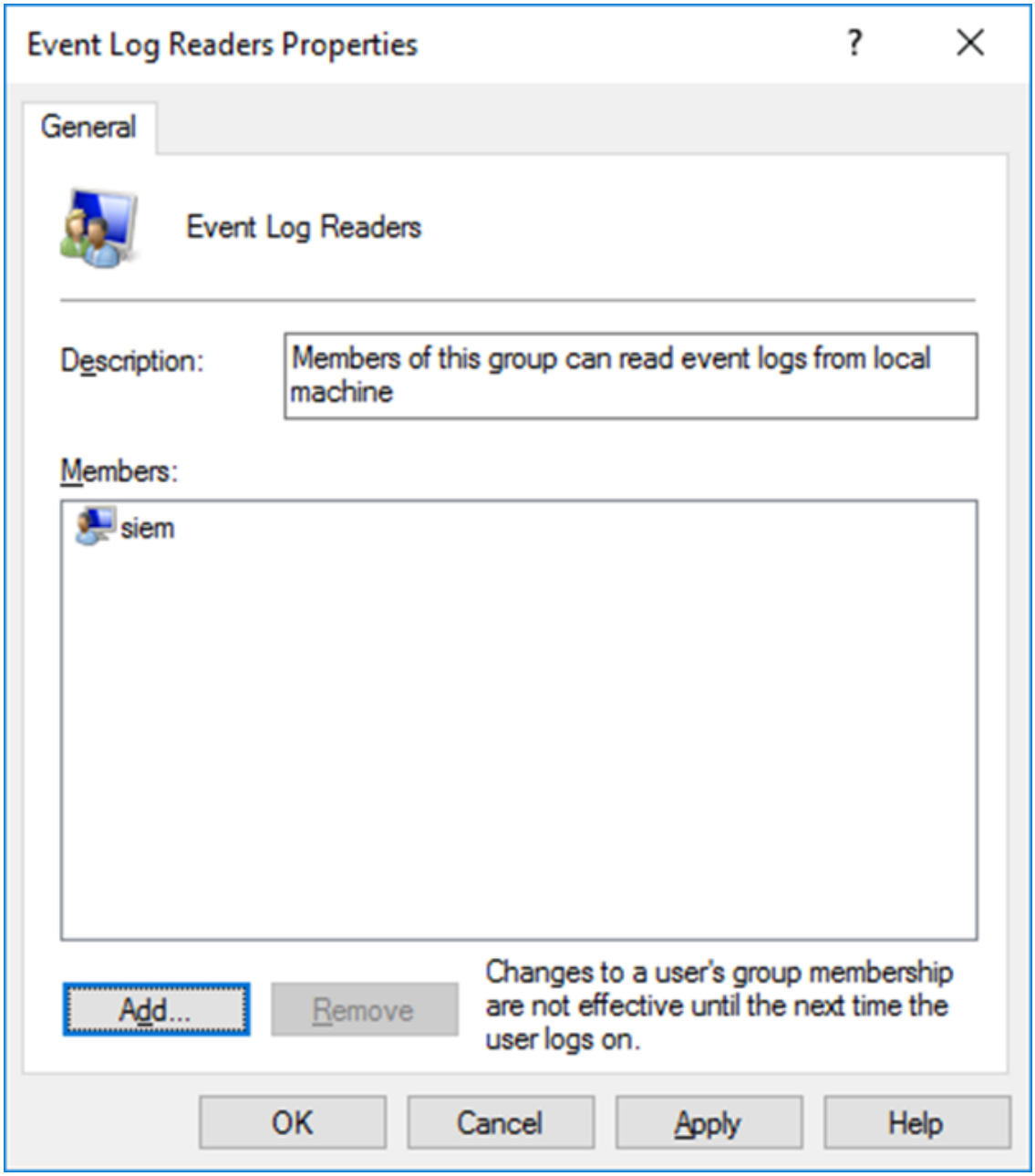

Рисунок 11 - Добавление пользователя в группу Event Log Readers.

Внесенные изменения вступают в действие при следующем входе нового пользователя в систему.

4.3. Приложение 2. Настройка расширенных политик аудита Windows.

Для настройки политик аудита на контроллерах домена используются групповые политики домена, которые необходимо сконфигурировать в соответствии с представленной инструкцией:

В групповой политике, применяемой для контроллеров домена, необходимо включить политику использования расширенной конфигурации политики аудита «Audit: Force audit policy subcategory settings (Windows Vista or later) (Аудит: принудительно переопределяет параметры категории политики аудита параметрами подкатегории политики аудита (Windows Visa или следующие версии))».

Данную политику необходимо включить в разделе «Computer Configuration (Конфигурация компьютера)» → «Windows Settings (Конфигурация Windows)» → «Security Settings (Параметры безопасности)» → «Local Policies (Локальные политики)» → «Security Options (Параметры безопасности)» (см. Рисунок 8).

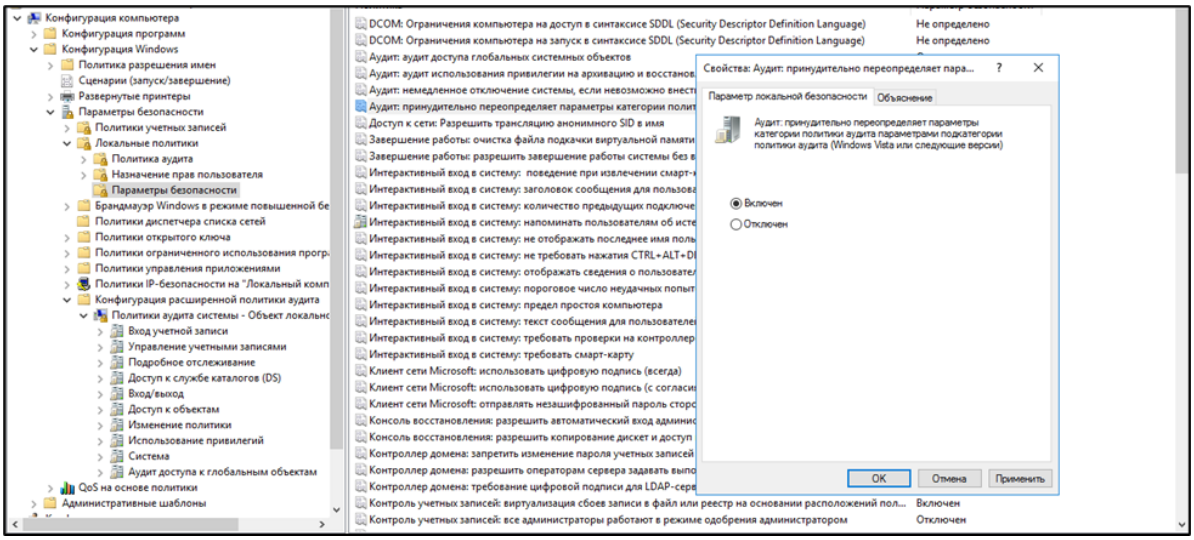

Рисунок 12 - исунок 1. Добавление Audit: Force audit policy subcategory settings

Для активации аудита для контроллеров домена необходимо настроить групповую политику, которая распространяется на контейнер содержащий DC (Контроллеры домена), в соответствии с Таблицей 1. (см. Рисунок 9).

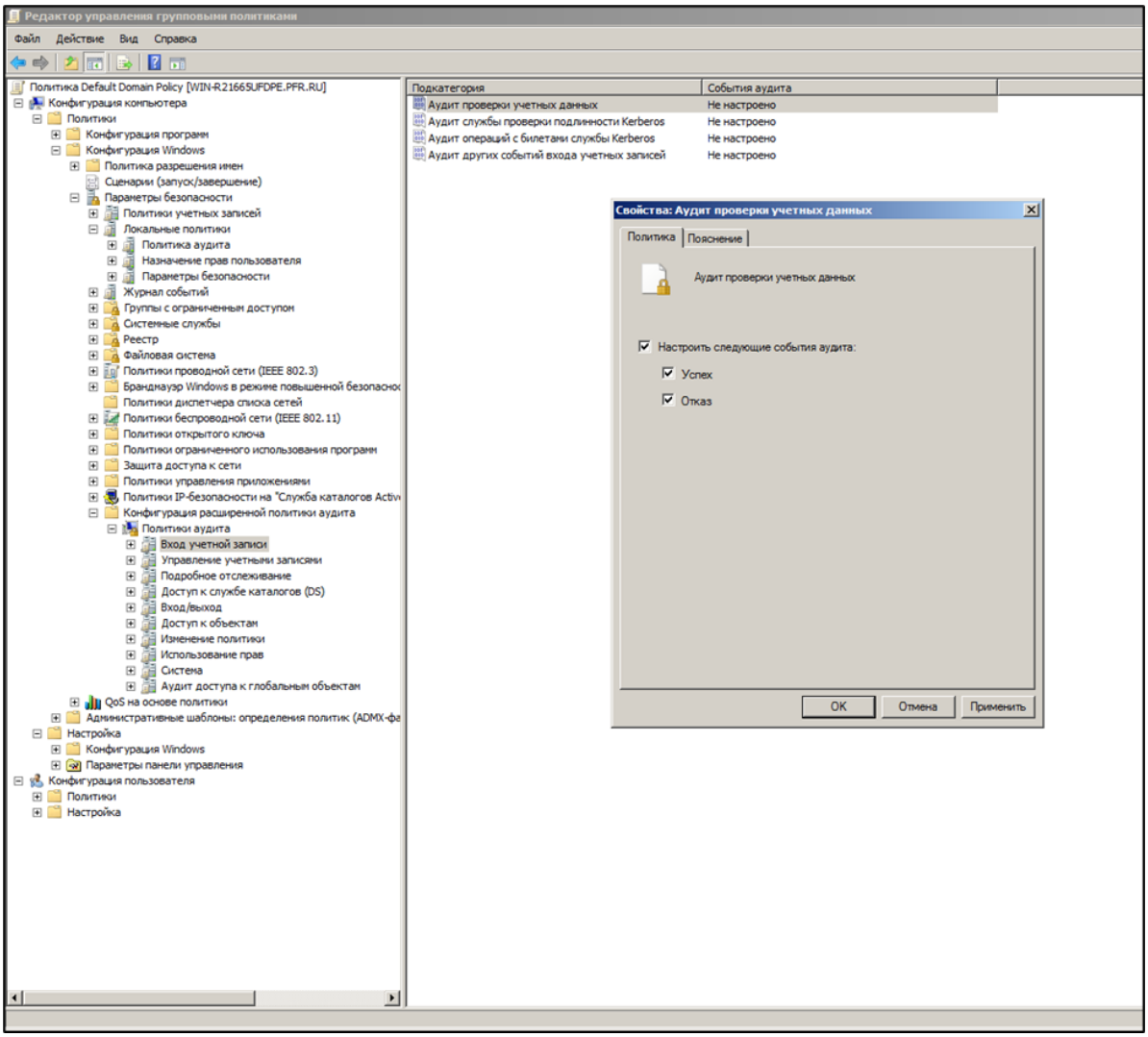

Рисунок 13 - исунок 2. Изменение политик аудита.

Таблица 1. Политики аудита ОС Windows 2008/2012

Политика аудита	**Тип событий**
Аудит проверки учетных данных (Account Logon→Audit Credential Validation)	Успех и Отказ
Аудит службы проверки подлинности Kerberos (Account Logon→Audit Kerberos Authentication Service)	Успех и Отказ
Аудит операций с билетами службы Kerberos (Account Logon→Audit Kerberos Service Ticket Operations)	Успех и Отказ
Аудит других событий входа учетных записей (Account Logon→Audit Other Account Logon Events)	Успех и Отказ
Аудит управления группами приложений(Account Management→Audit Application Group Management)	Успех и Отказ
Аудит управления учетными записями компьютеров (Account Management→Audit Computer Account Management)	Успех и Отказ
Аудит управления группами распространения (Account Management→Audit Distribution Group Management)	Успех и Отказ
Аудит других событий управления учетными записями (Account Management→Audit Other Account Management Events)	Успех и Отказ
Аудит управления группами безопасности (Account Management→Audit Security Group Management)	Успех и Отказ
Аудит управления учетными записями (Account Management→Audit User Account Management)	Успех и Отказ
Аудит активности DPAPI(Detailed Tracking→Audit DPAPI Activity)	Не фиксируются
Аудит создания процессов (Detailed Tracking→Audit Process Creation)	Успех и Отказ
Аудит завершения процессов (Detailed Tracking→Audit Process Termination)	Успех и Отказ
Аудит событий RPC (Detailed Tracking→Audit RPC Events)	Не фиксируются
Аудит подробной репликации службы каталогов (DS Access→Audit Detailed Directory Service Replication)	Не фиксируются
Аудит доступа к службе каталогов (DS Access→Audit Directory Service Access)	Успех и Отказ
Аудит изменения службы каталогов (DS Access→Audit Directory Service Changes)	Успех и Отказ

Политика аудита	Тип событий
Аудит репликации службы каталогов (DS Access→Audit Directory Service Replication)	Не фиксируются
Аудит блокировки учетных записей (Logon/Logoff→Audit Account Lockout)	Успех и Отказ
Аудит расширенного режима IPsec (Logon/Logoff→Audit IPsec Extended Mode)	Не фиксируются
Аудит основного режима IPsec (Logon/Logoff→Audit IPsec Main Mode)	Не фиксируются
Аудит быстрого режима IPsec (Logon/Logoff→Audit IPsec Quick Mode)	Не фиксируются
Аудит выхода из системы (Logon/Logoff→Audit Logoff)	Успех
Аудит входа в систему (Logon/Logoff→Audit Logon)	Успех и Отказ
Аудит сервера политики сети (Logon/Logoff→Audit Network Policy Server)	Не фиксируются
Аудит других событий входа/выхода (Logon/Logoff→Audit Other Logon/Logoff Events)	Успех и Отказ
Аудит специального входа (Logon/Logoff→Audit Special Logon)	Успех и Отказ
Аудит событий, создаваемых приложениями(Object Access→ Audit Application Generated)	Не фиксируются
Аудит сведений об общем файловом ресурсе (Object Access→ Audit Detailed File Share)	Не фиксируются
Аудит общего файлового ресурса (Object Access→ Audit File Share)	Успех и Отказ
Аудит файловой системы (Object Access→ Audit File System)	Успех и Отказ
Аудит подключения платформы фильтрации (Object Access→ Audit Filtering Platform Connection)	Не фиксируются
Аудит отбрасывания пакетов платформой фильтрации (Object Access→ Audit Filtering Platform Packet Drop)	Не фиксируются
Аудит работы с дискрипторами(Object Access→ Audit Handle Manipulation)	Не фиксируются
Аудит объектов ядра (Object Access→ Audit Kernel Object)	Не фиксируются

Политика аудита	Тип событий
Аудит других событий доступа к объектам(Object Access→ Audit Other Object Access Events)	Не фиксируются
Аудит реестра (Object Access → Audit Registry)	Успех и Отказ
Аудит диспетчера учетных записей безопасности (Object Access → Audit SAM)	Не фиксируются
Аудит изменения политики аудита (Policy Change→ Audit Policy Change)	Успех и отказ
Аудит изменения политики проверки подлинности (Policy Change→Audit Audit Authorization Policy Change)	Успех и Отказ
Аудит изменения политики авторизации (Policy Change→Audit Authorization Policy Change)	Успех и Отказ
Аудит изменения политики платформы фильтрации (Policy Change→Audit Filtering Platform Policy Change)	Не фиксируются
Аудит изменения политики на уровне правил MPSSVC (Policy Change→Audit MPSSVC Rule-Level Policy Change)	Успех и Отказ
Аудит других событий изменения политики (Policy Change→Audit Other Policy Change Events)	Успех и Отказ
Аудит использования привилегий, затрагивающих конфиденциальные данные (Privilege Use→Audit Sensitive Privilege Use)	Успех и Отказ
Аудит использования привилегий, не затрагивающих конфиденциальные данные (Privilege Use→Audit Non-Sensitive Privilege Use)	Успех и Отказ
Аудит драйвера IPsec (System→Audit IPsec Driver)	Не фиксируются
Аудит других системных событий (System→Audit Other System Events)	Не фиксируются
Аудит изменения состояния безопасности (System→Audit Security State Change)	Успех и Отказ
Аудит расширения системы безопасности (System→Audit Security System Extension)	Успех и Отказ
Аудит целостности системы (System→Audit System Integrity)	Успех и Отказ

5. Решения Network Security

5.1. Межсетевой экран Cisco ASA

5.1.1. Настройка источника

1. Подключиться к консоли устройства;
2. Для включения журналирования и экспорта событий с устройства, введите команды:

```
(config)# logging enable
```

```
(config)# logging host <имя интерфейса> <IP-адрес коллектора>
```

```
(config)# logging trap <уровень логирования> (указать один из уровней важности событий: alerts, critical, debugging, emergencies, errors, informational, notifications, warnings)
```

```
(config)# logging console <уровень логирования> (указать один из уровней важности событий: alerts, critical, debugging, emergencies, errors, informational, notifications, warnings)
```

```
(config)# logging asdm <уровень логирования> (указать один из уровней важности событий: alerts, critical, debugging, emergencies, errors, informational, notifications, warnings)
```

```
(config)# logging device-id ipaddress <id устройства>
```

```
(config)# logging timestamp
```

5.1.2. Включение источника на Платформе

Для информации! Включение источника в Платформе представлено в разделе **Управление источниками в Платформе, Включение/выключение поддерживаемых источников и их синхронизация**

1. Зайти в веб-консоль Платформы, перейти в раздел «Источники» - «Управление источниками»;
2. Найти в списке доступных источников (Cisco-ASA) и включить его;
3. Кликнуть на кнопку «Синхронизировать».

5.1.3. Настройка коллектора событий

Для информации! Пример конфигурационного файла с настройкой данного источника представлен в разделе **Пример конфигурационного файла лог-коллектора**. Более подробная информация о настройках лог-коллектора представлена в разделе **Руководство по настройке лог-коллектора**

1. В конфигурационный файл лог-коллектора (config.yaml) необходимо добавить input компонента UDP.

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Компонент UDP**

Основные параметры, которые необходимо указать:

```
host: "<ip адрес лог-коллектора>" (адрес, на котором запущен коллектор)
```

```
port: <порт для приема соединений> (порт, на который будут приниматься события, если при настройке источника оставили стандартный - 2520)
```

2. После настройки компонента сбора событий (input) необходимо настроить компонент отправки событий (output).

В качестве компонента отправки событий для данного источника предусмотрено использование отправки по протоколу TCP.

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Компонент отправки событий по протоколу TCP**

Основные параметры, которые необходимо указать:

`target_host: <"ip адрес или имя удаленного узла"> (адрес Платформы)`
`port: <"порт"> (стандартный порт для данного источника 2520)`

3. Далее необходимо включить компоненты сбора (collectors) и отправки (senders).

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Включение компонентов**

Основные параметры, которые нужно указать при включении компонентов сбора:

`collectors:`

`udp_reciever:`

`- <<: *<"id компонента сбора"> (ID компонента сбора, который указывали при объявлении компонента сбора)`

Основные параметры, которые нужно указать при включении компонентов отправки:

`senders:`

`tcp:`

`- <<: *<"id компонента отправки"> (ID компонента отправки, который указывали при объявлении компонента отправки)`

4. После чего необходимо произвести настройку маршрутизации событий.

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Маршрутизация событий**

Основные параметры, которые нужно указать при настройке маршрута:

`route_1: &route_1`

`collector_id:`

`- <"id компонента сбора">`

`sender_id:`

`- <"id компонента отправки">`

5. После нужно включить маршрут в разделе routers. Пример включения маршрута:

`routers:`

`- <<: *<название маршрута> (например - <<: *route_1)`

6. Системы антивирусной защиты

6.1. Kaspersky Security Center. Microsoft SQL Server.

6.1.1. Настройка источника

1. Создание учетной записи для сбора событий.

Для сбора событий с базы данных Kaspersky Security Center необходимо создать учетную запись с членством в роли db_datareader для базы KAV.

Процесс создания учетной записи приведен в Приложении 1. Создание учетной записи Microsoft SQL Server.

2. При использовании межсетевого экрана на узле, необходимо сделать правило для входящих соединений.

6.1.2. Включение источника на Платформе

Для информации! Включение источника в Платформе представлено в разделе **Управление источниками в Платформе, Включение/выключение поддерживаемых источников и их синхронизация**

1. Зайти в веб-консоль Платформы, перейти в раздел «Источники», «Управление источниками»;
2. Найти в списке доступных источников «Kaspersky-SecurityCenter-db» и включить его;
3. Кликнуть на кнопку «Синхронизировать».

6.1.3. Настройка коллектора событий

Для информации! Пример конфигурационного файла с настройкой данного источника представлен в разделе **Пример конфигурационного файла лог-коллектора**. Более подробная информация о настройках лог-коллектора представлена в разделе **Руководство по настройке лог-коллектора**

1. В конфигурационный файл лог-коллектора (config.yaml) необходимо добавить input компонента ODBC.

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Компонент ODBC**

Основные параметры, которые необходимо указать:

`connection_string: "Driver={ODBC Driver 17 for SQL Server};Server=<ip-адрес>;Port=1433;Database=KAV;UID=<username>;PWD=<password>;"`

Строка с sql запросом к базе представлена в Приложении 2. SQL запрос для KSC.

2. После настройки компонента сбора событий (input) - необходимо настроить компонент отправки событий (output).

В качестве компонента отправки событий для данного источника предусмотрено использование отправки по протоколу TCP.

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Компонент отправки событий по протоколу TCP**

Основные параметры, которые необходимо указать:

`target_host: <"ip адрес или имя удаленного узла"> (адрес Платформы)`

`port: <"порт"> (стандартный порт для данного источника 2604)`

3. Далее необходимо включить компоненты сбора (collectors) и отправки (senders).

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Включение компонентов**

Основные параметры, которые нужно указать при включении компонентов сбора:

`collectors:`

`odbc:`

`- <<: *<"id компонента сбора"> (ID компонента сбора, который указывали при объявлении компонента сбора)`

Основные параметры, которые нужно указать при включении компонентов отправки:

`senders:`

`tcp:`

`- <<: *<"id компонента отправки"> (ID компонента отправки, который указывали при объявлении компонента отправки)`

4. После чего необходимо произвести настройку маршрутизации событий.

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Маршрутизация событий**

Основные параметры, которые нужно указать при настройке маршрута:

`route_1: &route_1`

`collector_id:`

`- <"id компонента сбора">`

`sender_id:`

`- <"id компонента отправки">`

5. После нужно включить маршрут в разделе routers. Пример включения маршрута:

`routers:`

`- <<: *<название маршрута> (например - <<: *route_1)`

6.1.4. Приложение 1. Создание учетной записи Microsoft SQL Server.

Создание имени входа на сервер.

Настройку сервера необходимо выполнять от имени учетной записи, имеющей права локального администратора ОС Windows. Для создания данной учётной записи необходимо выполнить следующие действия:

1. В меню Пуск открыть среду разработки MS SQL Management Studio (Диспетчер конфигурации SQL Server).
2. В окне Connect to Server (Соединение с сервером) подключится к экземпляру необходимой базы данных (БД) с правами администратора sa (см. Рисунок 14).

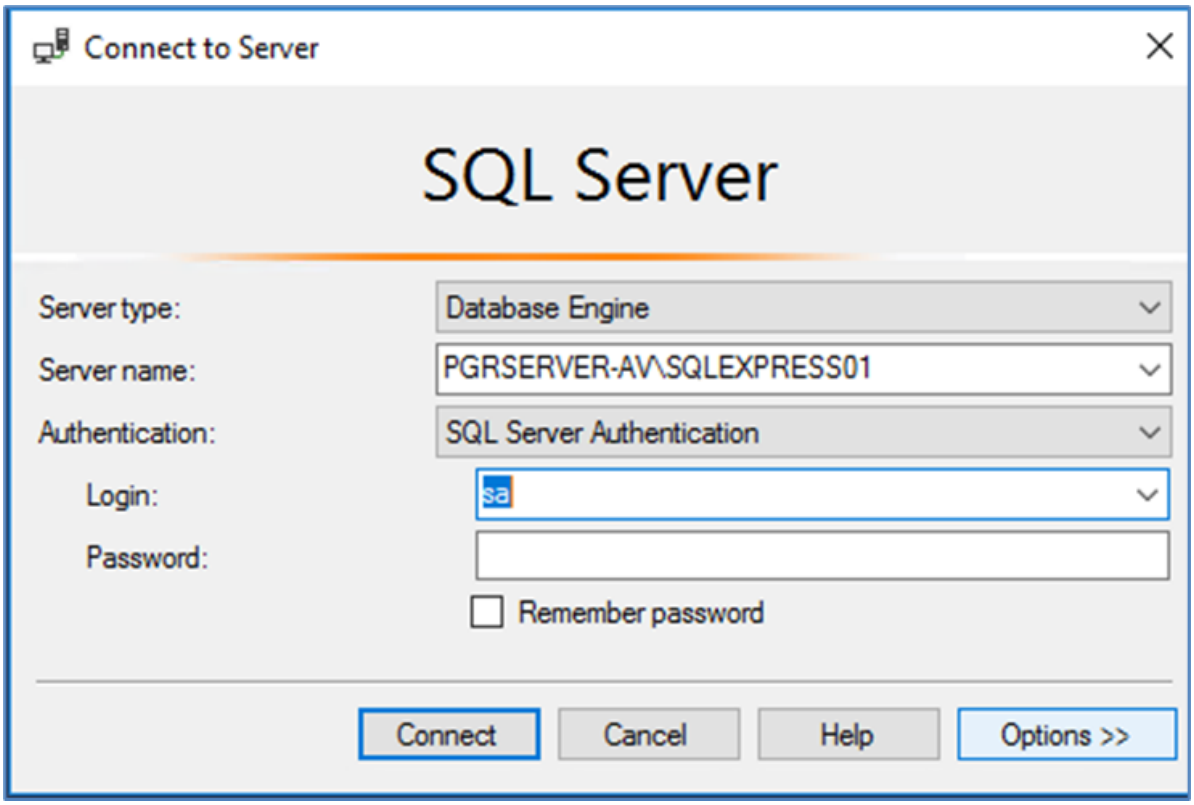

Рисунок 14 - Подключение к экземпляру БД

3. Подключится к экземпляру БД. Для предоставления доступа к экземпляру БД выполнить следующие действия:
    - В окне Object Explorer (Обозреватель объектов) выбранного экземпляра БД (SQLEXPRESS) открыть контекстное меню раздела Logins (Имена для входа):

    Security → Logins (Безопасность → Имена для входа)

    - В контекстном меню выбрать команду New Login (Создать имя для входа) (см. Рисунок 15).

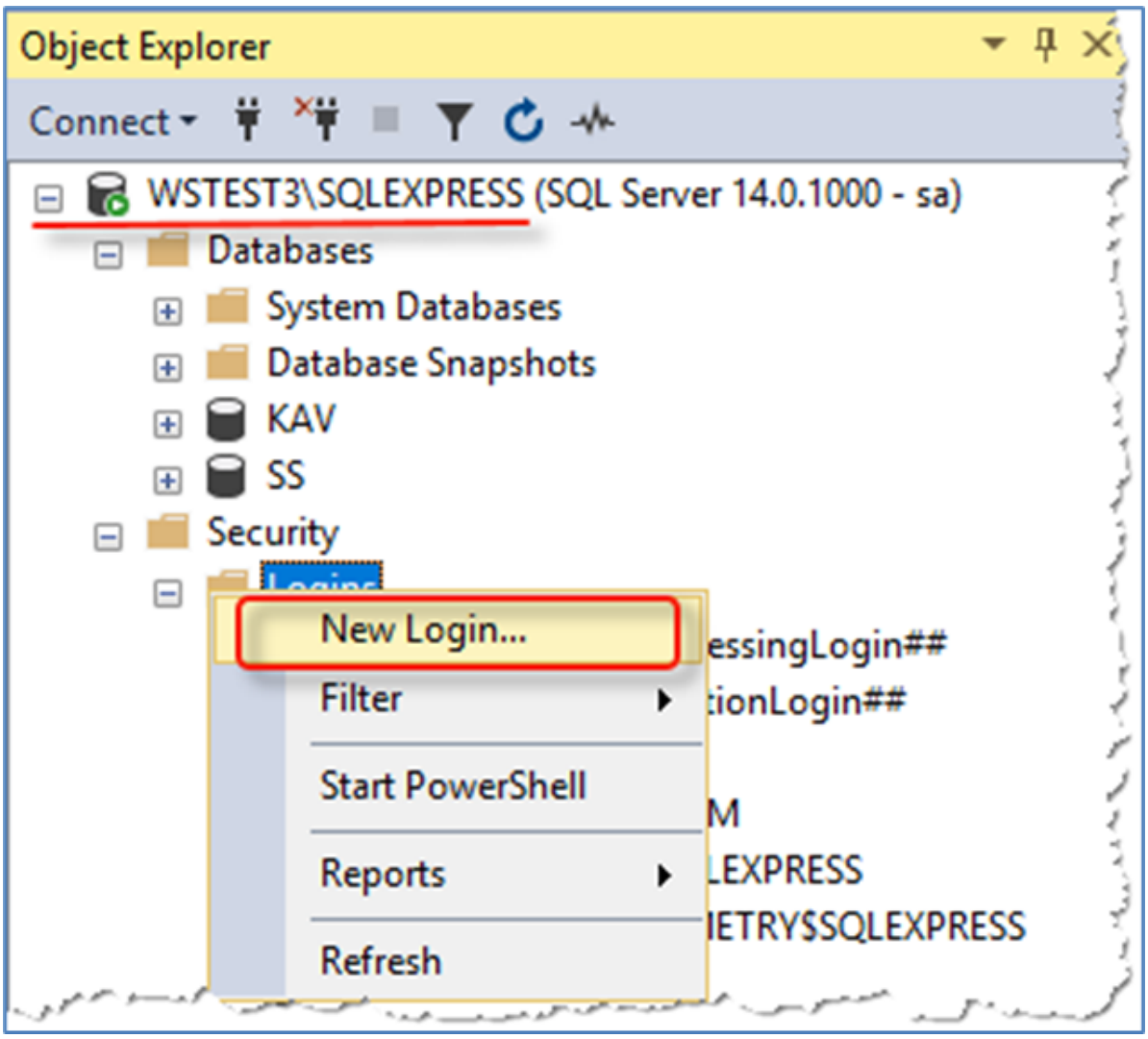

Рисунок 15 - Дерево каталогов экземпляра БД

- В открывшемся окне Login--New (Создание имени для входа) в разделе General (Общие) выполнить следующие настройки (см. Рисунок 16):
- Ввести имя пользователя (*radaruser*) в поле Login Name (Имя для входа).
- Установить пароль в полях Password и Confirm Password (Пароль, Подтверждение пароля).
- При необходимости выставить настройки в пунктах:
  - Enforce password policy (Требовать использование политики паролей);
  - Enforce password expiration (Задать срок окончания действия пароля).
- Выбрать режим SQL Server authentication (Проверка подлинности SQL Server).
- Выбрать *KAV* в качестве БД по умолчанию в раскрывающемся списке Default Database (База данных по умолчанию).

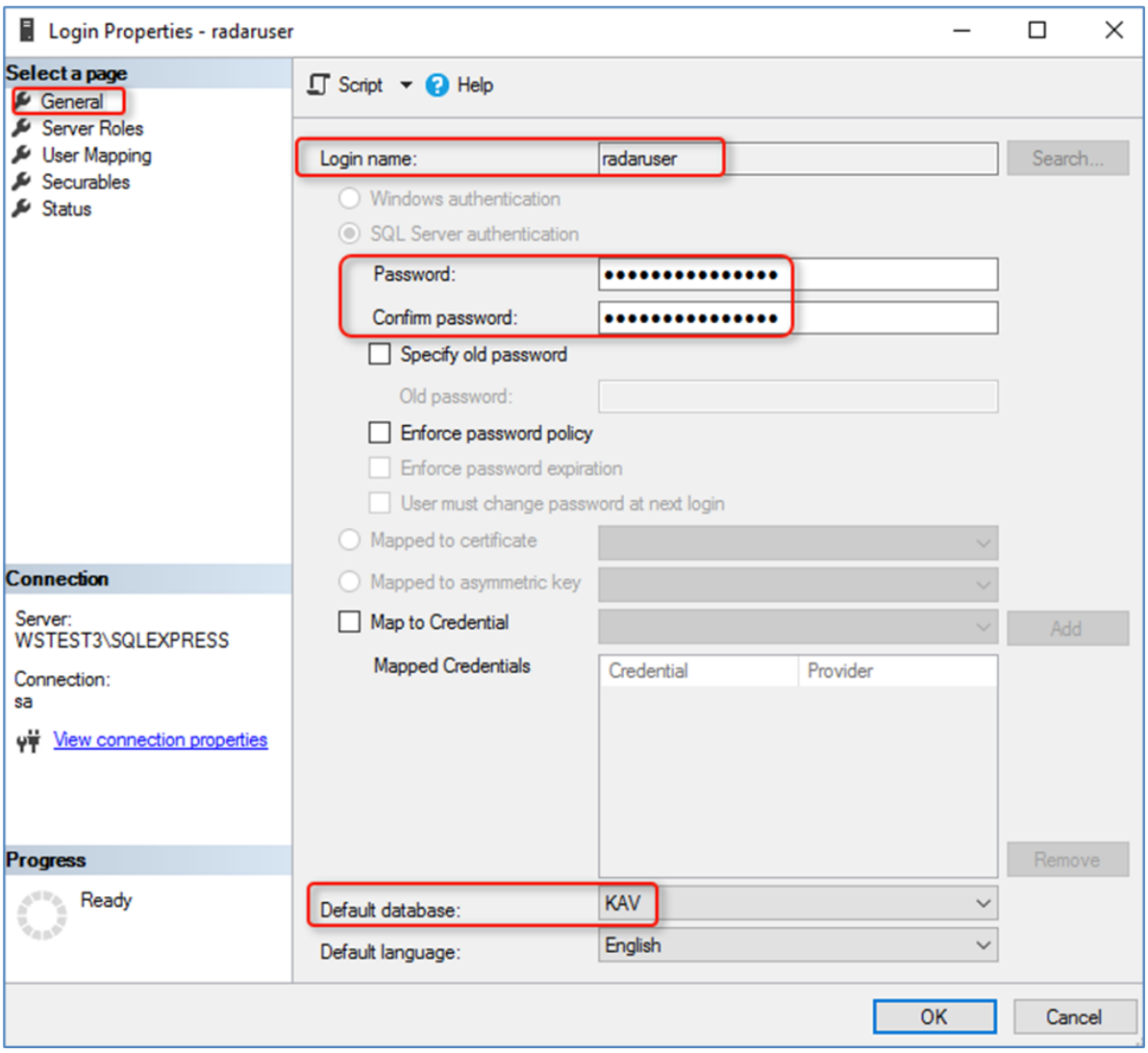

Рисунок 16 - Создание нового пользователя экземпляра БД

- В разделе Server Roles (Роли сервера) проверить что пользователю предоставлена роль *public* (см. Рисунок 17).
    - Если она не предоставлена, то предоставить пользователю роль *public*.

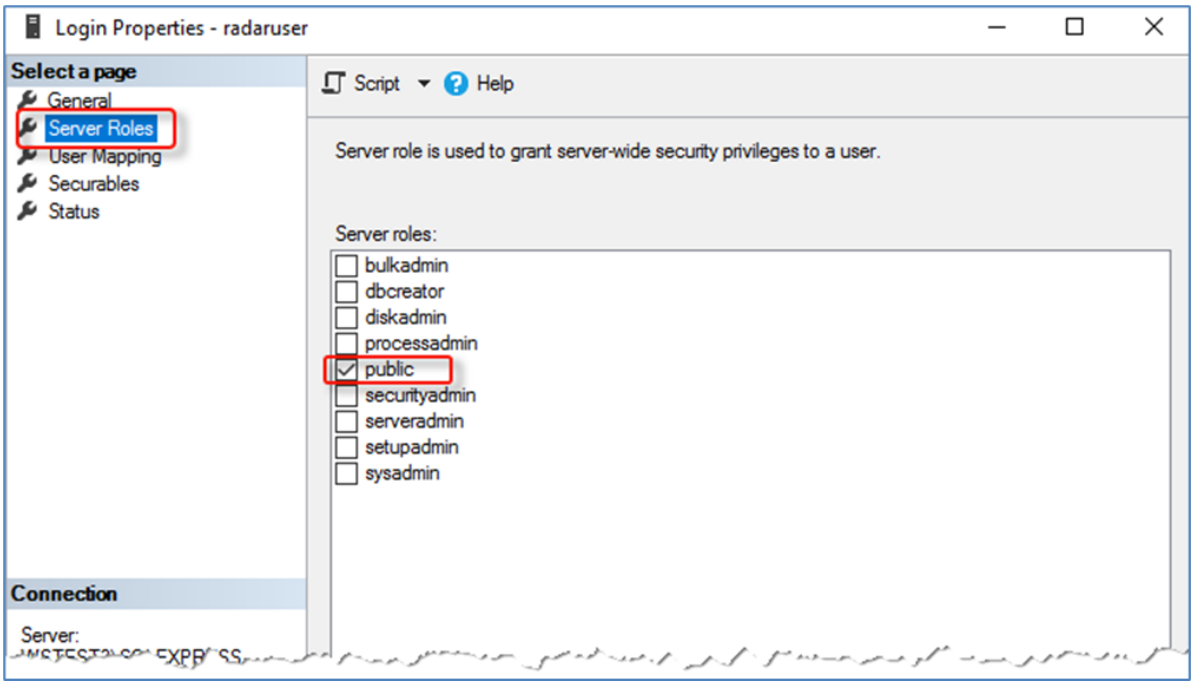

Рисунок 17 - Предоставление роли для создаваемого пользователя

- В разделе User Mapping (Сопоставления пользователей) для созданного пользователя (radaruser) выполнить следующие настройки:
- В поле User mapped to this login: (Пользователи, сопоставленные с этим именем для входа:) предоставить разрешение на подключение и чтение к БД *KAV*.
- В поле Database role membership for: <имя БД> (Членство в роли базы данных для: <имя БД>) установить для выбранной БД роль *db_datareader* (см. Рисунок 18).

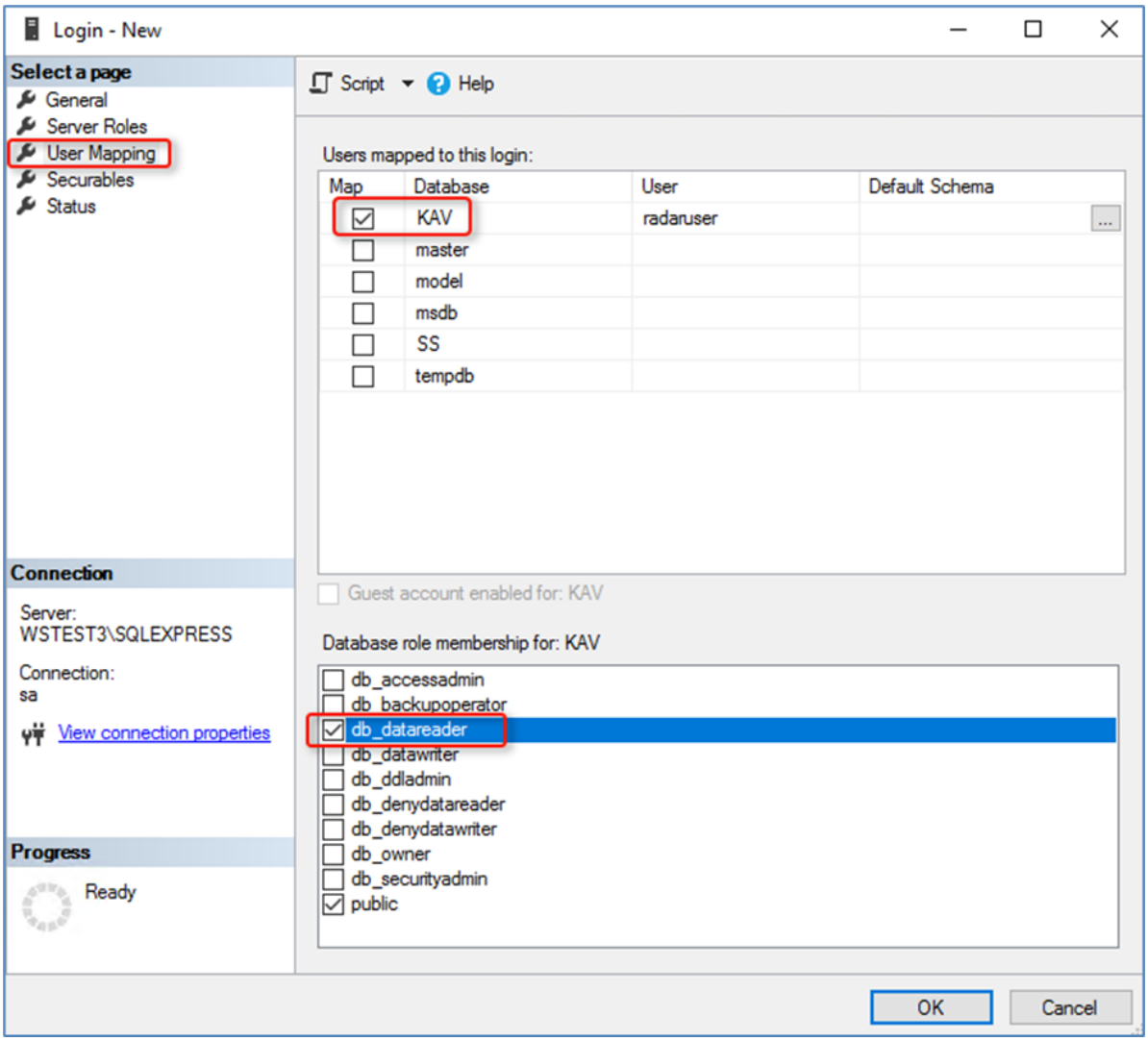

Рисунок 18 - Настройка прав доступа к БД KAV

- В разделе Securables (Защищаемые объекты) для созданного пользователя (radaruser) установить для выбранного сервера СУБД следующие разрешения в области Permission for: <имя сервера СУБД> (Разрешения для: <имя сервера СУБД>):
- *Connect SQL (подключение SQL)* (см. Рисунок 19).

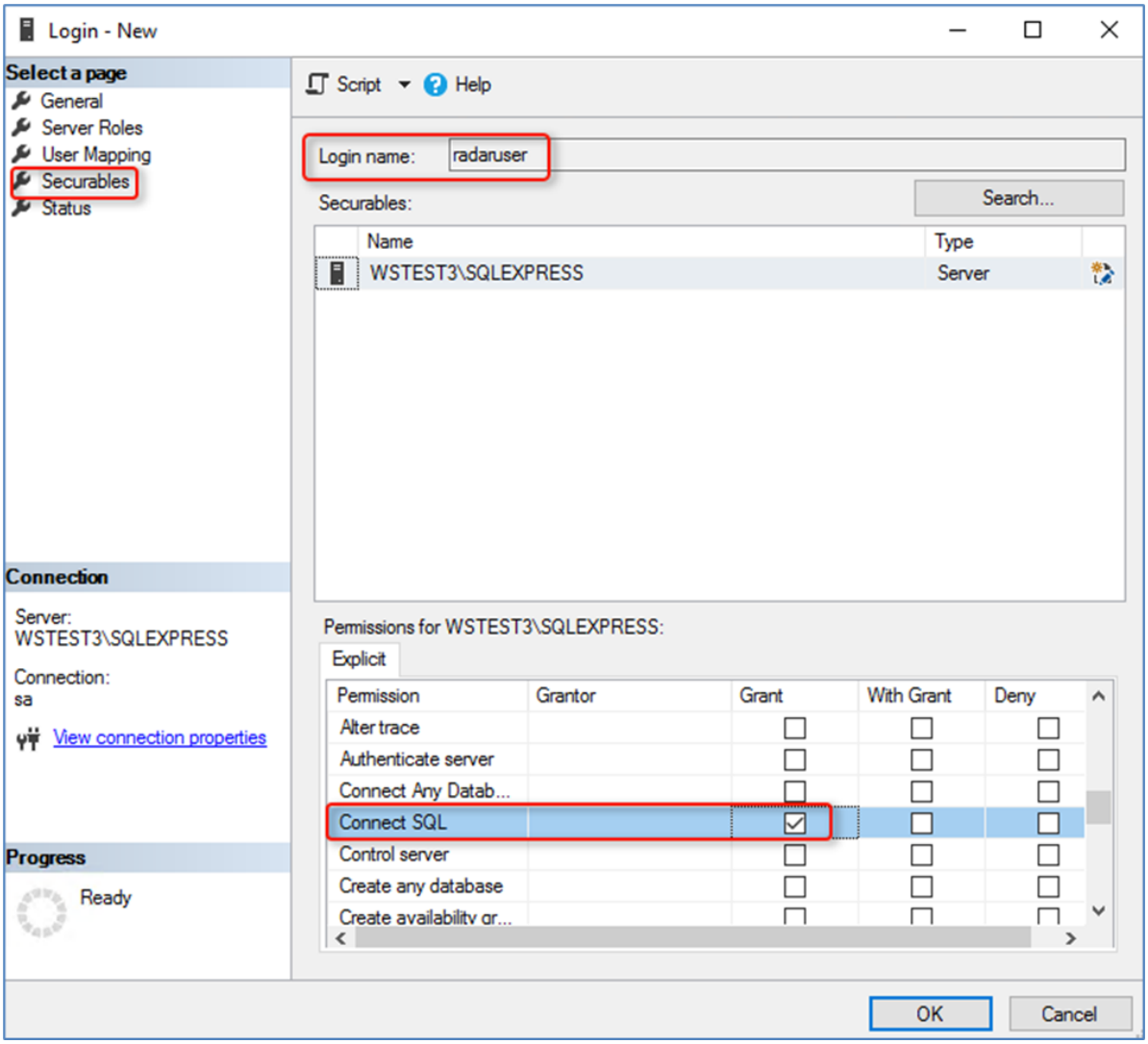

Рисунок 19 - Установка разрешения на подключение к БД

- Для сохранения введенных настроек для подключения к экземпляру БД нажать кнопку OK.

Создание пользователя в БД KAV. Для предоставления доступа к БД KAV выполнить следующие действия:

- В окне Object Explorer (Обозреватель объектов) выбранного экземпляра БД (SQLEXPRESS) выбрать раздел (см. Рисунок 20):

```
Database → <Имя БД> → Security → Users
(База данных → <Имя БД> → Безопасность → Пользователи).
```

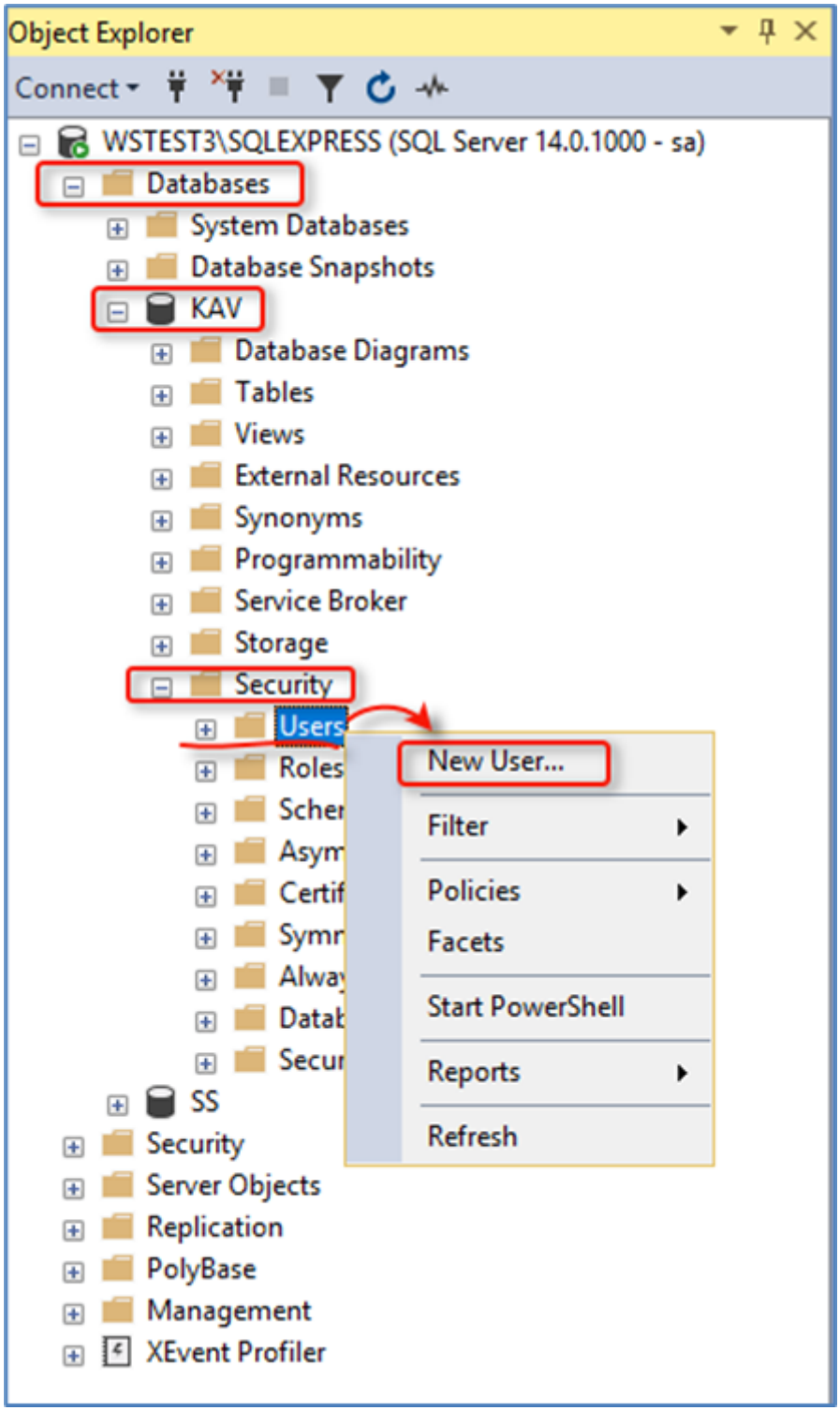

Рисунок 20 - Функция создания пользователя в БД KAV

- Открыть контекстное меню раздела Users (Пользователи) и выбрать функцию New User (Создать пользователя) (см. Рисунок 20).
- В открывшемся окне Database User - New (Пользователь базы данных - Создать) в разделе General (Общие) установить следующие параметры (см. Рисунок 21):
  - в поле *User name (Имя пользователя)* установить имя пользователя (dbuser);
  - в поле *Login name (Имя для входа)* указать созданного выше (см. шаг 3) пользователя экземпляра БД (*radaruser*).

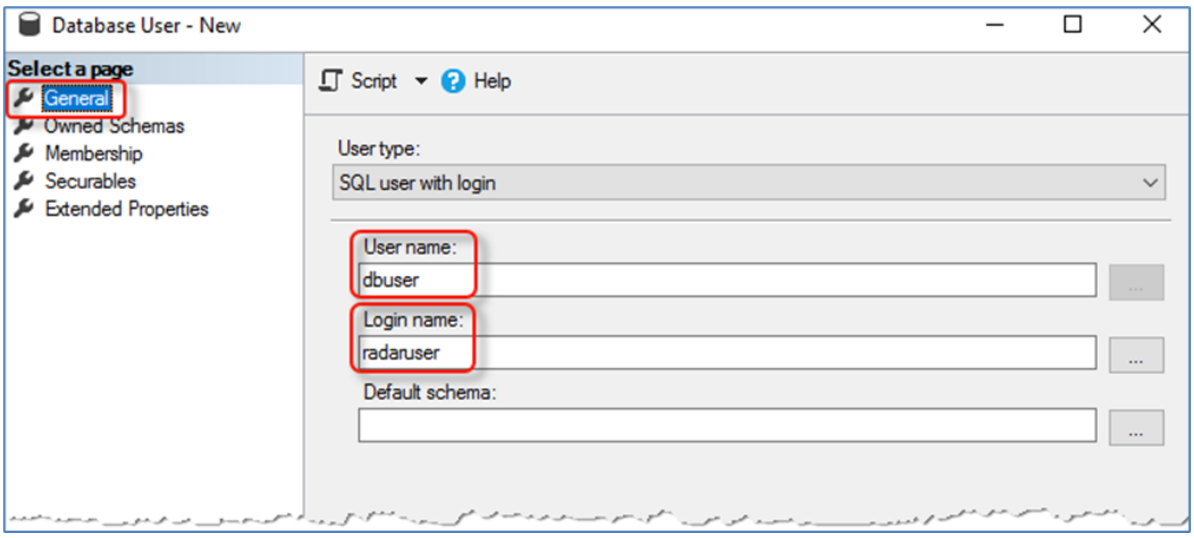

Рисунок 21 - Регистрация пользователя в БД KAV

- В разделе Membership (Членство) установить для пользователя роль *db_datareader* (см. Рисунок 22).

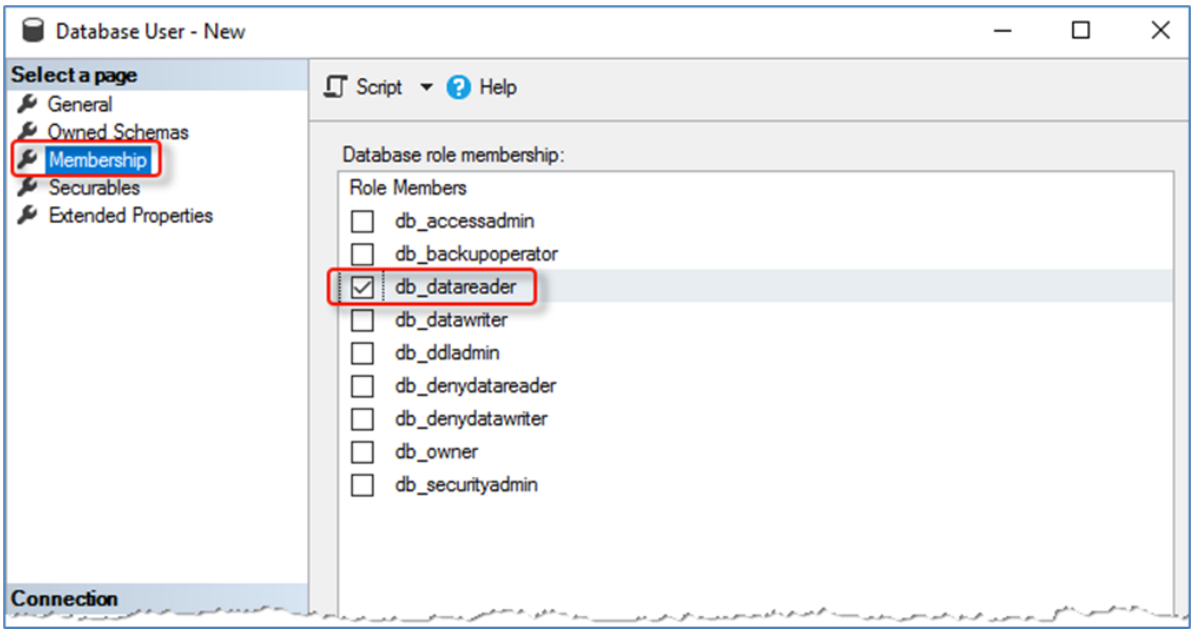

Рисунок 22 - Назначение роли

- Для сохранения всех введенных настроек при создании пользователя в БД KAV нажать кнопку OK.

Предоставление удаленного сетевого доступа. Для удаленного доступа к данным, необходимо настроить доступность для выбранного экземпляра БД (*SQLEXPRESS*):

- В меню Пуск необходимо запустить SQL Server Configuration Manager (Диспетчер конфигурации SQL Server).
- В панели диспетчера конфигурации выбрать службу (см. Рисунок 23):

  SQL Server Network Configuration → Protocols for SQLEXPRESS

  (Сетевая конфигурация SQL Server → Протоколы для SQLEXPRESS).
- В открывшемся справа списке протоколов выбрать протокол TCP/IP и в контекстном меню протокола перевести подключение по данному протоколу в режим «Включено», установив статус *Enabled (Включено)* (см. Рисунок 23).

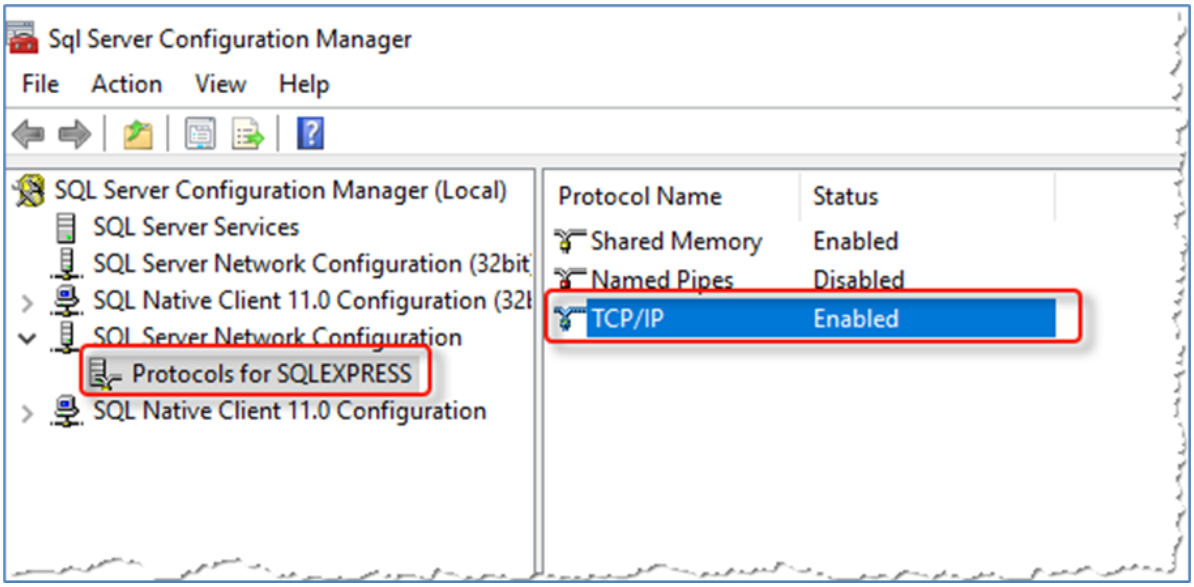

Рисунок 23 - 10. Подключение по протоколу TCP/IP

- В контекстном меню протокола TCP/IP выбрать функцию Properties (Свойства).
- В открывшемся окне TCP/IP Properties (Свойства TCP/IP) на вкладке IP Adresses (IP-адреса) выбрать блок параметров *IPAll* и ввести значение порта в поле TCP Port. Например: 1433 (см. Рисунок 24).
- Нажать кнопку OK для сохранения настроек доступа по протоколу TCP/IP.

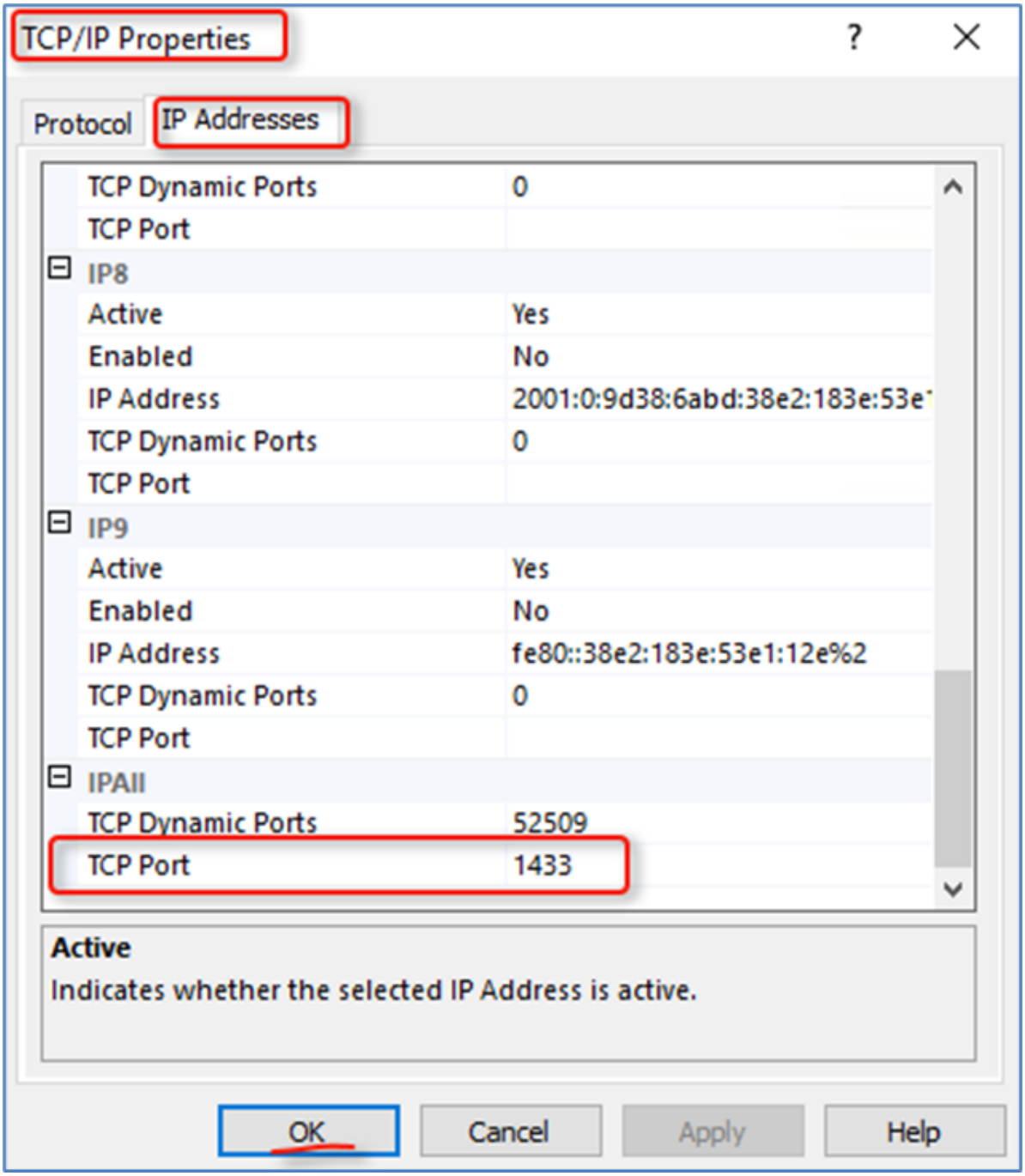

Рисунок 24 - Пример настройки протокола для удаленного доступа к БД

- Для применения сетевых настроек необходимо перезапустить службу MS SQL Server:
  - В меню Пуск выбрать раздел Servise (Службы).
  - В открывшемся окне Службы (Службы) выбрать службу SQL Server с запущенным экземпляром БД (SQLEXPRESS).
  - Выбрать функцию Restart the servise (Перезапустить службу) (см. Рисунок 12).

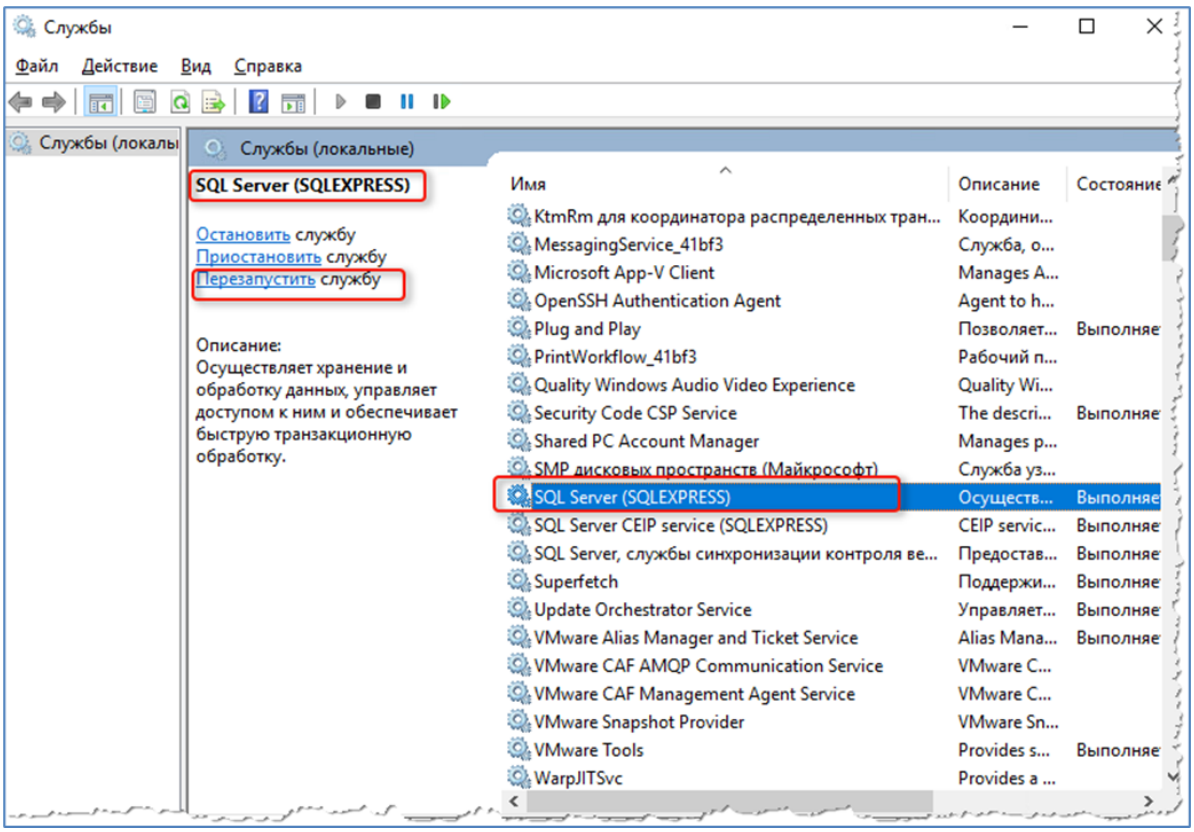

Рисунок 25 - Перезапуск службы MS SQL Server

6.1.5. Приложение 2. SQL запрос для KSC.

```
sql: >
    SELECT
      events.event_id AS event_id,
      events.nHostId AS host_id,
      events.severity AS severity,
      events.group_name AS group_name, event_type,
      events.event_type_display_name AS event_name,
      rise_time AS event_time,
      events.descr AS description,
      events.task_display_name AS task_name,
      events.task_id AS task_id,
      events.product_displ_version AS product_version,
      events.par1,
      events.par2,
      events.par3,
      events.par4,
      events.par5,
      events.par6,
      events.par7,
      events.par8,
      events.product_name,
      hosts_view.strDisplayName AS hostname,
      dnsdomains.strName AS domain,
      fqdns.wstrfqdn AS fqdn,
      CAST(((hosts.nIpAddress / 16777216) & 255) AS varchar(4)) + '.' +
      CAST(((hosts.nIpAddress / 65536) & 255) AS varchar(4)) + '.' +
      CAST(((hosts.nIpAddress / 256) & 255) AS varchar(4)) + '.' +
```

```
      CAST(((hosts.nIpAddress) & 255) AS varchar(4)) AS ip_address,
      hosts_view.nPlatformType AS platform_id,
      hosts_view.tmLastInfoUpdate AS last_update,
      hosts_view.nVirusCount AS virus_count
    FROM KAV.dbo.ev_event AS events
    JOIN KAV.dbo.Hosts AS hosts ON hosts.nId = events.nHostId
    JOIN KAV.dbo.v_hosts AS hosts_view ON hosts_view.nId = hosts.nId
    JOIN KAV.dbo.v_hst_fqdns AS fqdns ON fqdns.nId = hosts.nId
    RIGHT JOIN KAV.dbo.DnsDomains AS dnsdomains ON dnsdomains.nId =
hosts.nDnsDomain
    WHERE events.event_type IN (
      'FSEE_AKPLUGIN_CRITICAL_PATCHES_AVAILABLE',
      'FSEE_AKPLUGIN_PEP_APPLICATION_AUDIT_DENIED',
      'GNRL_EV_APP_LAUNCH_TESTED_DENIED',
      'GNRL_EV_APPLICATION_LAUNCH_DENIED',
      'GNRL_EV_ATTACK_DETECTED',
      'GNRL_EV_DEVCTRL_DEV_PLUGGED',
      'GNRL_EV_OBJECT_BLOCKED',
      'GNRL_EV_OBJECT_CURED',
      'GNRL_EV_OBJECT_DELETED',
      'GNRL_EV_OBJECT_NOTCURED',
      'GNRL_EV_OBJECT_QUARANTINED',
      'GNRL_EV_PTOTECTION_LEVEL_CHANGED',
      'GNRL_EV_SUSPICIOUS_OBJECT_FOUND',
      'GNRL_EV_VIRUS_FOUND',
      'GNRL_EV_VIRUS_OUTBREAK',
      'KLAUD_EV_ADMGROUP_CHANGED',
      'KLAUD_EV_SERVERCONNECT',
      'KLNAG_EV_INV_APP_INSTALLED',
      'KLNAG_EV_INV_APP_UNINSTALLED',
      'KLNAG_EV_INV_CMPTR_APP_INSTALLED',
      'KLPRCI_TaskState',
      'KLSRV_EVENT_HOSTS_CONFLICT',
      'KLSRV_EVENT_HOSTS_NEW_DETECTED',
      'KLSRV_HOST_STATUS_CRITICAL',
      'KLSRV_HOST_STATUS_WARNING',
      'KLSRV_SEAMLESS_UPDATE_REGISTERED',
      'KLSRV_UPD_BASES_UPDATED',
      '000000d1',
      '000000d3',
      '000000d4',
      '000000d5',
      '000000d6',
      '000000dd',
      '000000de',
      '000000df',
      '0000012f',
      '0000014d',
      '0000014e',
      '0000014f',
      '00000192',
      '00000193',
      '000000cf'
    )
    AND event_id > ?;
```

7. Сетевые устройства.

7.1. Cisco IOS. System logging.

7.1.1. Настройка источника

1. Подключиться к консоли устройства;
2. Для включения логирования всех попыток подключения к устройству, введите команды:

```
(config)# service timestamps log datetime localtime show-timezone year
(config)# logging userinfo
(config)# login on-failure log
(config)# login on-success log
```

3. Для включения логирования изменений конфигурации, введите команды:

```
(config)# archive
(config-archive)# log config
(config-archive-log-cfg)# logging enable
(config-archive-log-cfg)# notify syslog
(config-archive-log-cfg)# hidekeys
```

4. Для отправки событий на коллектор, введите команды:

```
(config)# logging facility local5
(config)# logging host <IP-адрес коллектора> transport tcp port <порт коллектора> (порт по умолчанию 2523)
```

7.1.2. Включение источника на Платформе

Для информации! Включение источника в Платформе представлено в разделе **Управление источниками в Платформе, Включение/выключение поддерживаемых источников и их синхронизация**

1. Зайти в веб-консоль Платформы, перейти в раздел «Источники», «Управление источниками»;
2. Найти в списке доступных источников (Cisco-IOSswitch) и включить его;
3. Кликнуть на кнопку «Синхронизировать».

7.1.3. Настройка коллектора событий

Для информации! Пример конфигурационного файла с настройкой данного источника представлен в разделе **Пример конфигурационного файла лог-коллектора**. Более подробная информация о настройках лог-коллектора представлена в разделе **Руководство по настройке лог-коллектора**

1. В конфигурационный файл лог-коллектора (config.yaml) необходимо добавить input компонента TCP.

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Компонент TCP**

Основные параметры, которые необходимо указать:

`target_host: <"ip адрес или имя удаленного узла"> (адрес Платформы)`

`port: <"порт"> (стандартный порт для данного источника 2523)`

2. После настройки компонента сбора событий (input) - необходимо настроить компонент отправки событий (output).

В качестве компонента отправки событий для данного источника предусмотрено использование отправки по протоколу TCP.

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Компонент отправки событий по протоколу TCP**

Основные параметры, которые необходимо указать:

`target_host: <"ip адрес или имя удаленного узла"> (адрес Платформы)`

`port: <"порт"> (стандартный порт для данного источника 2523)`

3. Далее необходимо включить компоненты сбора (collectors) и отправки (senders).

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Включение компонентов**

Основные параметры, которые нужно указать при включении компонентов сбора:

`collectors:`

`tcp_receiver:`

`- <<: *<"id компонента сбора"> (ID компонента сбора, который указывали при объявлении компонента сбора)`

Основные параметры, которые нужно указать при включении компонентов отправки:

`senders:`

`tcp:`

`- <<: *<"id компонента отправки"> (ID компонента отправки, который указывали при объявлении компонента отправки)`

4. После чего необходимо произвести настройку маршрутизации событий.

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Маршрутизация событий**

Основные параметры, которые нужно указать при настройке маршрута:

`route_1: &route_1`

`collector_id:`

`- <"id компонента сбора">`

`sender_id:`

`- <"id компонента отправки">`

5. После нужно включить маршрут в разделе routers. Пример включения маршрута:

`routers:`

`- <<: *<название маршрута> (например - <<: *route_1)`

7.2. Cisco IOS. Netflow v5.

7.2.1. Настройка источника

1. Подключиться к консоли устройства;
2. Для включения экспорта статистики сетевого трафика по протоколу NetFlow введите команды:

`(config)# ip-flow-export destination <IP-адрес коллектора> <порт коллектора> (по умолчанию 2162)`

`(config)# ip flow-export version 5`

`(config)# interface <интерфейс, с которого необходимо собирать статистику>`

`(config)# ip flow ingress`

`(config)# ip flow egress`

7.2.2. Включение источника на Платформе

Для информации! Включение источника в Платформе представлено в разделе **Управление источниками в Платформе, Включение/выключение поддерживаемых источников и их синхронизация**

1. Зайти в веб-консоль Платформы, перейти в раздел «Источники», «Управление источниками»;
2. Найти в списке доступных источников (Cisco-NetFlow) и включить его;
3. Кликнуть на кнопку «Синхронизировать».

7.2.3. Настройка коллектора событий

Для информации! Пример конфигурационного файла с настройкой данного источника представлен в разделе **Пример конфигурационного файла лог-коллектора**. Более подробная информация о настройках лог-коллектора представлена в разделе **Руководство по настройке лог-коллектора**

1. В конфигурационный файл лог-коллектора (config.yaml) необходимо добавить input компонента NetFlow.

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Компонент NetFlow**

Основные параметры, которые необходимо указать:

`host: "<ip адрес лог-коллектора>" (адрес на котором запущен коллектор)`

`port: <порт для приема соединений> (порт, на который будут приниматься события, если при настройке источника оставили стандартный - 2162)`

2. После настройки компонента сбора событий (input) - необходимо настроить компонент отправки событий (output).

В качестве компонента отправки событий для данного источника предусмотрено использование отправки по протоколу TCP.

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Компонент отправки событий по протоколу TCP**

Основные параметры, которые необходимо указать:

`target_host: <"ip адрес или имя удаленного узла"> (адрес Платформы)`

`port: <"порт"> (стандартный порт для данного источника 2162)`

3. Далее необходимо включить компоненты сбора (collectors) и отправки (senders).

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Включение компонентов**

Основные параметры, которые нужно указать при включении компонентов сбора:

`collectors:`

`nf_receiver:`

`- <<: *<"id компонента сбора"> (ID компонента сбора, который указывали при объявлении компонента сбора)`

Основные параметры, которые нужно указать при включении компонентов отправки:

`senders:`

`tcp:`

`- <<: *<"id компонента отправки"> (ID компонента отправки, который указывали при объявлении компонента отправки)`

4. После чего необходимо произвести настройку маршрутизации событий.

Пример настройки по умолчанию можно найти в документации: **Руководство по настройке лог-коллектора, Маршрутизация событий**

Основные параметры, которые нужно указать при настройке маршрута:

`route_1: &route_1`

`collector_id:`

`- <"id компонента сбора">`

`sender_id:`

`- <"id компонента отправки">`

5. После нужно включить маршрут в разделе routers. Пример включения маршрута:

`routers:`

`- <<: *<название маршрута> (например - <<: *route_1)`

8. Другое

8.1. ОС Windows. Утилита Sysmon

Об утилите

Sysmon (System Monitor) - утилита, которая позволяет получить более полные сведения о событиях Windows.

Ссылка на ресурс Microsoft для подробного изучения:

https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon#overview-of-sysmon-capabilities

Для запуска утилиты необходимо, чтобы на машине, на которой планируется сбор событий, было расположено два файла: файл-установщик с расширением .bat или .exe и файл конфигурации с расширением .xml. Для удобства работы рекомендуется расположить эти файлы в одной папке.

Актуальную версию утилиты можно скачать с официального ресурса Microsoft:

https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

8.1.1. Настройка источника

1. Установите и настройте утилиту Sysmon:

- Нажмите **Пуск+S** на клавиатуре
- Введите в строке поиска **cmd** и нажмите **Enter**
- Перейдите в папку, где лежат файл-установщик и файл конфигурации с помощью команды `cd <directory>`

  Пример: C:\Windows\system32> cd c:\Sysmon
- Установите утилиту Sysmon с помощью команды `sysmon.exe -i <configfile>`

```
Пример: C:\Windows>sysmon.exe -i sysmon.xml
```

2. После успешной установки в **Просмотре событий Windows** (Event Viewer) появится новый журнал (Channel) **Microsoft-Windows-Sysmon/Operational**.

8.1.2. Включение источника на Платформе

Процесс включения источника на Платформе не отличается от включения источника на Платформе для Microsoft Windows

8.1.3. Настройка коллектора событий

Процесс настройки лог-коллектора отличается от настройки коллектора событий для Microsoft Windows только настройкой журналов для сбора событий.

Для отправки событий журнала Sysmon на Платформу необходимо внести изменение в файл конфигурации лог-коллектора. В разделе **eventlog_collector** необходимо указать в строке **channel** имена всех журналов, события которых нужно отправить на Платформу, через запятую.

```
Пример: channel: ['Security','Microsoft-Windows-Sysmon/Operational']
```

9. Описание

Раздел «Правила обработки событий» содержит описание этапов обработки событий и рекомендации по настройке правил для их обработки. Описаны **поля нормализации**. Описаны **специальные функции для работы с полями нормализации** для дополнительной обработки событий прямо в веб-интерфейсе Платформы.

В рамках услуги по технической поддержке могут быть разработаны правила разбора событий для источников, не входящих в стандартный пакет поставки. Срок разработки от 1 рабочего дня.

Платформа гарантирует обработку и анализ событий в режиме, близком к реальному времени.

Платформа обеспечивает обработку мультиязычных событий.

9.1. Этапы обработки события

Событие, поступившее в Платформу, проходит следующие этапы обработки:

- **Сбор** – получение события от целевой системы/лог-коллектора, сохранение на диск в raw-формате или добавление в очередь.
- **Фильтрация** – выделение событий, удовлетворяющих условиям правил фильтрации.
- **Определение типа** – определение типа системы от которой поступило событие для выбора правильных правил разбора и нормализации. Определение типа может быть статическим (задается в конфигурационном файле) и динамическим (с помощью специального правила).
- **Разбор** – разбиение необработанного текста события на фрагменты полезных данных.
- **Нормализация** – приведение всех данных, содержащихся в событии, к единой форме представления. На данном этапе также происходит категоризация событий.
- **Обогащение** – добавление в нормализованное событие дополнительной информации, полезной для выявления и расследования инцидентов.
- **Корреляция** – сопоставление данных из одного или нескольких событий с дополнительной информацией с целью выявления инцидента информационной безопасности.

10. Описание этапов разбора

10.1. Проверка этапов парсинга

В основном, все источники посылают события в формате RAW-JSON. При разборе событий в этом формате необходимо в качестве первого этапа использовать парсер JSON, а потом один из представленных в руководстве.

При использовании нескольких этапов разбора событий в каждом дополнительно создаваемом парсере необходимо указывать в поле «Цель» ту переменную, значение которой необходимо разобрать.

10.1.1. JSON

Сырое событие:

{"rs_collector_hostname":"v-stand-09","rs_relay_fqdn":"172.30.254.106","rs_relay_ip":"172.30.254.106","rs_collector_ts":"2021-11-18T11:52:54.446148+03:00","__rs_module":"2613-Kaspersky-SecurityCenter-db-host-activity","fqdn":"WINSRV02.demo.local","ip_address":"192.168.100.100","last_info_update":"2021-11-18T08:18:49.000000+00:00","last_net_agent_connected":null,"last_update":null,"last_visible":"2021-09-23T10:06:28.000000+00:00","nId":43,"nLastRtpState":0,"nStatus":0,"rs_agent_fqdn":"log-collector","rs_agent_ip":"172.30.254.106","rs_agent_ts":"2021-11-18T11:52:54.4389503+03:00","wstrDisplayName":"WINSRV02","wstrDnsDomain":"demo.local"}

Результат обработки представлен на рисунке 26.

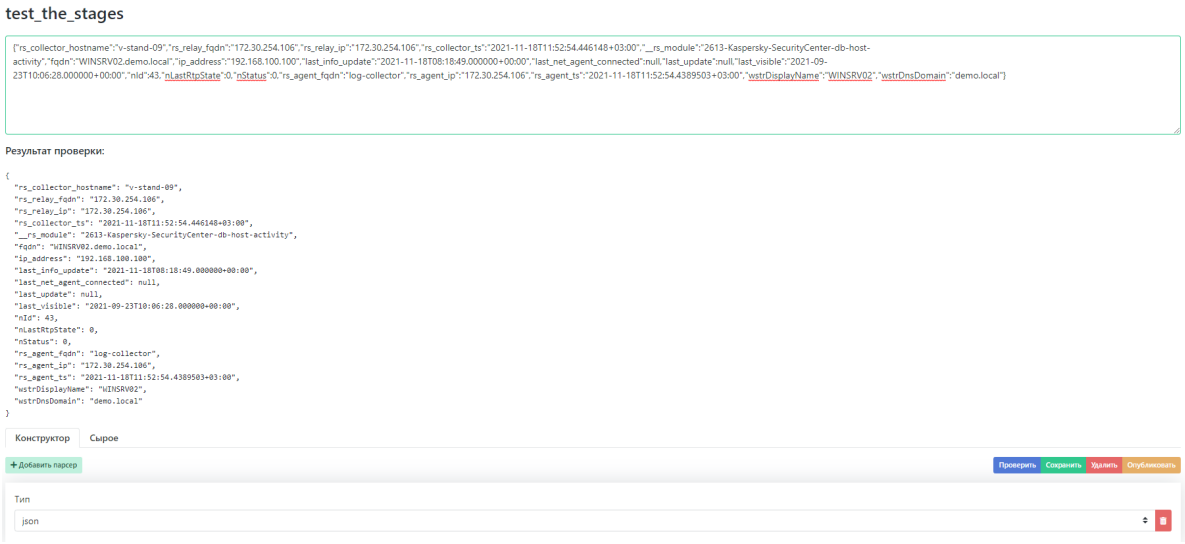

Рисунок 26 - Результат обработки этапа JSON

Результат обработки в текстовом виде:


```
{
  "rs_collector_hostname": "v-stand-09",
  "rs_relay_fqdn": "172.30.254.106",
  "rs_relay_ip": "172.30.254.106",
  "rs_collector_ts": "2021-11-18T11:52:54.446148+03:00",
  "__rs_module": "2613-Kaspersky-SecurityCenter-db-host-activity",
  "fqdn": "WINSRV02.demo.local",
  "ip_address": "192.168.100.100",
  "last_info_update": "2021-11-18T08:18:49.000000+00:00",
  "last_net_agent_connected": null,
  "last_update": null,
  "last_visible": "2021-09-23T10:06:28.000000+00:00",
  "nId": 43,
  "nLastRtpState": 0,
  "nStatus": 0,
  "rs_agent_fqdn": "log-collector",
  "rs_agent_ip": "172.30.254.106",
  "rs_agent_ts": "2021-11-18T11:52:54.4389503+03:00",
  "wstrDisplayName": "WINSRV02",
  "wstrDnsDomain": "demo.local"
}
```

10.1.2. CEF_NONSTRICT

Сырое событие:

```
CEF:1|IP Flow|IP Flow|9|flow|NetFlow Event|Unknown| eventId=13252253246
start=1623223861208 end=1623223861272 proto=TCP in=1098
categoryBehavior=/Communicate categoryDeviceGroup=/Network Equipment
catdt=Network Monitoring categoryOutcome=/Attempt categoryObject=/Host
art=1623224462176 rt=1623223873000 deviceDirection=0 src=172.0.218.2
sourceZoneURI=/All Zones/ArcSight System/Private Address Space Zones/RFC1918:
10.0.0.0-10.255.255.255 spt=8787 dst=172.0.18.108 destinationZoneURI=/All
Zones/ArcSight System/Private Address Space Zones/RFC1918: 10.0.0.0-
10.255.255.255 dpt=53445 fileType=NAT Source IPv4 Address: fileHash=NAT Source
Port: oldFileType=NAT Destination IPv4 Address: oldFileHash=NAT Destination Port:
cs1=172.0.245.1 cs4=13 cs5=26 cn1=9 cn3=0 cs1Label=nexthop cs2Label=src_as
cs3Label=dst_as cs4Label=src_mask cs5Label=dst_mask cs6Label=tcp_flags descr
cn1Label=in_pkts cn2Label=out_pkts cn3Label=tcp_flags ahost=arcsight-test
agt=172.0.6.96 agentZoneURI=/All Zones/ArcSight System/Private Address Space
Zones/RFC1918: 10.0.0.0-10.255.255.255 amac=34-B3-54-BC-66-C6 av=7.14.0.8241.0
atz=Europe/Moscow at=cisco_netflow dvchost=arcsight-test dvc=172.0.255.245
deviceZoneURI=/All Zones/ArcSight System/Private Address Space Zones/RFC1918:
10.0.0.0-10.255.255.255 dtz=Europe/Moscow geid=0 _cefVer=1.0
ad.flow__sampler__id=0 ad.vendor__51=0 ad.DevicePort=61673
ad.interface__output__snmp=312 ad.src__tos=0 ad.pkthdr__uptime=444691076
ad.pkthdr__seq=787165105 ad.pkthdr__source__id=517 ad.pkthdr__count=32
ad.interface__input__snmp=153 aid=3hughqHkBABCBSulnxz6OxA\\=\\=
```

Результат обработки в текстовом виде:


```
{
  "rs_collector_hostname": "radar-balancer-01",
  "rs_relay_fqdn": "arcsight-test",
  "rs_relay_ip": "172.0.0.96",
  "rs_collector_ts": "2021-06-09T10:41:02.253872+03:00",
  "__rs_module": "3500-Arcsight-Smartconnector-Netflow-cef",
  "cef_version": 1,
  "vendor": "IP Flow",
  "product": "IP Flow",
  "version": "9",
  "signature": "flow",
  "name": "NetFlow Event",
  "severity": "Unknown",
  "eventId": "13252253246",
  "start": "1623223861208",
  "end": "1623223861272",
  "proto": "TCP",
  "in": "1098",
  "categoryBehavior": "/Communicate",
  "categoryDeviceGroup": "/Network Equipment",
  "catdt": "Network Monitoring",
  "categoryOutcome": "/Attempt",
  "categoryObject": "/Host",
  "art": "1623224462176",
  "rt": "1623223873000",
```

```
  "deviceDirection": "0",
  "src": "172.0.218.2",
  "sourceZoneURI": "/All Zones/ArcSight System/Private Address Space
Zones/RFC1918: 10.0.0.0-10.255.255.255",
  "spt": "8787",
  "dst": "172.0.18.108",
  "destinationZoneURI": "/All Zones/ArcSight System/Private Address Space
Zones/RFC1918: 10.0.0.0-10.255.255.255",
  "dpt": "53445",
  "fileType": "NAT Source IPv4 Address:",
  "fileHash": "NAT Source Port:",
  "oldFileType": "NAT Destination IPv4 Address:",
  "oldFileHash": "NAT Destination Port:",
  "ahost": "arcsight-test",
  "agt": "172.0.6.96",
  "agentZoneURI": "/All Zones/ArcSight System/Private Address Space
Zones/RFC1918: 10.0.0.0-10.255.255.255",
  "amac": "34-B3-54-BC-66-C6",
  "av": "7.14.0.8241.0",
  "atz": "Europe/Moscow",
  "at": "cisco_netflow",
  "dvchost": "arcsight-test",
  "dvc": "172.0.255.245",
  "deviceZoneURI": "/All Zones/ArcSight System/Private Address Space
Zones/RFC1918: 10.0.0.0-10.255.255.255",
  "dtz": "Europe/Moscow",
  "geid": "0",
  "_cefVer": "1.0",
  "ad.flow__sampler__id": "0",
  "ad.vendor__51": "0",
  "ad.DevicePort": "61673",
  "ad.interface__output__snmp": "312",
  "ad.src__tos": "0",
  "ad.pkthdr__uptime": "444691076",
  "ad.pkthdr__seq": "787165105",
  "ad.pkthdr__source__id": "517",
  "ad.pkthdr__count": "32",
  "ad.interface__input__snmp": "153",
  "aid": "3hughqHkBABCBSulnxz6OxA==",
  "nexthop": "172.0.245.1",
  "src_mask": "13",
  "dst_mask": "26",
  "in_pkts": "9",
  "tcp_flags": "0"
}
```


10.1.3. CEF

Сырое событие:

```
CEF:0|InfoTeCS|IDS|2.4.3-371989|1:2023753:2|ET SCAN MS Terminal Server Traffic on
Non-standard Port|2|cat=1 cn1=348158796 cn1Label=EventID cnt=1 cs1=attempted-
recon cs1Label=IDSClass cs2=emerging-scan cs2Label=IDSGroup cs3= cs3Label=CVEID
cs4= cs4Label=ExternalRef cs5= cs5Label=IDSTags deviceExternalId=341000778
deviceFacility=Signature dmac=00:1c:58:8b:46:00 dpt=54321 dst=62.33.180.235
proto=TCP rt=May 31 2021 19:36:57.181 YEKT smac=84:78:ac:34:5e:a2 spt=2324
src=95.142.121.19
```

Результат обработки представлен на рисунке 27:

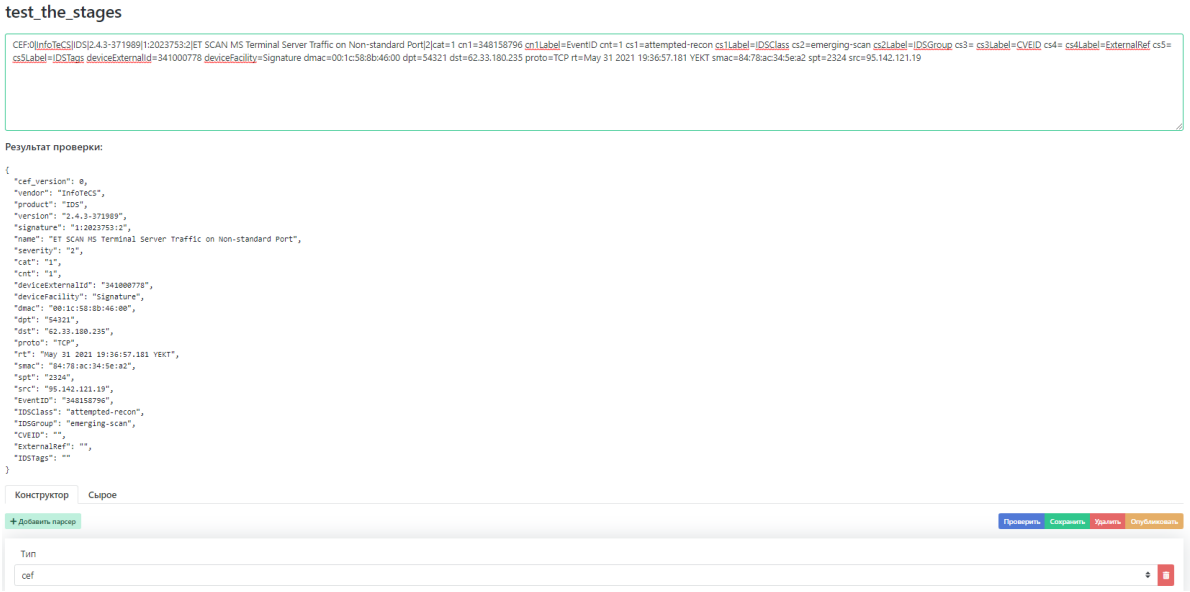

Рисунок 27 - Результат обработки этапа CEF

Результат обработки в текстовом виде:


```
{
  "cef_version": 0,
  "vendor": "InfoTeCS",
  "product": "IDS",
  "version": "2.4.3-371989",
  "signature": "1:2023753:2",
  "name": "ET SCAN MS Terminal Server Traffic on Non-standard Port",
  "severity": "2",
  "cat": "1",
  "cnt": "1",
  "deviceExternalId": "341000778",
  "deviceFacility": "Signature",
  "dmac": "00:1c:58:8b:46:00",
  "dpt": "54321",
  "dst": "62.33.180.235",
  "proto": "TCP",
  "rt": "May 31 2021 19:36:57.181 YEKT",
  "smac": "84:78:ac:34:5e:a2",
  "spt": "2324",
  "src": "95.142.121.19",
  "EventID": "348158796",
  "IDSClass": "attempted-recon",
  "IDSGroup": "emerging-scan",
  "CVEID": "",
  "ExternalRef": "",
```

```
  "IDSTags": ""
}
```


10.1.4. XML

Сырое событие:


```
<AuditRecord><Audit_Type>1</Audit_Type><Session_Id>250388</Session_Id>
<StatementId>1</StatementId><EntryId>1</EntryId><Extended_Timestamp>2020-08-
25T19:57:32.604660Z</Extended_Timestamp><DB_User>RADAR</DB_User>
<OS_User>oracle</OS_User><Userhost>805cd2dc9016</Userhost>
<OS_Process>1313</OS_Process><Terminal>pts/0</Terminal>
<Instance_Number>0</Instance_Number><Action>100</Action>
<TransactionId>12001300EE070000</TransactionId><Returncode>0</Returncode>
<Comment_Text>Authenticated by: DATABASE</Comment_Text><Priv_Used>5</Priv_Used>
<DBID>2722566360</DBID><Current_User>RADAR</Current_User>\\n</AuditRecord>
```


Результат обработки представлен на рисунке 28:

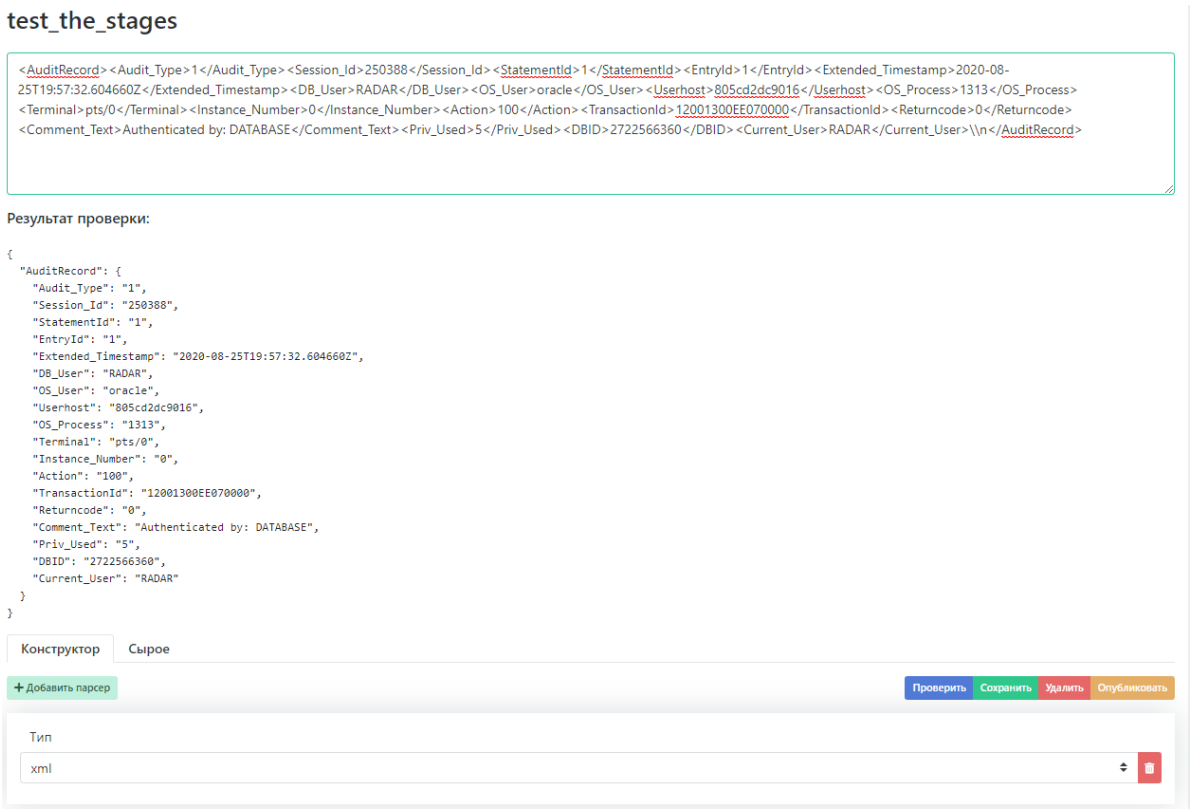

Рисунок 28 - Результат обработки этапа XML

Результат обработки в текстовом виде:


```
{
  "AuditRecord": {
    "Audit_Type": "1",
    "Session_Id": "250388",
    "StatementId": "1",
    "EntryId": "1",
    "Extended_Timestamp": "2020-08-25T19:57:32.604660Z",
    "DB_User": "RADAR",
    "OS_User": "oracle",
    "Userhost": "805cd2dc9016",
```

```
    "OS_Process": "1313",
    "Terminal": "pts/0",
    "Instance_Number": "0",
    "Action": "100",
    "TransactionId": "12001300EE070000",
    "Returncode": "0",
    "Comment_Text": "Authenticated by: DATABASE",
    "Priv_Used": "5",
    "DBID": "2722566360",
    "Current_User": "RADAR"
  }
}
```


10.1.5. CSV

Сырое событие:


```
"-1","domain618\\user286","10.10.200.10","POST","2619","500","host333.domain66.net","/path5","DENIED","","1557410124","2019-05-09 13:55:24","https","Streaming Media","","","Minimal Risk","Block URLs Whose Category Is in Category Blocklist for Default Groups","403","10.10.23.19","","Blocked by URL filtering","Other","","Google Update/1.3.33.23;winhttp\
```


Настройка этапа разбора представлена на рисунках:

Рисунок 29

Результат разбора представлен на рисунке 30:

Результат проверки:


```
{
  "user_id": "-1",
  "username": "domain618\\user286",
  "source_ip": "10.10.200.10",
  "http_action": "POST",
  "server_to_client_bytes": "2619",
  "client_to_server_bytes": "500",
  "requested_host": "host333.domain66.net",
  "requested_path": "/path5",
  "result": "DENIED",
  "virus": "",
  "request_timestamp_epoch": "1557410124",
  "request_timestamp": "2019-05-09 13:55:24",
  "uri_scheme": "https",
  "category": "Streaming Media",
  "media_type": "",
  "application_type": "",
  "reputation": "Minimal Risk",
  "last_rule": "Block URLs Whose Category Is in Category Blocklist for Default Groups",
  "http_status_code": "403",
  "client_ip": "10.10.23.19",
  "location": "",
  "block_reason": "Blocked by URL filtering",
  "user_agent_product": "Other",
  "user_agent_version": "",
  "user_agent_comment": "Google Update/1.3.33.23;winhttp\n"
}
```


Рисунок 30 - Результат обработки этапа CSV

Результат разбора в текстовом виде:


```
{
  "user_id": "-1",
  "username": "domain618\\user286",
  "source_ip": "10.10.200.10",
  "http_action": "POST",
  "server_to_client_bytes": "2619",
  "client_to_server_bytes": "500",
  "requested_host": "host333.domain66.net",
  "requested_path": "/path5",
  "result": "DENIED",
  "virus": "",
  "request_timestamp_epoch": "1557410124",
  "request_timestamp": "2019-05-09 13:55:24",
  "uri_scheme": "https",
  "category": "Streaming Media",
  "media_type": "",
  "application_type": "",
  "reputation": "Minimal Risk",
  "last_rule": "Block URLs Whose Category Is in Category Blocklist for Default Groups",
  "http_status_code": "403",
  "client_ip": "10.10.23.19",
  "location": "",
  "block_reason": "Blocked by URL filtering",
  "user_agent_product": "Other",
  "user_agent_version": "",
  "user_agent_comment": "Google Update/1.3.33.23;winhttp\n"
}
```

10.1.6. GROK

Для работы парсера необходимо завалидировать (проверить) паттерн и в случае успеха добавить его для использования, нажав на зеленый плюсик рядом с синей кнопкой «Валидировать».

Сырое событие:

```
<86>v-demo-checkpoint sshd[13236]: Accepted password for admin from 192.168.200.2
port 1091 ssh2
```

GROK-паттерн:

```
<?>%{DATA:application_name}\s+%{WORD:service}.*?\s+%{DATA:attempt}\s+for\s+%
{USERNAME:username}\s+from\s+%{IPORHOST:from_host}\s+port\s+%
{BASE10NUM:source_port}\s+%{DATA:transport}$
```

Результат обработки представлен на рисунке 31:

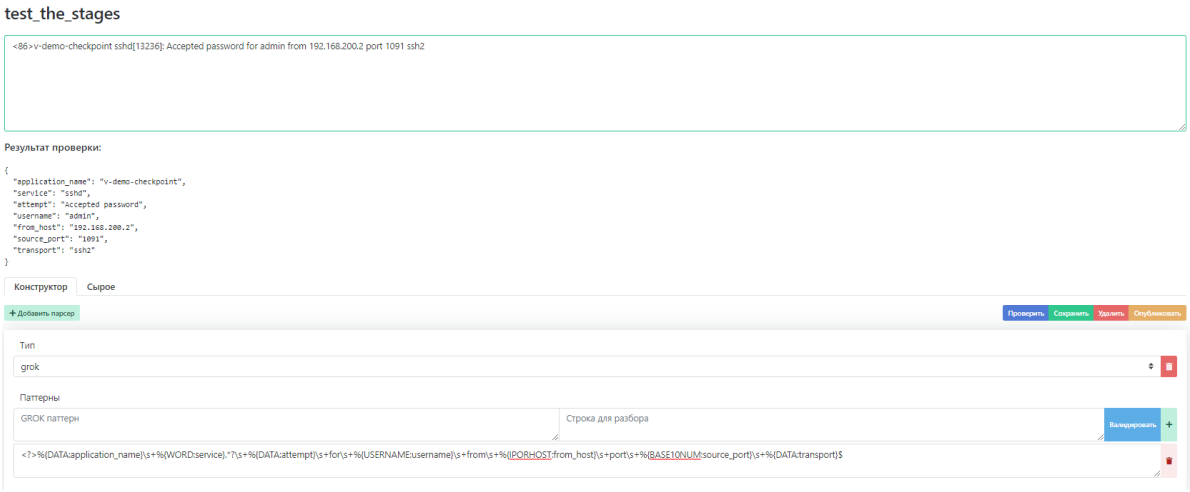

Рисунок 31 - Результат обработки этапа GROK

Результат обработки в текстовом виде:

```
{
  "application_name": "v-demo-checkpoint",
  "service": "sshd",
  "attempt": "Accepted password",
  "username": "admin",
  "from_host": "192.168.200.2",
  "source_port": "1091",
  "transport": "ssh2"
}
```

11. Разработка правил разбора и нормализации

11.1. Создание правил разбора

1. Для создания правил разбора необходимо перейти в раздел «Источники» → «Управление источниками». Далее выбрать вкладку «Правила разбора», после чего откроется страница создания, редактирования и просмотра парсеров (Рисунок 1).

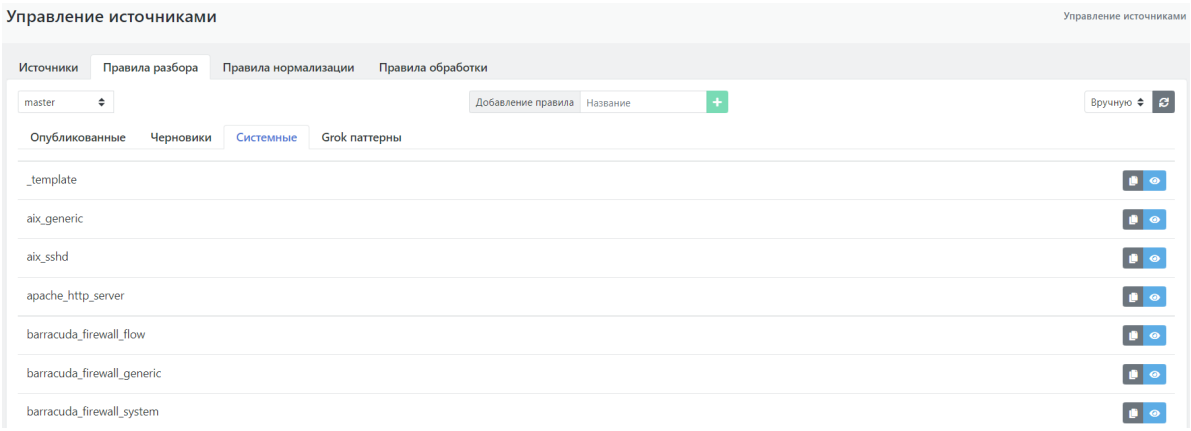

Рисунок 32 - Страница создания, редактирования и просмотра правил разбора

2. Для создания нового правила разбора необходимо указать название в поле «Добавление правила» и нажать на «+», после чего откроется форма создания правила разбора (Рисунок 2).

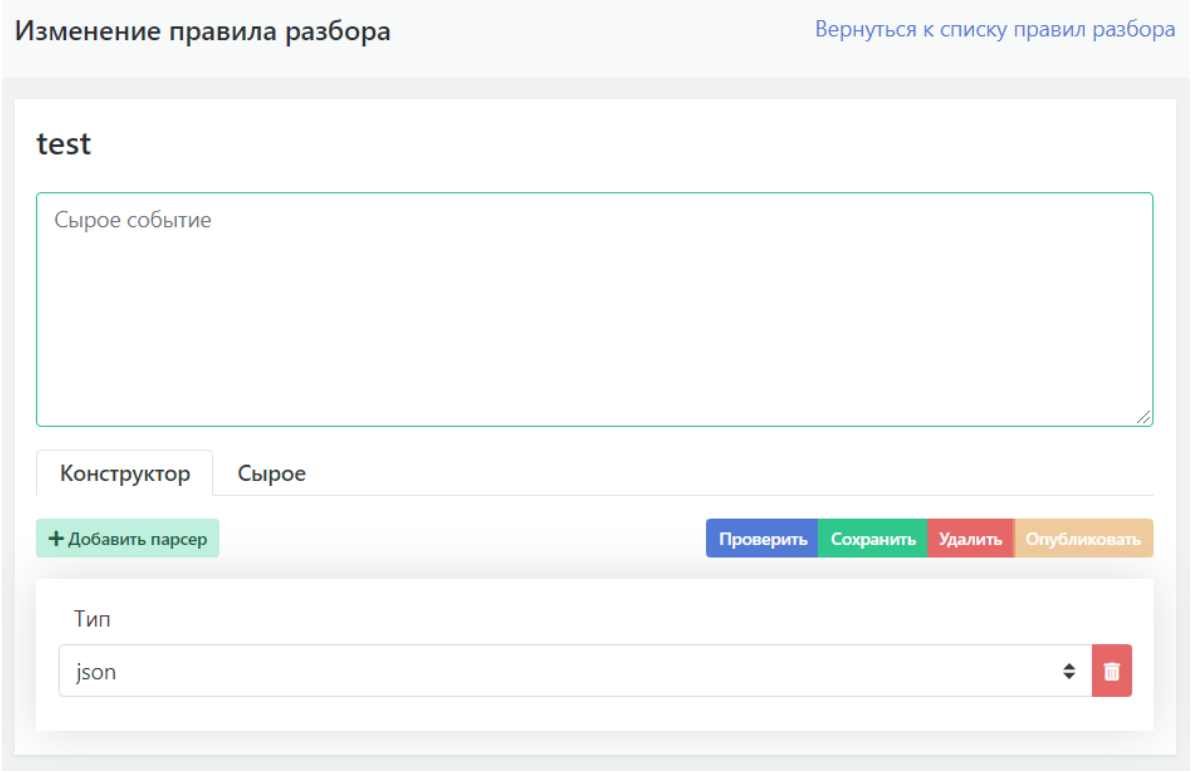

Рисунок 33 - Окно создания правила разбора

3. Рассмотрим процесс создания парсера на примере событий от продукта Micro Focus ArcSight SmartConnector.

Тип продукта: сетевое оборудование

Сырое событие:

{"rs_collector_hostname":"radar-balancer-01","rs_relay_fqdn":"arcsight-test","rs_relay_ip":"172.0.0.96","rs_collector_ts":"2021-06-09T10:41:02.253872+03:00","__rs_module":"3500-Arcsight-Smartconnector-Netflow-cef","message":"CEF:1|IP Flow|IP Flow|9|flow|NetFlow Event|Unknown|
eventId=13252253246 start=1623223861208 end=1623223861272 proto=TCP in=1098
categoryBehavior=\/Communicate categoryDeviceGroup=\/Network Equipment
catdt=Network Monitoring categoryOutcome=\/Attempt categoryObject=\/Host
art=1623224462176 rt=1623223873000 deviceDirection=0 src=172.0.218.2
sourceZoneURI=\/All Zones\/ArcSight System\/Private Address Space Zones\/RFC1918:
10.0.0.0-10.255.255.255 spt=8787 dst=172.0.18.108 destinationZoneURI=\/All
Zones\/ArcSight System\/Private Address Space Zones\/RFC1918: 10.0.0.0-
10.255.255.255 dpt=53445 fileType=NAT Source IPv4 Address: fileHash=NAT Source
Port: oldFileType=NAT Destination IPv4 Address: oldFileHash=NAT Destination Port:
cs1=172.0.245.1 cs4=13 cs5=26 cn1=9 cn3=0 cs1Label=nexthop cs2Label=src_as
cs3Label=dst_as cs4Label=src_mask cs5Label=dst_mask cs6Label=tcp_flags descr
cn1Label=in_pkts cn2Label=out_pkts cn3Label=tcp_flags ahost=arcsight-test
agt=172.0.6.96 agentZoneURI=\/All Zones\/ArcSight System\/Private Address Space
Zones\/RFC1918: 10.0.0.0-10.255.255.255 amac=34-B3-54-BC-66-C6 av=7.14.0.8241.0
atz=Europe\/Moscow at=cisco_netflow dvchost=arcsight-test dvc=172.0.255.245
deviceZoneURI=\/All Zones\/ArcSight System\/Private Address Space Zones\/RFC1918:
10.0.0.0-10.255.255.255 dtz=Europe\/Moscow geid=0 _cefVer=1.0
ad.flow__sampler__id=0 ad.vendor__51=0 ad.DevicePort=61673
ad.interface__output__snmp=312 ad.src__tos=0 ad.pkthdr__uptime=444691076
ad.pkthdr__seq=787165105 ad.pkthdr__source__id=517 ad.pkthdr__count=32
ad.interface__input__snmp=153 aid=3hughqHkBABCBSulnxz6OxA\\=\\="}

> Обратите внимание, что сырое событие представлено в формате JSON

4. Сырое событие необходимо вставить в соответствующее поле.
5. Во вкладке «Тип» нужно указать «json».
6. После нажатия кнопки «Проверить» получаем результат разобранного JSON события.

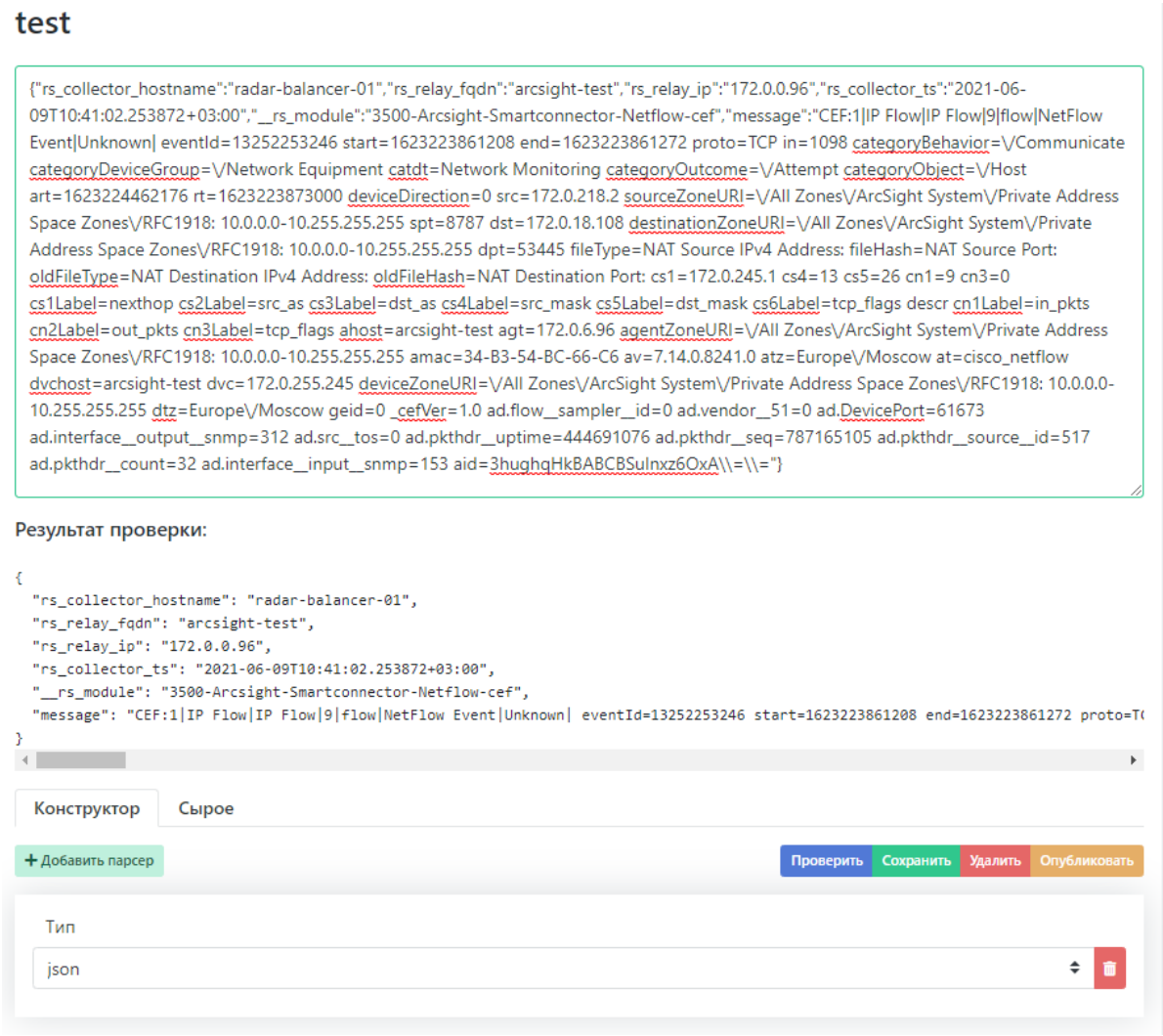

Рисунок 34 - Результат разобранного события

7. Как видно из результата, одного этапа разбора недостаточно, потому что основная информация данного события находится в поле «message».
8. В качестве второго этапа разбора необходимо использовать этап CEF. Для этого следует нажать на кнопку «Добавить парсер» и выбрать «cef_nonstrict» (этот этап используется для разбора формата CEF версии 1).

9. Далее в поле «Цель» второго этапа разбора нужно указать название поля, которое необходимо разобрать, в случае рассматриваемого примера - это поле «message».

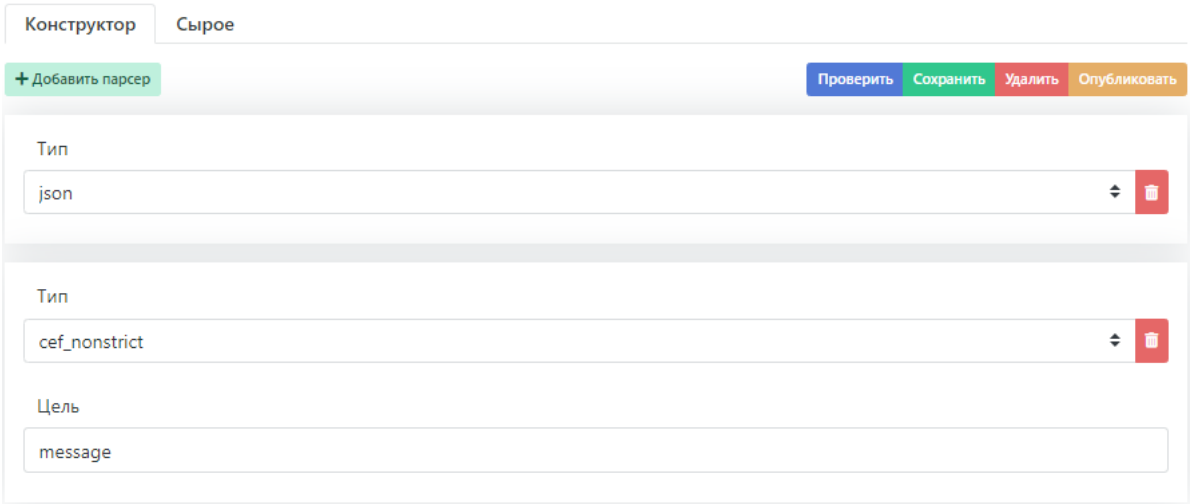

Рисунок 35 - Добавление второго этапа разбора

10. Проверяем работоспособность этапов правила разбора нажатием на кнопку «Проверить».
11. Результатом проверки правила должен быть вывод полностью разобранного события:


```
{
  "rs_collector_hostname": "radar-balancer-01",
  "rs_relay_fqdn": "arcsight-test",
  "rs_relay_ip": "172.0.0.96",
  "rs_collector_ts": "2021-06-09T10:41:02.253872+03:00",
  "__rs_module": "3500-Arcsight-Smartconnector-Netflow-cef",
  "cef_version": 1,
  "vendor": "IP Flow",
  "product": "IP Flow",
  "version": "9",
  "signature": "flow",
  "name": "NetFlow Event",
  "severity": "Unknown",
  "eventId": "13252253246",
  "start": "1623223861208",
  "end": "1623223861272",
  "proto": "TCP",
  "in": "1098",
  "categoryBehavior": "/Communicate",
  "categoryDeviceGroup": "/Network Equipment",
  "catdt": "Network Monitoring",
  "categoryOutcome": "/Attempt",
  "categoryObject": "/Host",
  "art": "1623224462176",
  "rt": "1623223873000",
  "deviceDirection": "0",
  "src": "172.0.218.2",
  "sourceZoneURI": "/All Zones/ArcSight System/Private Address Space
Zones/RFC1918: 10.0.0.0-10.255.255.255",
  "spt": "8787",
  "dst": "172.0.18.108",
  "destinationZoneURI": "/All Zones/ArcSight System/Private Address Space
Zones/RFC1918: 10.0.0.0-10.255.255.255",
  "dpt": "53445",
  "fileType": "NAT Source IPv4 Address:",
  "fileHash": "NAT Source Port:",
  "oldFileType": "NAT Destination IPv4 Address:",
  "oldFileHash": "NAT Destination Port:",
  "ahost": "arcsight-test",
  "agt": "172.0.6.96",
  "agentZoneURI": "/All Zones/ArcSight System/Private Address Space
Zones/RFC1918: 10.0.0.0-10.255.255.255",
  "amac": "34-B3-54-BC-66-C6",
  "av": "7.14.0.8241.0",
  "atz": "Europe/Moscow",
  "at": "cisco_netflow",
  "dvchost": "arcsight-test",
  "dvc": "172.0.255.245",
  "deviceZoneURI": "/All Zones/ArcSight System/Private Address Space
Zones/RFC1918: 10.0.0.0-10.255.255.255",
  "dtz": "Europe/Moscow",
  "geid": "0",
```

```
  "_cefVer": "1.0",
  "ad.flow__sampler__id": "0",
  "ad.vendor__51": "0",
  "ad.DevicePort": "61673",
  "ad.interface__output__snmp": "312",
  "ad.src__tos": "0",
  "ad.pkthdr__uptime": "444691076",
  "ad.pkthdr__seq": "787165105",
  "ad.pkthdr__source__id": "517",
  "ad.pkthdr__count": "32",
  "ad.interface__input__snmp": "153",
  "aid": "3hughqHkBABCBSulnxz6OxA==",
  "nexthop": "172.0.245.1",
  "src_mask": "13",
  "dst_mask": "26",
  "in_pkts": "9",
  "tcp_flags": "0"
}
```


12. После успешной проверки этапов разбора необходимо нажать кнопку «Сохранить» для сохранения правила и следом нажать кнопку «Опубликовать» для последующего его использования.

> Описание и примеры использования возможных этапов разбора событий представлены в документации по описанию этапов разбора

11.2. Создание правил нормализации

1. Для создания правила нормализации необходимо перейти в раздел “Источники” → “Управление источниками”, выбрать вкладку “Правила нормализации”, после чего откроется страница создания, редактирования и просмотра правил нормализации.
2. Для создания нового правила нормализации необходимо указать название в поле "Добавление правила" и нажать на “+”, после чего откроется окно создания правила нормализации, изображенное на рисунке 36;

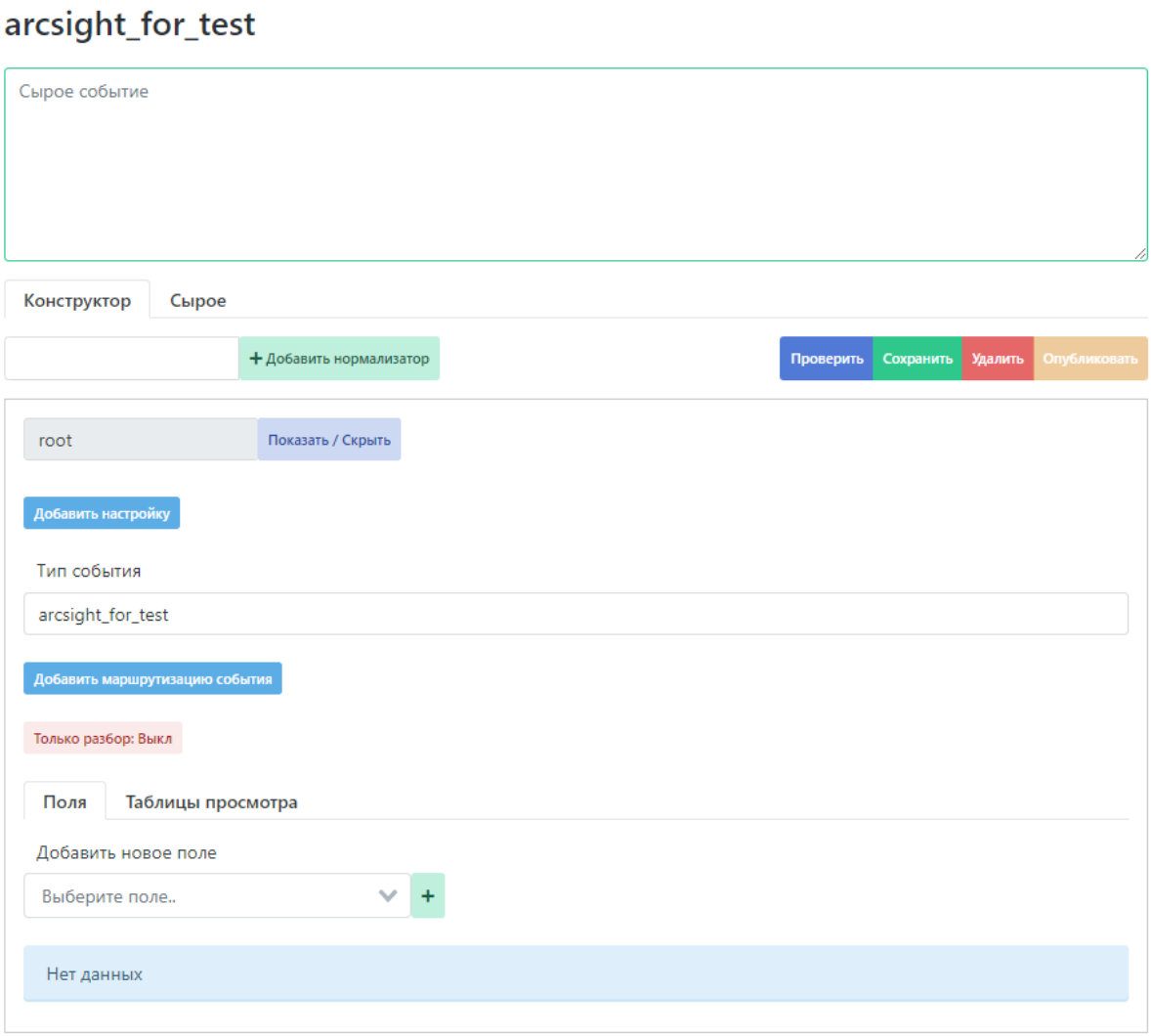

Рисунок 36 - Окно создания правила нормализации

> В качестве названия правила нормализации нужно указывать уникальный идентификатор сообщения для данного источника, в случае с примером: **arcsight_for_test**

3. В поле для сырого события необходимо вставить разобранное событие (результат создания правила разбора данного документа).
4. После создания правила, внутри него автоматически создается нормализатор root.yaml

> root.yaml – файл содержит декларации преобразований, общих для всех событий данной системы. Преобразования, указанные в этом файле применяются ко всем событиям системы, прошедшим стадию парсинга. Как правило они содержат классификатор источника и данные, содержащиеся в заголовке события Данный нормализатор разрабатывается один на систему.

5. Для добавления поля нормализации нужно во вкладке "Добавить новое поле" выбрать необходимое поле и нажать на "+", в результате чего в нормализатор добавится выбранное поле, как изображено на рисунке 37. Можно использовать как преднастроенные системные поля нормализации, так и добавлять пользовательские поля;

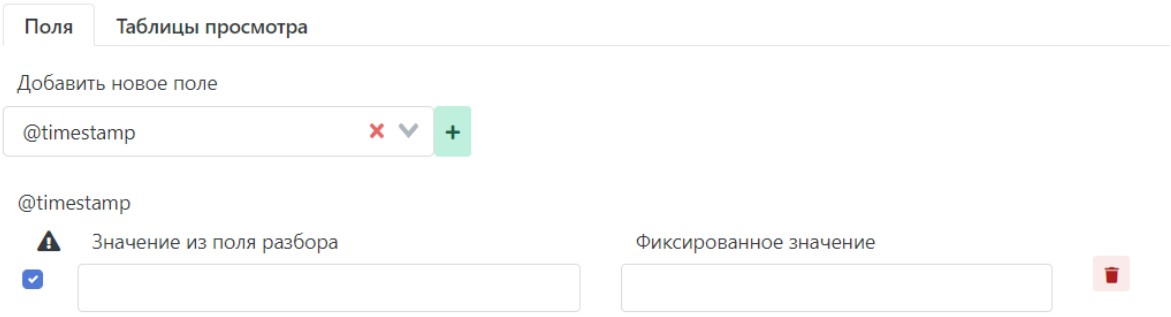

Рисунок 37 - Добавление поля нормализации

> В поле **"Значение из поля разбора"** нужно указывать одно из разобранных полей, таким образом в поле нормализации будет записано значение поля из разбора. Также в данное поле можно записывать специальные функции для работы с полями (ниже будет представлено описание некоторых из них);
>
> В поле **"Фиксированное значение"** нужно указывать произвольный набор символов соответствующий типу данного поля нормализации.
>
> Описание и типы полей нормализации представлены в документации описание полей нормализации

6. Исходя из этого, в данном файле нормализации можно использовать поля, изображенные на рисунках;

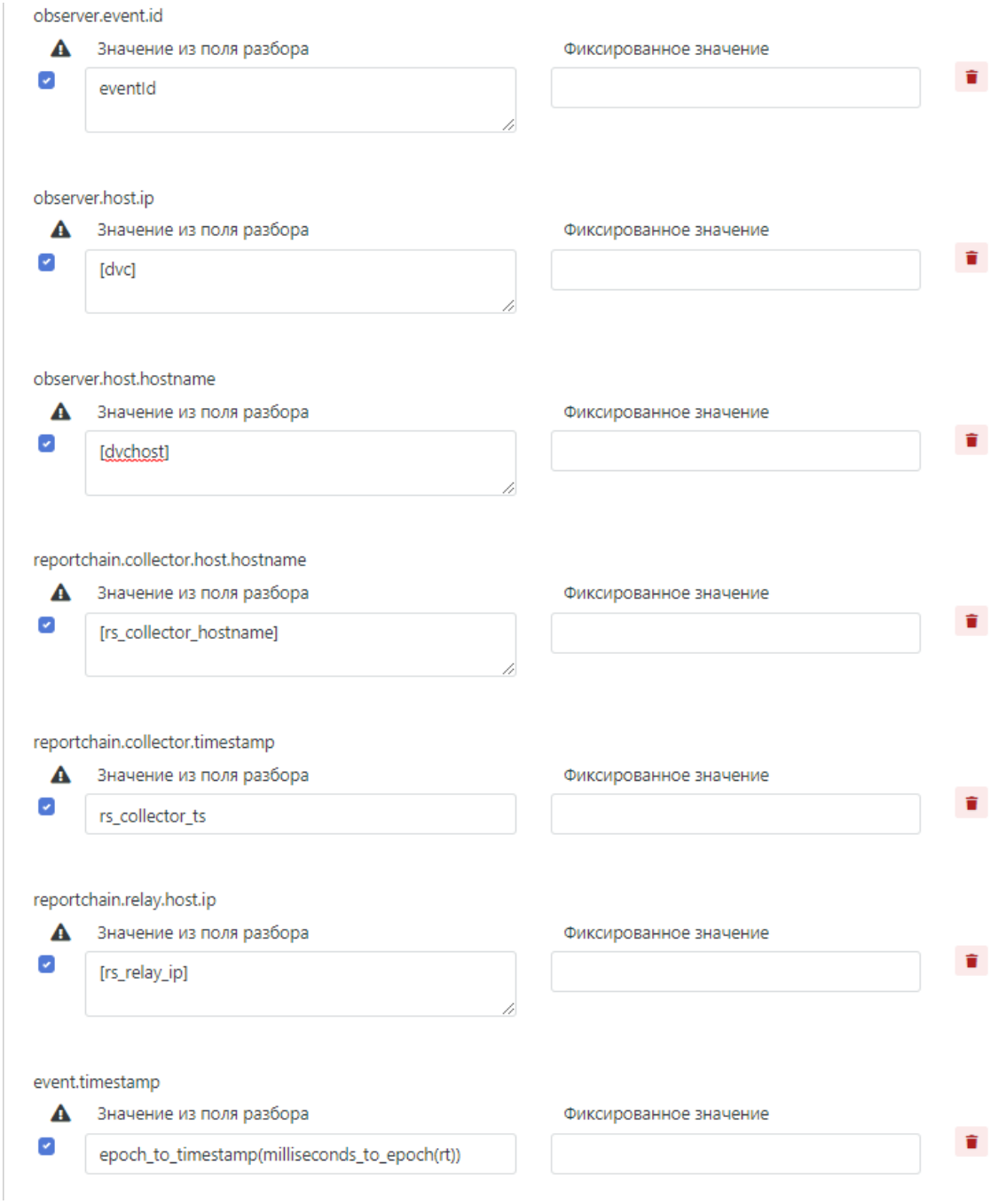

Рисунок 38

7. После добавления необходимых для нормализации полей - нужно нажать на кнопку "Проверить" для проверки, что во все поля нормализации записались нужные данные. Результат проверки изображен на рисунке 39;

Результат проверки:


```
{
  "event": {
    "uuid": "297acb480ed2433ca67daa1a67fb63ef",
    "logsource": {
      "name": "Microfocus ArcSight Smartconnector",
      "product": "arcsight",
      "vendor": "microfocus"
    },
    "timestamp": "2021-06-09T07:31:13+00:00"
  },
  "raw": null,
  "@timestamp": "2021-06-09T07:31:13+00:00",
  "observer": {
    "event": {
      "id": "13252253246"
    },
    "host": {
      "hostname": [
        "arcsight-test"
      ],
      "ip": [
        "172.0.255.245"
      ]
    }
  },
  "reportchain": {
    "collector": {
      "host": {
        "hostname": [
          "radar-balancer-01"
        ]
      },
      "timestamp": "2021-06-09T10:41:02.253872+03:00"
    },
    "relay": {
      "host": {
        "ip": [
          "172.0.0.96"
        ]
      }
    }
  }
}
```


Рисунок 39 - Результат проверки нормализации

8. После настройки нормализатора root.yaml, необходимо перейти к созданию нормализатора для определенного типа событий от данного источника, в случае с примером - это Netflow;
9. Для этого в поле рядом с кнопкой "Добавить нормализатор" нужно ввести имя нормализатора и нажать на кнопку "+ Добавить нормализатор", как изображено на рисунке 40;

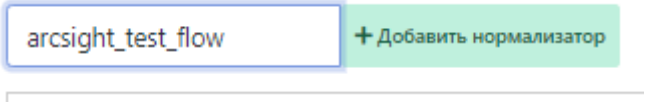

Рисунок 40 - Добавление нового нормализатора

10. Далее необходимо добавить маршрутизацию для данного нормализатора. Это делается для того, чтобы не все события от данного источника нормализовались по данному "сценарию

нормализации", а только те, которые нужны;

11. Для этого необходимо в поле "Маршрутизация события" ввести условия нормализации по данному сценарию. В качестве переменных в условии нужно использовать поля разобранного события. Заполненное поле маршрутизации изображено на рисунке 41;

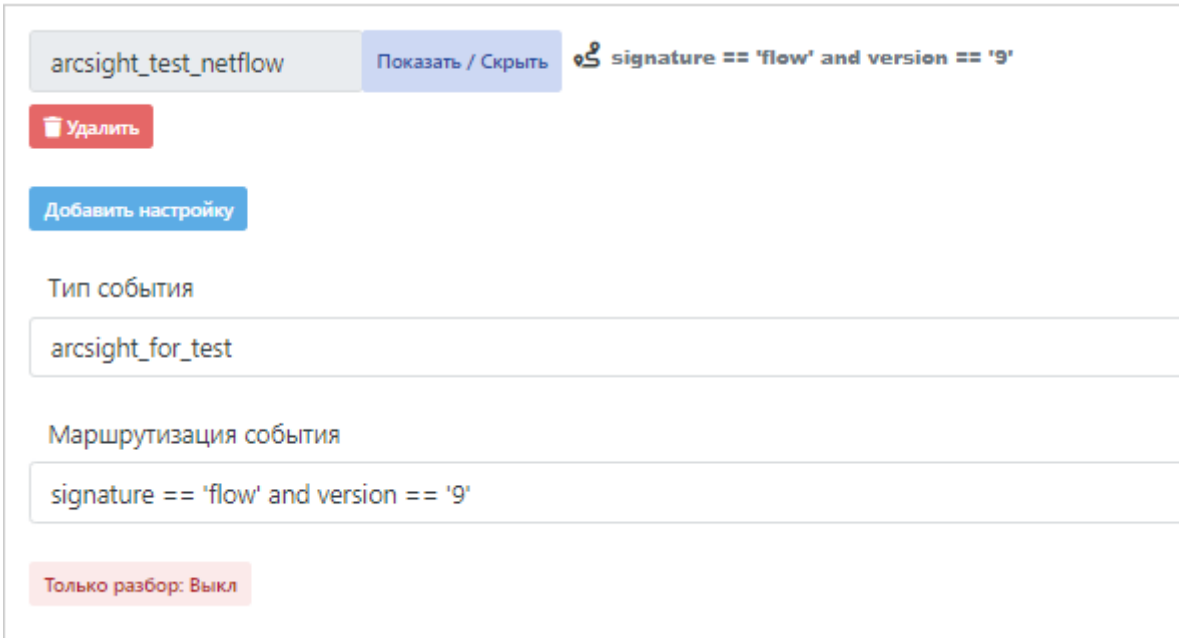

Рисунок 41 - Маршрутизация события

> Таким образом, все события, которые подходят под условие:
> **signature == 'flow' and version == '9'**
> будут нормализованы по данному файлу нормализации

12. Для более гибкой, понятной и правильной нормализации в данном файле нормализации используются специальные функции, которые подробно описаны в разделе **Специальные функции для работы с полями нормализации.**
13. Также в данном файле нормализации используются дополнительные настройки, описание которых представлены ниже;

> **Функция Tapping (Поле "Настройка")**
>
> К сожалению, логлайны, поступающие от клиентов, иногда могут быть непредсказуемыми. Таким образом, существует возможность выполнения кода Python в качестве этапа предварительной обработки.
> Событие доступно с помощью переменной *line*.
>
> Пример использования:
>
> ```
> tcp_flags = line.parsed['tcp']
>   line.parsed['flow_tags'] =
>   [
>       f"tcp_{flag}"
>       for flag in
>           ("syn", "fin", "rst", "psh", "ack", "cwr", "ecn", "urg")
>       if tcp_flags.get(flag, False)
>   ]
> ```
>
> В результате выполнения данной настройки, в нормализации можно использовать поле "flow_tags"

> **ПРЕДУПРЕЖДЕНИЕ!**
> Использование данного механизма может сказаться на производительности и скорости работы обработчиков событий.

14. Таким образом, таблица просмотра (функция lookup) для данного файла нормализации представлена на рисунке 42;

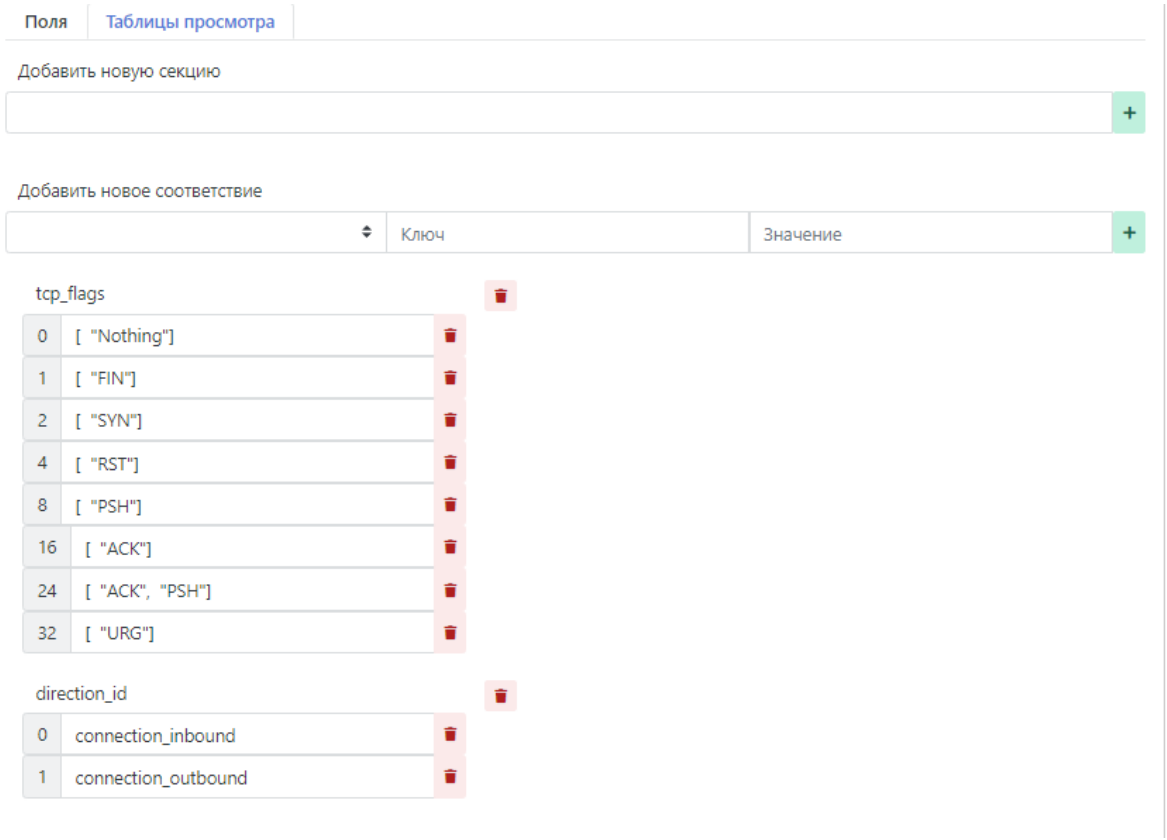

Рисунок 42 - Таблицы просмотра

15. Дополнительная настройка нормализатора изображена на рисунке 43;

Рисунок 43 - Дополнительная настройка нормализатора

> Данная настройка необходима для того чтобы в разобранном событии найти поле "in" и/или "out" и присвоить их в новые поля "in\out_bytes" в формате целых чисел.

16. Поля нормализации используемые в данном файле нормализации представлены на рисунках;

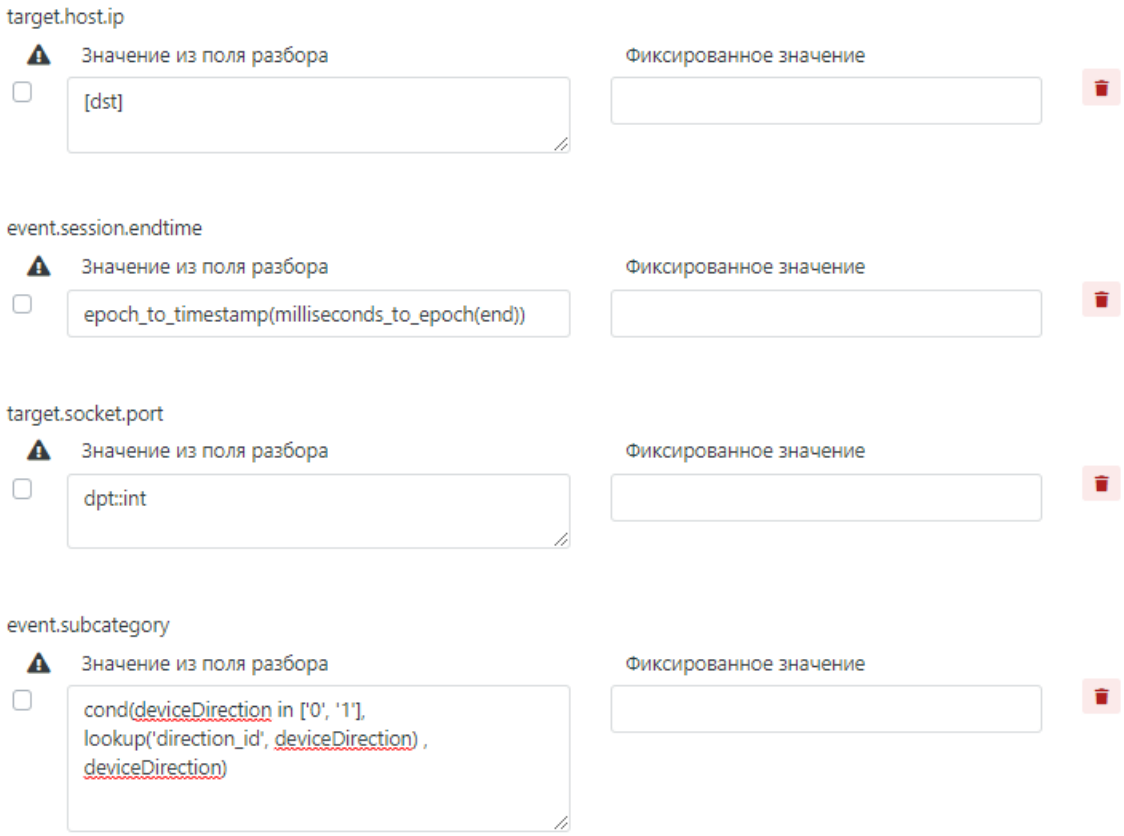

Рисунок 44

17. После добавления необходимых для нормализации полей, настроек и таблиц просмотра - нужно нажать на кнопку "Проверить" для проверки, что во все поля нормализации записались нужные данные;

Результат проверки:


```
{
  "event": {
    "uuid": "283f402ba7a04a3786ca8a52e19be872",
    "application": {
      "name": "smartconnector",
      "protocol": "TCP"
    },
    "bytes": {
      "received": 1098
    },
    "category": "connection",
    "description": "A connection was observed",
    "logsource": {
      "application": "flow",
      "name": "Microfocus ArcSight Smartconnector",
      "product": "arcsight",
      "subsystem": "communication",
      "vendor": "microfocus"
    },
    "packets": {
      "received": 9
```

```
    },
    "session": {
      "duration": 64,
      "endtime": "2021-06-09T07:31:01.272000+00:00",
      "flags": [
        "Unknown tcp flag"
      ],
      "starttime": "2021-06-09T07:31:01.208000+00:00"
    },
    "severity": 0,
    "subcategory": "connection_inbound",
    "timestamp": "2021-06-09T07:31:13+00:00"
  },
  "raw": null,
  "@timestamp": "2021-06-09T07:31:13+00:00",
  "action": "connect",
  "initiator": {
    "host": {
      "ip": [
        "172.0.218.2"
      ]
    },
    "socket": {
      "port": 8787
    }
  },
  "observer": {
    "event": {
      "id": "13252253246"
    },
    "host": {
      "hostname": [
        "arcsight-test"
      ],
      "ip": [
        "172.0.255.245"
      ]
    }
  },
  "reportchain": {
    "collector": {
      "timestamp": "2021-06-09T10:41:02.253872+03:00"
    },
    "relay": {
      "host": {
        "ip": [
          "172.0.0.96"
        ]
      }
    }
  },
  "target": {
    "host": {
      "ip": [
        "172.0.18.108"
```

```
      ]
    },
    "socket": {
      "port": 53445
    }
  }
}
```

18. После настройки основного нормализатора arcsight_test_netflow.yaml, необходимо перейти к созданию нормализатора по умолчанию parsed_only.yaml;

> parsed_only.yaml – файл используется как «нормализатор по умолчанию». Для событий прошедших через этот нормализатор создается специализированных индекс в ElasticSearch, содержащий нормализованные данные. Данный нормализатор разрабатывается один на систему. В него попадают события, которые не прошли ни по одной из маршрутизаций в других нормализаторах

19. Для его создания, в поле названия нормализатора нужно ввести "parsed_only" и нажать на кнопку "+ Добавить нормализатор";

20. После чего, в маршрутизации события указать "fallback" и кликнуть на кнопку "Только разбор" чтобы она перешла в статус "Вкл", как изображено на рисунке 45;

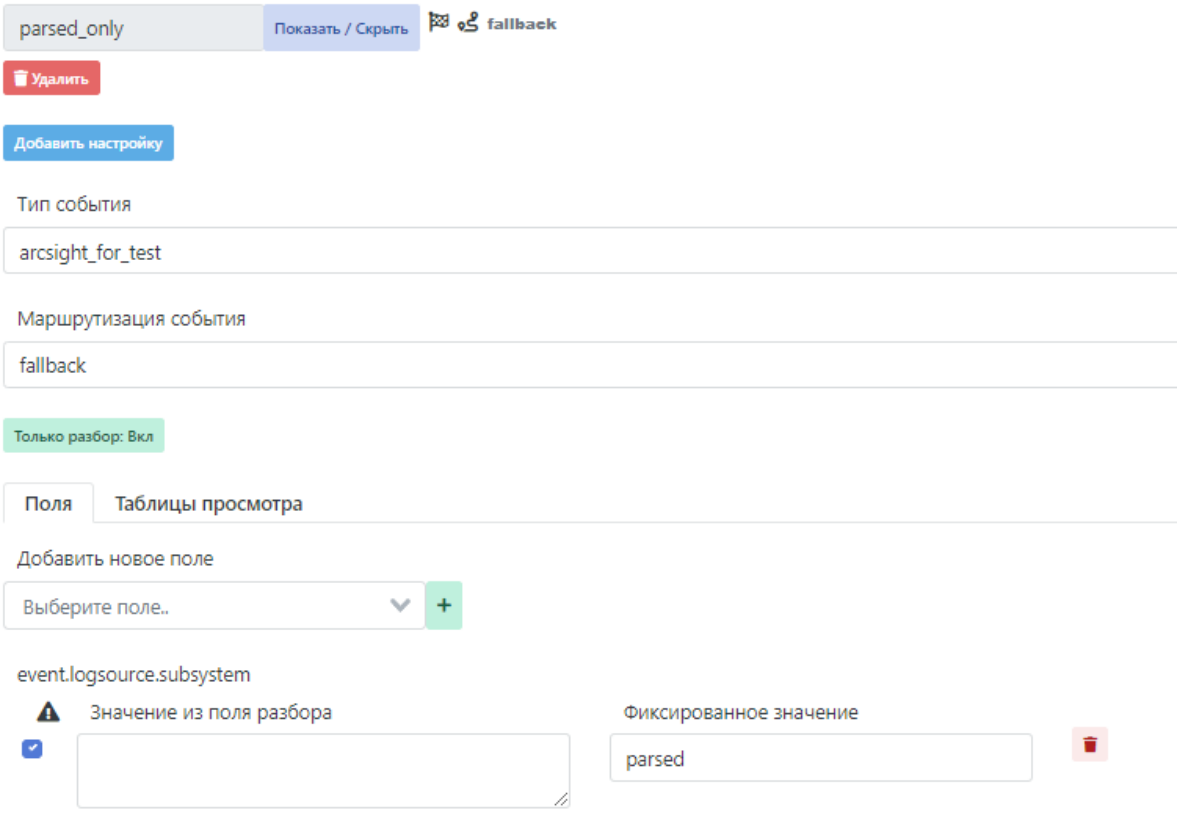

Рисунок 45 - Добавление нормализатора "parsed_only.yaml"

21. В данном нормализаторе не требуется добавления полей нормализации. В автоматически созданные поля необходимо указать "parsed" в поле "Фиксированное значение";

22. После - нужно еще раз провести проверку нормализации, нажав на кнопку "Проверить", если ошибки отсутствуют и в результате проверки ожидаемый результат можно перейти к сохранению нормализатора;

23. Для этого необходимо нажать кнопку "Сохранить" и следом кнопку "Опубликовать";

24. После чего в разделе "Правила нормализации" во вкладке "Опубликованные" должен появиться разработанный нормализатор. Это значит, что теперь его можно использовать.

12. Описание полей нормализации

Поле	Тип данных	Обязательно	Описание
@timestamp	datetime_iso	Да	Временная метка
action	keyword	Да	Действия, выполненные инициатором
event.anomaly.description	text	Нет	Описание аномалии
event.anomaly.name	text	Нет	Название аномалии
event.application.category	[keyword]	Нет	Категория приложения
event.application.content-type	keyword	Нет	Тип контента, на которое ссылается приложение, например PNG
event.application.description	keyword	Нет	Дополнительное описание приложения
event.application.name	keyword	Нет	Наименование приложения например Web Browsing, Amazon Base, Microsoft Azure Base
event.application.protocol	keyword	Нет	Наименование протокола прикладного уровня, например FTP, WebDAV, Telnet
event.application.target	keyword	Нет	Тип цели, с которой работает приложение URL, Resource
event.application.vendor	keyword	Нет	Производитель приложения
event.application.version	keyword	Нет	Версия приложения
event.auth.key.length	integer	Нет	Длина ключа аутентификации
event.auth.method.description	text	Нет	Описание метода, используемого для аутентификации RDP, Network Authentication, Command Line, Web-Client
event.auth.method.id	keyword	Нет	Идентификатор метода аутентификации
event.auth.method.name	keyword	Нет	Наименование метода аутентификации, например keyboard-interactive, public key, service, batch
event.auth.protocol.name	keyword	Нет	Наименование метода аутентификации, например, SSH, NTML, Kerberos (AuthenticationPackageName in Windows)
event.auth.protocol.version	keyword	Нет	Версия протокола аутентификации
event.blacklist	blacklist	Нет	Черный список
event.bytes.received	integer	Нет	Количество полученных байт в рамках сессии, Цель -> Инициатор
event.bytes.sent	integer	Нет	Количество отправленных байт в рамках сессии, Инициатор -> Цель
event.bytes.total	integer	Нет	Общее количество байт, отправленных в рамках сессии
event.category	keyword	Да	Категория в рамках приложения/подсистемы
event.context.raw	raw_text	Нет	Контекст события
event.correlation.id	keyword	Нет	Идентификатор сессии, позволяющий связать события
event.correlation.sequence	integer	Нет	Количество последовательных сессий
event.correlation.total	integer	Нет	Количество сессий
event.description	text	Да	Текстовое описание события

Поле	Тип данных	Обязательно	Описание
event.dns.answer.host.fqdn	[domain]	Нет	FQDN-имя на которое получен DNS-ответ
event.dns.answer.host.hostname	[domain]	Нет	Hostname на который получен DNS-ответ
event.dns.answer.host.ip	[ip]	Нет	IP адрес на который получен DNS-ответ
event.dns.answer.original	keyword	Нет	Оригинальный DNS-ответ
event.dns.id	keyword	Нет	Идентификатор DNS-запроса
event.dns.query.host.fqdn	[domain]	Нет	FQDN-имя запрашиваемое в рамках DNS-запроса
event.dns.query.host.hostname	[domain]	Нет	Hostname, запрашиваемый в рамках DNS-запроса
event.dns.query.host.ip	[ip]	Нет	IP-адрес, запрашиваемый в рамках DNS-запроса
event.dns.query.original	keyword	Нет	Оригинальный DNS-запрос
event.dns.ttl	integer	Нет	DNS time to live (время жизни)
event.dns.type	keyword	Нет	Тип DNS-записи https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-4
event.endtime	datetime_iso	Нет	Время завершения события
event.eventlist	keyword	Нет	Список событий
event.file.hash.md5	keyword	Нет	MD5-хеш файла
event.file.hash.sha1	keyword	Нет	SHA1-хеш файла
event.file.hash.sha256	keyword	Нет	SHA256-хеш файла
event.flow.id	keyword	Нет	Идентификатор Flow
event.finding.id	keyword	Нет	ID артефакта
event.finding.name	keyword	Нет	Название артефакта
event.http.protocol.name	keyword	Нет	Наименование HTTP-протокола (HTTP)
event.http.protocol.version	keyword	Нет	Версия HTTP-протокола
event.logsource.application	keyword	Да	Приложение породившее событие
event.logsource.host	keyword	Нет	Хост источника события
event.logsource.input	keyword	Нет	Input источника события
event.logsource.language	keyword	Нет	Язык источника события
event.logsource.name	keyword	Да	Наименование источника события
event.logsource.product	keyword	Да	Наименование продукта источника
event.logsource.subsystem	keyword	Да	Наименование подсистемы источника
event.logsource.vendor	keyword	Да	Наименование вендора источника
event.packet.payload.printable	text	Нет	Человекочитаемые данные из полезной нагрузки
event.packet.payload.raw	raw_text	Нет	Данные полезной нагрузки
event.packet.raw	raw_text	Нет	Сырые данные из пакета
event.packets.received	integer	Нет	Количество пакетов, полученных в рамках сессии, Цель -> Инициатор
event.packets.sent	integer	Нет	Количество пакетов, отправленных в рамках сессии, Инициатор -> Цель
event.packets.total	integer	Нет	Количество пакетов, переданных в рамках сессии
event.result.analysis_output	text	Нет	Результат анализа

Поле	Тип данных	Обязательно	Описание
event.result.description	text	Нет	Описание результата
event.result.id	keyword	Нет	ID результата
event.result.incident_identifier	keyword	Нет	Идентификатор инцидента результата
event.result.mitigation	text	Нет	
event.result.name	keyword	Нет	Наименование результата
event.result.risk_impact	text	Нет	
event.result.solution	text	Нет	Решение
event.result.synopsis	text	Нет	Краткое изложение
event.service.name	keyword	Нет	Сервис, передающий событие, например HTTP
event.session.duration	integer	Нет	Длительность сессии (в секундах)
event.session.endtime	datetime_iso	Нет	Время окончания сессии
event.session.flags	[keyword]	Нет	TCP-флаги окончания сессии
event.session.id	keyword	Нет	Идентификатор сессии
event.session.starttime	datetime_iso	Нет	Время начала сессии
event.severity	float	Да	Severity события, получаемое из заголовка Syslog, по умолчанию: 0
event.socket.protocol	keyword	Нет	Протокол транспортного уровня, например, TCP, UDP
event.subcategory	keyword	Да	Подкатегория события
event.timestamp	datetime_iso	Да	Время, в которое произошло событие
event.tls.fingerprint	keyword	Нет	TLS Certificate Fingerprint
event.tls.issuerdn	text	Нет	TLS Certificate Issuer DN
event.tls.not-after	datetime_iso	Нет	TLS Certificate date validation
event.tls.not-before	datetime_iso	Нет	TLS Certificate date validation
event.tls.sni	domain	Нет	TLS Certificate SNI
event.tls.subject	text	Нет	TLS Certificate Subject
event.uuid	keyword	Нет	UUID события
event.worker.host	keyword	Нет	
event.worker.ip	keyword	Нет	
initiator.antivirus.scan.endtime	datetime_iso	Нет	Время окончания антивирусного сканирования
initiator.antivirus.scan.starttime	datetime_iso	Нет	Время начала антивирусного сканирования
initiator.antivirus.scan.type	keyword	Нет	Тип антивирусного сканирования, например: On-Access, Schedule Scan, Quick Scan, Custom Scan...
initiator.command.executed	text	Нет	Выполненная команда
initiator.command.info	keyword	Нет	Информация о команде
initiator.command.path.original	keyword	Нет	
initiator.command.type	keyword	Нет	Тип конманды
initiator.file.hash.md5	keyword	Нет	MD5-хеш файла
initiator.file.hash.sha1	keyword	Нет	SHA1-хеш файла
initiator.file.hash.sha256	keyword	Нет	SHA256-хеш файла

Поле	Тип данных	Обязательно	Описание
initiator.geoip	geo	Нет	Данные GeoIP (автоматически проставляются подсистемой обработки событий)
initiator.host.fqdn	[domain]	Нет	FQDN инициатора
initiator.host.hostname	[domain]	Нет	Hostname инициатора
initiator.host.ip	[ip]	Нет	IP инициатора
initiator.http.method	keyword	Нет	https://wiki.squid-cache.org/SquidFaq/SquidLogs#Request_methods
initiator.http.user-agent	user_agent	Нет	HTTP User Agent
initiator.interface.mac	mac	Нет	MAC-адрес инициатора
initiator.interface.name	keyword	Нет	Имя интерфейса инициатора
initiator.nat.ip	ip	Нет	NAT IP-адрес
initiator.nat.port	port	Нет	NAT порт
initiator.process.command	text	Нет	Выполненная команда
initiator.process.guid	keyword	Нет	GUID-процесса
initiator.process.id	keyword	Нет	ID-процесса
initiator.process.hash.impash	keyword	Нет	impash-процесса
initiator.process.hash.md5	keyword	Нет	md5-процесса
initiator.process.hash.sha1	keyword	Нет	sha1-процесса
initiator.process.hash.sha256	keyword	Нет	sha256-процесса
initiator.process.parent.id	keyword	Нет	
initiator.process.hash.path.original	keyword	Нет	
initiator.process.path.drive	keyword	Нет	Диск, на котором запущен процесс C:, \ (сетевой каталог)
initiator.process.path.extension	keyword	Нет	Расширение запущенного файла
initiator.process.path.full	path	Нет	Полный путь до запущенного файла e.g. C:\Windows\System32\service.exe, \radarservices\company\secret.txt
initiator.process.path.name	keyword	Нет	Имя процесса, например: service, secret
initiator.process.path.original	keyword	Нет	Оригинальное имя процесса
initiator.process.path.path	path	Нет	Каталог в котором запущен процесс
initiator.process.working-directory	path	Нет	Рабочий каталог процесса
initiator.registry.path.original	keyword	Нет	
initiator.session.id	keyword	Нет	Logon-сессия связанная с инициатором (Windows: SubjectLogonID)
initiator.shell.name	keyword	Нет	Название shell
initiator.shell.version	keyword	Нет	Версия shell
initiator.socket.port	port	Нет	Порт инициатора
initiator.user.domain	keyword	Нет	Домен, которому принадлежит пользователь-инициатор
initiator.user.group.id	keyword	Нет	ID группы пользователя-инициатора
initiator.user.group.name	keyword	Нет	Имя группы пользователя-инициатора
initiator.user.id	keyword	Нет	ID пользователя, например SID или UID
initiator.user.name	keyword	Нет	Имя пользователя-инициатора
initiator.user.privileges.code	[keyword]	Нет	

Поле	Тип данных	Обязательно	Описание
initiator.user.privileges.description	[keyword]	Нет	
initiator.user.privileges.name	[keyword]	Нет	
initiator.user.privileges.original	[keyword]	Нет	
initiator.user.subcategory	text	Нет	
initiator.vpn.host.ip	[ip]	Нет	IP адрес, назначенный VPN-сервером
observer.blacklist	blacklist	Нет	
observer.event.id	keyword	Нет	ID-события
observer.event.type	keyword	Нет	Тип события Windows Channel
observer.file.hash.md5	keyword	Нет	MD5-хеш файла
observer.file.hash.sha1	keyword	Нет	SHA1-хеш файла
observer.file.hash.sha256	keyword	Нет	SHA256-хеш файла
observer.socket.port	port	Нет	Порт обзервера
observer.host.fqdn	[domain]	Нет	FQDN обзервера
observer.host.hostname	[domain]	Нет	Hostname обзервера
observer.host.ip	[ip]	Нет	IP обзервера
observer.interface.in.mac	mac	Нет	MAC-адрес интерфейса, на который получено событие
observer.interface.in.name	keyword	Нет	Имя интерфейса, на который получено событие
observer.interface.out.mac	mac	Нет	MAC-адрес интерфейса, с которого отправлено событие
observer.interface.out.name	keyword	Нет	Имя интерфейса, с которого отправлено событие
observer.rule.category	keyword	Нет	Категория правила, например в Suricata "Potentially Bad Traffic", "Misc Attack"
observer.rule.id	keyword	Нет	ID правила, по которому сгенерировалось событие, например: Suricata SID, Firewall rule ID
observer.rule.metadata.affected-product	keyword	Нет	Приложение, подверженное атаке
observer.rule.metadata.attack-target	keyword	Нет	Тип атакуемой цели
observer.rule.metadata.deployment	[keyword]	Нет	Тип развертывания
observer.rule.metadata.malware-family	keyword	Нет	Семейство вредоносного кода, обнаруживаемое правилом
observer.rule.name	keyword	Нет	Наименование правила
observer.rule.original	text	Нет	Исходный текст правила
observer.rule.threshold.count	integer	Нет	Сработавший порог по количеству для правила
observer.rule.threshold.seconds	integer	Нет	Сработавший порог по времени для правила
observer.rule.threshold.track	keyword	Нет	
observer.rule.threshold.type	keyword	Нет	
observer.zone.in.name	keyword	Нет	Имя сетевой зоны (inbound)
observer.zone.out.name	keyword	Нет	Имя сетевой зоны (outbound)
outcome.description	text	Нет	Описание результата
outcome.name	keyword	Нет	Нормализованное представление результата

Поле	Тип данных	Обязательно	Описание
outcome.original	keyword	Нет	Вендор-специфичное представление для результата
raw	raw_text	Нет	Изначальное событие
executed.description	text	Нет	Описание реакции на событие
reaction.executed.name	keyword	Нет	Нормализованное представление реакции на событие
reaction.executed.original	keyword	Нет	Вендор-специфичное представление реакции на событие
reaction.executed.reason	keyword	Нет	Описание причины применения указанной реакции на событие
reaction.executed.user.domain	keyword	Нет	Домен пользователя
reaction.executed.user.id	keyword	Нет	ID пользователя
reaction.executed.user.name	keyword	Нет	Логин пользователя
reaction.requested.description	text	Нет	Описание требуемой реакции
reaction.requested.name	keyword	Нет	Нормализованное представление требуемой реакции
reaction.requested.original	keyword	Нет	Вендор-специфичное представление требуемой реакции
reaction.requested.reason	keyword	Нет	Описание причин требуемой реакции
reportchain.collector.host.fqdn	[domain]	Да	FQDN модуля Платформы Радар получившего событие
reportchain.collector.host.hostname	[domain]	Нет	Hostname модуля Платформы Радар, получившего событие
reportchain.collector.host.ip	[ip]	Нет	IP модуля Платформы Радар, получившего событие
reportchain.collector.timestamp	datetime_iso	Да	Время получения события Платформой Радар
reportchain.relay.host.fqdn	[domain]	Нет	FQDN хоста, отправившего событие по syslog
reportchain.relay.host.hostname	[domain]	Нет	Hostname хоста, отправившего событие по syslog
reportchain.relay.host.ip	[ip]	Нет	IP хоста, отправившего событие по syslog
reportchain.relay.timestamp	datetime_iso	Нет	Отметка времени получения события хостом с NxLog (временная отметка агента)
tags	[keyword]	Нет	Тэги
target.auth.encryption	keyword	Нет	Ticket Encryption Type
target.access_mask.original	keyword	Нет	
target.auth.options.name	keyword	Нет	Ticket Options
target.auth.options.original	keyword	Нет	
target.auth.process.name	keyword	Нет	Процесс, выполняющий аутентификацию sshd, Schannel, Advapi (LogonProcessName in Windows)
target.auth.service.domain	keyword	Нет	
target.auth.service.id	keyword	Нет	
target.auth.service.name	keyword	Нет	Наименование сервиса в Kerberos Realm, которому был отправлен TGT-запрос
target.command.executed	text	Нет	Выполненная команда
target.command.path.original	keyword	Нет	Путь до запущенного процесса

Поле	Тип данных	Обязательно	Описание
target.config.changes.description	[keyword]	Нет	Тип изменений в конфигурации
target.config.changes.id	[keyword]	Нет	ID изменений в конфигурации
target.database.name	keyword	Нет	Название БД
target.email.file.drive	[path]	Нет	Информация о email-аттаче
target.email.file.extention	[keyword]	Нет	Информация о email-аттаче
target.email.file.fullname	[keyword]	Нет	Информация о email-аттаче
target.email.file.name	[keyword]	Нет	Информация о email-аттаче
target.email.file.path	[path]	Нет	Информация о email-аттаче
target.email.receivers	[keyword]	Нет	Email-адреса получатели письма
target.email.sender	keyword	Нет	Email-адрес отправителя
target.email.subject	text	Нет	Тема письма
target.email.url.full	[keyword]	Нет	URL в письме
target.email.url.host.fqdn	[domain]	Нет	URL в письме
target.email.url.host.hostname	[domain]	Нет	URL в письме
target.email.url.host.ip	[ip]	Нет	URL в письме
target.file.content-type	text	Нет	Content-Type файла
target.file.drive	path	Нет	Диск, на котором находится файл
target.file.extension	keyword	Нет	Расширение файла
target.file.fullname	keyword	Нет	Полное имя файла
target.file.hash.md5	keyword	Нет	MD5-хеш файла
target.file.hash.sha1	keyword	Нет	SHA1-хеш файла
target.file.hash.sha256	keyword	Нет	SHA256-хеш файла
target.file.name	keyword	Нет	Имя файла
target.file.path	path	Нет	Полный путь до файла
target.file.size	integer	Нет	Размер файла (в байтах)
target.host.geoip	geo	Нет	Данные GeoIP (автоматически проставляются подсистемой обработки событий)
target.group.domain	keyword	Нет	Домен группы
target.group.id	keyword	Нет	ID группы
target.group.name	keyword	Нет	Имя группы
target.host.fqdn	[domain]	Нет	FQDN хоста
target.host.hostname	[domain]	Нет	Hostname
target.host.ip	[ip]	Нет	IP-адрес хоста
target.http.content-type	keyword	Нет	Content type ответа, например, text/html
target.http.redirect.host.fqdn	[domain]	Нет	Host part of the redirected URL
target.http.redirect.host.hostname	[domain]	Нет	Host part of the redirected URL
target.http.redirect.host.ip	[ip]	Нет	Host part of the redirected URL
target.http.redirect.path	[text]	Нет	Path of the redirected URL http://hostname.tld:port/path
target.http.redirect.port	[port]	Нет	Port of the redirected URL
target.http.redirect.protocol	[keyword]	Нет	Protocol of the redirected URL (http or https)
target.http.referer.host.fqdn	[domain]	Нет	Host part of the referer URL

Поле	Тип данных	Обязательно	Описание
target.http.referer.host.hostname	[domain]	Нет	Host part of the referer URL
target.http.referer.host.ip	[ip]	Нет	Host part of the referer URL
target.http.referer.path	[text]	Нет	Path of the referer URL http://hostname.tld:port/path
target.http.referer.port	[port]	Нет	Port of the referer URL
target.http.referer.protocol	[keyword]	Нет	Protocol of the referer URL (http or https)
target.http.status.code	integer	Нет	https://wiki.squid-cache.org/SquidFaq/SquidLogs#HTTP_status_codes
target.http.status.description	text	Нет	https://wiki.squid-cache.org/SquidFaq/SquidLogs#HTTP_status_codes
target.http.status.name	keyword	Нет	https://wiki.squid-cache.org/SquidFaq/SquidLogs#HTTP_status_codes
target.http.url.host.fqdn	[domain]	Нет	Host part of the targeted URL
target.http.url.host.hostname	[domain]	Нет	Host part of the targeted URL
target.http.url.host.ip	[ip]	Нет	Host part of the targeted URL
target.http.url.path	[text]	Нет	Path of the targeted URL http://hostname.tld:port/path
target.http.url.port	[port]	Нет	Port of the targeted URL
target.http.url.protocol	[keyword]	Нет	Protocol of the targeted URL (http or https)
target.interface.mac	mac	Нет	MAC-адрес интерфейса
target.interface.name	keyword	Нет	Имя интерфейса
target.nat.ip	ip	Нет	NAT IP-адрес
target.nat.port	port	Нет	NAT порт
target.object.attribute.name	keyword	Нет	Атрибут объекта, который был модифицирован
target.object.attribute.value	text	Нет	Значение атрибута
target.object.domain	keyword	Нет	Домен объекта, который был модифицирован
target.object.id	keyword	Нет	ID объекта, который был модифицирован
target.object.name	keyword	Нет	Имя объекта, который был модифицирован
target.object.type	keyword	Нет	Класс объекта, который был модифицирован
target.object.server	keyword	Нет	Сервер объекта, который был модифицирован
target.object.handle.id	keyword	Нет	
target.permissons.granted.name	[keyword]	Нет	
target.permissons.granted.original	[keyword]	Нет	
target.permissons.requested.description	[keyword]	Нет	
target.permissons.requested.name	[keyword]	Нет	
target.permissons.requested.original	[keyword]	Нет	
target.policy.category.description	text	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4719
target.policy.category.id	keyword	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4719
target.policy.changes.description	[text]	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4719

Поле	Тип данных	Обязательно	Описание
target.policy.changes.id	[keyword]	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4719
target.policy.subcategory.description	text	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4719
target.policy.subcategory.id	keyword	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4719
target.process.args	keyword	Нет	Аргументы
target.process.command	text	Нет	Выполненная команда
target.process.guid	keyword	Нет	GUID процесса
target.process.hash.impash	keyword	Нет	impash-хеш файла
target.process.hash.md5	keyword	Нет	MD5-хеш файла
target.process.hash.sha1	keyword	Нет	SHA1-хеш файла
target.process.hash.sha256	keyword	Нет	SHA256-хеш файла
target.process.id	keyword	Нет	ID процесса
target.process.integrity.description	keyword	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4688
target.process.integrity.id	keyword	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4688
target.process.integrity.id-hex	keyword	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4688
target.process.integrity.name	keyword	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4688
target.process.path.drive	keyword	Нет	Диск на котором запущен процесс C:, \ (сетевой каталог)
target.process.path.extension	keyword	Нет	Расширение запущенного файла
target.process.path.full	path	Нет	Полный путь до запущенного файла e.g. C:\Windows\System32\service.exe, \radarservices\company\secret.txt
target.process.path.name	keyword	Нет	Имя процесса, например: service, secret
target.process.path.original	text	Нет	Оригинальное имя процесса
target.process.path.path	path	Нет	Каталог, в котором запущен процесс
target.process.path.file.internal.name	keyword	Нет	Имя файла
target.process.privileges.code	keyword	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4688
target.process.privileges.description	[keyword]	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4688
target.process.privileges.original	[keyword]	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4688
target.process.privileges.description	[keyword]	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4688
target.process.working-directory	path	Нет	
target.registry.path.original	keyword	Нет	
target.instance.name	keyword	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4688
target.rule.action	keyword	Нет	Действие, выполненное на межсетевом экране: allow, bypass, deny, log only, discard/reject

Поле	Тип данных	Обязательно	Описание
target.rule.chain	keyword	Нет	Windows: inbound или outbound, Linux: цепочка iptables
target.rule.dst-addresses	[keyword]	Нет	IP-адрес в правиле межсетевого экрана
target.rule.dst-ports	[keyword]	Нет	Порт в правиле межсетевого экрана
target.rule.id	keyword	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4946
target.rule.name	keyword	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4946
target.rule.profiles	[keyword]	Нет	https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4946
target.rule.src-addresses	[keyword]	Нет	IP-адрес источника в правиле межсетевого экрана
target.rule.src-ports	[keyword]	Нет	Порт источника в правиле межсетевого экрана
target.rule.status	keyword	Нет	Индикатор активности правила
target.service.name	keyword	Нет	Название сервиса
target.path.original	keyword	Нет	
target.service.start_type.new.description	keyword	Нет	
target.service.start_type.new.name	keyword	Нет	
target.service.start_type.new.original	keyword	Нет	
target.service.start_type.old.description	keyword	Нет	
target.service.start_type.old.name	keyword	Нет	
target.service.start_type.old.original	keyword	Нет	
target.service.status.name	keyword	Нет	
target.service.status.original	keyword	Нет	
target.service.type.description	keyword	Нет	
target.service.type.name	keyword	Нет	
target.service.type.original	keyword	Нет	
target.session.id	keyword	Нет	ID сессии (Windows: TargetLogonID)
target.share.local_path.original	keyword	Нет	
target.share.relative_name.original	keyword	Нет	
target.share.remote_path.original	keyword	Нет	
target.shell.name	keyword	Нет	
target.shell.version	keyword	Нет	Версия shell
target.syscall.id	keyword	Нет	syscall id
target.syscall.name	keyword	Нет	Системный вызов
target.task.args	keyword	Нет	Аргументы выполнения
target.task.auth.method.name	keyword	Нет	Метод аутендификации
target.task.command	keyword	Нет	
target.task.description	text	Нет	Описание
target.task.name	keyword	Нет	Имя системного вызова
target.task.privileges.name	keyword	Нет	
target.task.privileges.original	keyword	Нет	
target.task.status.name	keyword	Нет	

Поле	Тип данных	Обязательно	Описание
target.task.status.original	keyword	Нет	
target.task.status.visibility.original	keyword	Нет	
target.task.status.working-directory	keyword	Нет	Рабочая директория
target.socket.port	port	Нет	Порт
target.threat.category	keyword	Нет	Категория угрозы, например: Potentially Unwanted Software
target.threat.confidence	keyword	Нет	Уровень доверия результату детектирования
target.threat.content-type	keyword	Нет	Content type угрозы: data, file, packet, url
target.threat.description	text	Нет	Описание угрозы
target.threat.detection_delta	integer	Нет	Окно реагирования
target.threat.origin.name	keyword	Нет	
target.threat.origin.original	keyword	Нет	
target.threat.severity	keyword	Нет	Уровень угрозы
target.threat.status.original	keyword	Нет	
target.threat.name	keyword	Нет	Наименование угрозы PUA:Win32/FustionCore
target.user.category	text	Нет	Категория пользователься
target.user.delegations	[keyword]	Нет	Windows: AllowedToDelegateTo
target.user.description	keyword	Нет	Описание пользователя
target.user.domain	keyword	Нет	Домен пользователя
target.user.group.id	keyword	Нет	ID группы пользователя
target.user.group.name	keyword	Нет	Имя группы пользователя
target.home.path.original	keyword	Нет	Путь к домашней дирректории
target.user.id	keyword	Нет	ID пользователя, например, SID или UID
target.user.id-history	[keyword]	Нет	Windows: SidHistory
target.user.name	keyword	Нет	Имя пользователя
target.user.primary-group	keyword	Нет	Windows: PrimaryGroupId
target.user.privileges.code	[keyword]	Нет	
target.user.privileges.description	[keyword]	Нет	
target.user.privileges.name	[keyword]	Нет	
target.user.privileges.original	[keyword]	Нет	
target.user.spn.delegators	[keyword]	Нет	
target.user.spn.names	[keyword]	Нет	
target.user.subcategory	text	Нет	
target.user.uac.attribute.new-value	keyword	Нет	
target.user.uac.attribute.old-value	keyword	Нет	
target.user.uac.status	[keyword]	Нет	

13. Описание специальных функций

В случае необходимости дополнительной обработки данных перед процессом нормализации можно воспользоваться функциями и операторами, позволяющими выполнять сложные операции прямо на странице настройки правила нормализации. Эти операции компилируются непосредственно в байт-коде Python.

Для корректного распознавания логического выражения используйте перенос | и описывайте выражение с новой строки.

По умолчанию все поля, которые указывают при работе с функциями и операторами в рамках дополнительной обработки данных, являются обязательными. Однако в поступающих данных указанные поля иногда могут не присутствовать. И чтобы не возникало ошибки, можно пометить эти поля как необязательные, отметив это в настройке правила нормализации или добавив в описание функции строку "**required: false**". В таком случае, поле будет обработано и выведено далее, если оно присутствует во входящих данных.

13.1. Строковые функции

13.1.1. Преобразование к нижнему регистру (lower)

Преобразование поля или определенной строки к нижнему регистру.

Использование функции: **lower**(**string**), где **string** — строка, преобразуемая к нижнему регистру.
Пример:

```
my_section.my_field:
  field: lower(hostname)
```

13.1.2. Преобразование к верхнему регистру (upper)

Преобразование поля или определенной строки к верхнему регистру.

Использование функции: **upper**(**string)**, где **string** — строка, преобразуемая к верхнему регистру.
Пример:

```
my_section.my_field:
  field: upper(software_name)
```

13.1.3. Удаление элементов из строки (strip)

Функция убирает из строки все элементы, указанные перечислением в необязательном первом аргументе. Если указан только один — второй — аргумент, то будут удалены только пробелы.

Использование функции: **strip**("**strip_chars**", **string**), где **string** — строка, из которой необходимо убрать перечисленные символы **strip_chars**.
Пример использования функции для удаления пробелов:

```
section.stripped_field:
  field: strip(messy_string)
```

Пример использования функции для удаления запятой:

```
section.stripped_field_comma:
  field: strip(",", messy_comma_string)
```

Пример использования функции для удаления различных знаков препинания:

```
section.stripped_field_multiple_possible:
  field: strip(",\'.", messy_multiple_possible_string)
```

13.1.4. Разбиение строки (split)

Функция разделяет строку по указанному разделителю и возвращает её в виде списка.

Использование функции: **split**(**string**, **separator**)[**index**], где **string** — строка, которую необходимо преобразовать, **separator** — разделитель, а **index** — индекс требуемого элемента (допускается использование отрицательного индекса как в Python). [0], [1] и т.д. — первый, второй и т.д. элементы с начала строки, [-1] — первый элемент с конца строки.
Пример:

```
section.proto_name:
  field: split(http.protocol, '/')[0]
```

Пример использования функции для вывода элемента первого с конца:

```
section.other_proto_name:
  field: split(http.other_name, ',')[-1]
```

13.1.5. Проверка по регулярному выражению (match)

Функция возвращает **true**, если строка соответствует заданному регулярному выражению.

Использование функции: **match**('**regular_expression**', **string**), где **string** — строка, которую необходимо проверить на соответствие, **regular_expression** — регулярное выражение.
Пример:

```
section.is_expected_code:
  required: false
  field: match('(1..|2..|418)', str(http.status))
```

Подробнее про **required: false** можно прочитать в начале раздела.

13.1.6. Замена строки (replace)

Функция выполняет замену в строке, возвращая новую строку с проведенной заменой.

Использование функции: **replace**(**string**, **old_value**, **new_value**), где **string** — строка, в которой необходимо произвести замену, **old_value** — заменяемое значение, **new_value** — новое значение.
Пример замены немецкого написания слова "benutzer" на английский "user":

```
section.user_info:
  field: replace(line.full_user_name, 'benutzer', 'user')
```

13.2. Логические операторы

Инфиксные операторы также доступны внутри нормализаторов YAML. В этом разделе доступны почти все операторы Python.

Логические операторы возвращают **true** или **false** в зависимости от выражения.

13.2.1. Логическое НЕ (not)

Оператор **not** возвращает **true**, если поле не соответствует заданному значению, иначе **false**.

Пример:

```
section.is_not_using_firefox:
  field: not browser_name == 'Firefox'
```

13.2.2. Равенство (==)

Оператор **==** возвращает **true**, если оба операнда равны, иначе **false**.

Пример:

```
section.is_using_firefox:
  field: software_name == 'Firefox'
```

13.2.3. Неравенство (!=)

Оператор **!=** возвращает **true**, если оба операнда различны, иначе **false**.

Пример:

```
section.is_not_1_3:
  field: version != 1.3
```

13.2.4. Больше (>)

Оператор **>** возвращает **true**, если один операнд больше другого, иначе **false**.

Пример:

```
section.is_newer_than_1_0:
  field: version > 1.0
```

13.2.5. Больше или равно (>=)

Оператор **>=** возвращает **true**, если один операнд больше или равен другому, иначе **false**.

Пример:

```
section.is_at_least_1_0:
  field: version >= 1.0
```

13.2.6. Меньше

Оператор **<** возвращает **true**, если один операнд меньше другого, иначе **false**.

Пример:

```
section.is_prior_to_1_0:
  field: version < 1.0
```

13.2.7. Меньше или равно

Оператор **<=** возвращает **true**, если один операнд меньше или равен другому, иначе **false**.

Пример:

```
section.is_prior_or_1_0:
  field: version <= 1.0
```

13.2.8. Логическое И (and)

Оператор **and** объединяет условия между собой. Если все выражения оцениваются как **true**, то возвращается **true**, если хотя бы одно — **false**, то возвращается **false**.

Использование оператора: **bool_expr_1 and bool_expr_2**, где **bool_expr** — логическое выражение. Допускается использование более двух логических выражений.
Пример:

```
section.is_firefox_and_windows:
  field: browser_name == 'Firefox' and os_name == 'Windows'
```

13.2.9. Логическое ИЛИ (or)

Оператор **or** возвращает значение **true**, если хотя бы одно из выражений оценивается как **true**, в ином случае — **false**.

Использование оператора: **bool_expr_1 or bool_expr_2**, где **bool_expr** — логическое выражение. Допускается использование более двух логических выражений.
Пример:

```
section.is_firefox_or_windows:
  field: browser_name == 'Firefox' or os_name == 'Windows'
```

13.2.10. Проверка наличия элемента (in)

Оператор **in** проверяет вхождение элемента в массив значений. Функция также работает для проверки вхождения подстроки в строку.

Использование оператора: **variable in (value_1, value_2, value_3)**, где **variable** — переменная, **value** — значение.
Пример:

```
section.is_firefox:
  field: |
     'Firefox' in http.user_agent
```

> **Важно!** Используйте перенос | для корректного распознавания логического выражения

13.3. Арифметические операторы

13.3.1. Умножение (*)

Оператор * умножает два операнда.

Пример:

```
section.total_cpu_freq:
  field: cpu_number * frequency
```

13.3.2. Деление (/)

Оператор **/** делит первый операнд на второй.

Пример:

```
section.division:
  field: first_value / second_value
```

13.3.3. Сложение (+)

Оператор **+** суммирует два операнда.

Пример:

```
section.sum:
  field: first_value + second_value
```

13.3.4. Вычитание (-)

Оператор **-** вычитает из первого операнда второй операнд.

Пример:

```
section.difference:
  field: first_value - second_value
```

13.4. Условные конструкции

13.4.1. cond

Функция **cond** работает как оператор **if/else**. Если указанное в первом аргументе логическое выражение оценивается как **true**, то выводится второй аргумент; если **false** - третий.

Использование функции: **cond**(**bool_expr**, ‘**Значение, если истина**’, ‘**Значение, если ложь**’), где **bool_expr** — логическое выражение.
Пример:

```
section.browser_hint:
  field: |
    cond(browser_name == 'Firefox', 'Firefox detected',
         'Other browser detected')
```

> **Важно!** Используйте перенос | для корректного распознавания логического выражения

Функцию **cond** можно использовать как переключатель и описывать более сложные случаи.

Использование функции: **cond**(**bool_expr**, ‘**Значение, если истина**’, **another_bool_expr**, '**Значение, если истина**', '**Значение по умолчанию**'), где **bool_expr** — логическое выражение.
Пример:

```
section.firewall_status: |
  cond(type == 'utm', 'Suspicious activity was detected',
       action == 'close', 'A connection was closed',
       action == 'start', 'A connection was started',
       'A connection was allowed')
```

> **Важно!** Используйте перенос | для корректного распознавания логического выражения

Пример без указания значения по умолчанию:

```
reason.type: |
  cond(action == 'reset', 'flow/reset',
       action == 'deny', 'flow/deny')
  required: false
```

> **Важно!** Используйте перенос | для корректного распознавания логического выражения

Если значение по умолчанию отсутствует, поле пропускается. В этом случае необходимо отметить в форме настройки правила нормализации рядом с соответствующим полем, что оно необязательное или добавить в поле ввода "**required: false**", иначе будет ошибка.
Если поле является необязательным, а условие ссылается на входную переменную, которая отсутствует, условие будет считаться ложным.
Если условие истинно, но в значении отсутствует поле ввода, это поле будет удалено из вывода.

Подробнее про используемый в примере **required: false** можно прочитать в начале раздела.

13.4.2. optional

Функция проверяет, присутствуют ли все указанные поля, если нет — возвращает значение по умолчанию (или **false**).

Пример:

```
outcome:
  field: |
    cond(optional(tcp.rst, false), 'failed',
         optional(tcp.fin, false), 'success',
         'pending')
```

> **Важно!** Используйте перенос | для корректного распознавания логического выражения

Если выражения относятся к нескольким полям, все они должны присутствовать. Следующий пример вернёт **NaN**, если a, b или c не присутствуют в проанализированных данных. "**NaN**" указывается, если данные отсутствуют, не существуют.
Пример:

```
sum:
  field: optional(a + b + c, float('nan'))
```

13.5. Поиск данных

Массивы, которые используются в нескольких нормализаторах, размещаются в **lookups.yaml**. Это специальный файл, содержащий только глобальные поисковые запросы, доступные в каждом нормализаторе.

Необходимо убедиться, что каждый массив имеет уникальное имя.

13.5.1. lookup

Функция **lookup** работает как поиск значений по ключу. Значения, содержащиеся в массивах, доступны только с помощью этой функции.
Допустим, в "Таблицах просмотра" определен следующий массив с названием "protos":

```
lookup:
  protos:
    0: NotSecureProtocol
    1: SecureProtocol
    2: VerySecureProtocol
    3: Telnet
```

Тогда есть возможность получить доступ к этим значениям следующим образом, где **protocol_id** является полем события:

```
section.field:
  field: lookup('protos', proto_id)
```

Если ключ не содержится в словаре, анализ завершится неудачей. Чтобы избежать этого, можно указать возвращаемое значение по умолчанию на случай, если ключ не найден.
В примере, если **proto_id** не является допустимым ключом в **protos**, будет возвращено значение "**Unknown protocol**":

```
section.field:
  field: lookup('protos', proto_id, 'Unknown protocol')
```

13.5.2. exists

Функция **exists** проверяет, имеет ли поле полезное значение: не null, не пустую строку и не "-".

Возвращает **true** или **false**.

Пример:

```
section.user_data.is_user_set:
  field: exists(line.app.data.user)
```

Пример использования функции **exists** в сочетании с функцией **cond**:

```
section.system.app_name:
  field: cond(exists(line.app.name), line.app.name, "unknown application")
```

13.6. Преобразование типа данных

13.6.1. Строковый формат (str)

Преобразование значения поля в строковый формат.

Использование функции: **variable::str**, где **variable** — переменная.
Пример:

```
section.field_that_is_a_string:
  field: field_that_is_a_int::str
```

13.6.2. Формат целого числа (int)

Преобразование значения поля в формат целого числа.

Использование функции: **variable::int**, где **variable** — переменная.
Пример:

```
section.field_that_is_a_int:
  field: field_that_is_a_string::int
```

13.6.3. Формат числа с плавающей точкой (float)

Преобразование значения поля в формат числа с плавающей точкой.

Использование функции: **variable::float**, где **variable** — переменная.
Пример:

```
section.field_that_is_a_float:
  field: field_that_is_a_int::float
```

13.7. Функции проверки корректного представления данных

13.7.1. Проверка IP-адреса (is_ip)

Функция для определения, является ли предоставленная строка допустимым адресом IPv4 или IPv6.

Пример:

```
section.is_valid_ip:
  field: is_ip(string)
```

13.7.2. Проверка имени хоста (is_hostname)

Функция для определения, является ли предоставленная строка допустимым именем хоста. Она не должна быть пустой и содержать точки.

Пример:

```
section.is_valid_hostname:
  field: is_hostname(string)
```

13.7.3. Проверка доменного имени (is_fqdn)

Функция для определения, является ли предоставленная строка корректным доменным именем. Доменное имя должно содержать хотя бы одну точку, метки между точками не должны быть пустыми. Это не должен быть IP-адрес. Доменное имя может заканчиваться точкой.

Пример:

```
section.is_valid_fqdn:
  field: is_fqdn(string)
```

13.8. Функции для работы со временными отметками

13.8.1. Приведение к ISO 8601 (parse_timestamp)

Функция выполняет перебор всех указанных в качестве аргументов форматов временной отметки и пытается разобрать строку **my_ts**. Функция перебирает форматы временной отметки до тех пор, пока метка времени не будет проанализирована и возвращена в виде строки в формате ISO 8601.

Форматы должны быть строковыми константами. Допустимые форматы: «iso8601» и все директивы синтаксического анализа, поддерживаемые функцией Python strptime.

Использование функции: **parse_timestamp**(**my_ts**, **format1**[, **format2**, **format3**...]), где **my_ts** — отметка времени, **format** — формат временной отметки.
Пример:

```
"@timestamp":
  field: parse_timestamp(
                        date + ' ' + time,
                        '%m/%d/%Y %I:%M:%S %p',
                        '%Y/%m/%d',
                        'iso8601'
                        )
```

> **Важно!** Синтаксический анализ временных меток с помощью функции **parse_timestamp** довольно медленный. Рекомендуется для создания временной метки ISO 8601 в первую очередь использовать простые строковые операции, и, только в случае невозможности этого, использовать функцию **parse_timestamp**.

13.8.2. Приведение к Unix time (timestamp_to_epoch)

Функция принимает временную метку ISO и преобразует ее в секунды, начиная с временной метки эпохи, в виде числа с плавающей точкой. Если в необработанной строке журнала присутствует **tzinfo** (информация о смещении времени от времени UTC, о переходе на летнее время и проч), то это значение будет использоваться для локализации отметки времени перед преобразованием.

Использование функции: **timestamp_to_epoch**(**my_ts**), где **my_ts** — отметка времени.
Пример:

```
section.since_epoch:
  field: timestamp_to_epoch(ts)
```

13.8.3. Приведение к UTC (epoch_to_timestamp)

Функция принимает временную метку эпохи в секундах и преобразует ее во временную метку UTC.

Использование функции: **epoch_to_timestamp**(**my_epoch**)
Пример:

```
section.date:
  field: epoch_to_timestamp(epoch)
```

13.9. Функции для дополнительной нормализации

13.9.1. Нормализация User Agent (normalize_http_user_agent)

Функция обращается к строке User agent и производит её дополнительный разбор по следующим полям:

- **full** — содержимое строки User Agent
- **name** — название браузера
- **os** — семейство и версия операционной системы
- **device** — устройство
- **major** — мажорная версия браузера
- **minor** — минорная версия браузера

Использование функции: **normalize_http_user_agent**(**string**), где **string** — строка, которую необходимо преобразовать.
Пример использования функции.
Событие:

```
{"src_ip":"10.10.10.10", "dst_ip":"20.20.20.20", "cs_user_agent":"Mozilla/5.0
(X11; Linux x86_64;rv:60.0) Gecko/20100101 Firefox/60.0"}
```

Нормализатор:

```
section.user_agent:
  field: normalize_http_user_agent(cs_user_agent)
```

Результат:

```
"section": {
"user-agent": {
"device": "Other",
"full": "Mozilla/5.0 (X11; Linux x86_64;rv:60.0) Gecko/20100101 Firefox/60.0",
"major": 60,
"minor": 0,
"name": "Firefox",
"os": "Linux"
}
```

13.9.2. Нормализация MAC-адреса (normalize_mac_address)

Функция имеет один обязательный аргумент (MAC-адрес) и необязательный — второй — аргумент. Второй аргумент имеет логический тип данных и по умолчанию **true**. Этот аргумент определяет поведение в случае неверного MAC-адреса (по умолчанию строка журнала отправляется в Index Error).

Поддерживаются следующие форматы:

- AA-BB-CC-DD-EE-FF
- AAA.BBB.CCC.DDD
- AAA:BBB:CCC:DDD
- AAA-BBB-CCC-DDD
- AAABBBCCCDDD

Если MAC-адрес действителен, функция преобразует его в стандартный формат **AA:BB:CC:DD:11:22**.
Например, MAC-адрес формата **FF-BA-CD-1D-32-11** функция преобразует в формат **FF:BA:CD:1D:32:11**.

Если MAC-адрес недействителен, а второй аргумент **true** (по умолчанию), строка будет отправлена в Index Error. Если второй аргумент **False**, то будет возвращена пустая строка, а для события **event.anomaly.malformed_mac_address** будет задана нормализованная строка журнала.

Использование функции: **normalize_mac_address**(**mac_address**)

Пример обработки события с действительным MAC-адресом и без указания второго аргумента.
Событие:

```
{"src_ip":"10.10.10.10", "mac_address":"AA-BB-CC-DD-EE-FF"}
```

Нормализатор:

```
section.client_mac:
  field: normalize_mac_address(mac_address)
```

Результат:

```
"section": {
   "client_mac": "AA:BB:CC:DD:EE:FF"
}
```

Пример обработки события с недействительным MAC-адресом и **false** в качестве второго аргумента.

```
section.client_mac:
  field: normalize_mac_address("AA:BB:CC", false)
```

Результат:

```
{
    "event": {
        "anomaly": {
            "malformed_mac_address": [
                "AA:BB:CC"
            ]
        }
    },
    "section": {
        "client_mac": ""
    }
}
```

13.9.3. Нормализация данных по хосту (normalize_host)

Функция предназначена для корректного формирования информации о хосте. Принимает на вход ряд полей и возвращает в виде словаря с тремя ключами: **IP**, **FQDN**, **Hostname**, где **IP** — массив IP-адресов, **FQDN** — массив доменных имен, **Hostname** — массив имен хостов.

Использование функции: **normalize_host**(**field1** [, **field2**, **field3**, ... , **fieldN**]), где **field** — поле. Пример:

```
target.host:
  field: normalize_host('127.0.0.1','lt-mail','lt-mail.domain','','10.0.0.2')
```

Результат:

```
"target": {
"host": {
"fqdn": ["lt-mail.domain"],
"hostname": ["lt-mail"],
"ip": ["10.0.0.2","127.0.0.1"]
}
}
```

13.9.4. Нормализация данных URL (normalize_url)

Функция разбивает URL-адрес на составляющие и возвращает в виде словаря. Второй аргумент является необязательным, в случае его отсутствия значением по умолчанию является пустая строка.

Пример использования функции:

```
url:
  field: normalize_url(field, type)
```

Пример результата:

```
"url": {
    'protocol': 'http',
    'host': {'hostname': ['pangeoradar.ru'], 'ip': [], 'fqdn': []},
    'path': '/',
    'params': '',
    'username': '',
    'password': '',
    'port': 80,
    'query': '',
    'fragment': '',
    'original': 'https://pangeoradar.ru/',
    'source-type': 'something',
}
```

Где:

- **protocol** — протокол
- **host** — структура {'hostname': [], 'ip': [], 'fqdn': []}, в которую передается Hostname, IP или FQDN
- **path** — путь
- **params** — параметры
- **username** — имя пользователя
- **password** — пароль
- **port** — порт
- **query** — запрос
- **fragment** — фрагмент страницы
- **original** — оригинальный URL, переданный в функцию
- **source-type** — тип источника

Событие:

```
{"src_ip":"10.10.10.10", "url":"http://user:pass@NetLoc:80/path;parameters?
query=argument#fragment"}
```

Нормализатор:

```
field: normalize_url(data['url'], 'url')
```

Результат:

```
"target": {
"http": {
"url": {
"fragment": "fragment",
"host": {"fqdn": [], "hostname": ["netloc"], "ip": []},
"original": "http://user:pass@NetLoc:80/path;parameters?
query=argument#fragment",
"params": "parameters",
"password": "pass",
"path": "/path",
"port": 80,
"protocol": "http",
"query": "query=argument",
"source-type": "",
"username": "user"
}
}
}
```

13.9.5. Нормализация данных Windows SID (normalize_windows_sid)

Функция принимает одно поле (Windows SID) в качестве входных данных и возвращает словарь с тремя ключами: **category**, **subcategory** и **desc**, где **category** — категория, **subcategory** — подкатегория и **desc** — описание.

Пример использования функции:

```
initiator.user.id_details:
    field: normalize_windows_sid(SubjectUserSid)

target.user.id_details:
    field: normalize_windows_sid(TargetUserSid)
```

Пример результата:

```
"initiator" : {
    "user" : {
        "id_details" : {
            "category" : "builtin_system_account",
            "subcategory" : "builtin_anonymous_account",
            "desc" : "ANONYMOUS LOGON"
        }
    }
}
```

Перед использованием этой функции в "Таблицах просмотра" необходимо описать массив «windows_sids». Он должен предоставить записи для случаев:

1. **sid** равен строке,
2. **sid** начинается с подстроки,
3. **sid** начинается с подстроки и заканчивается другой подстрокой,
4. **sid** начинается с подстроки и не заканчивается другой подстрокой.

Пример lookup, который охватывает все 4 варианта случаев. Будет взято первое совпадение:

```
lookup:
    windows_sids:
        - "sid": "S-1-5-7"
          "match_type": equal
          "category": builtin_system_account
          "subcategory": builtin_anonymous_account
          "desc": ANONYMOUS LOGON
        - "sid": "S-1-5-111-"
          "match_type": start
          "category": builtin_system_account
          "subcategory": builtin_virtual_sshd_account
          "desc": TBD
        - "sid": "S-1-5-21-"
          "match_type": start_end
          "ends":
            - "end": "-500"
              "category": standard_account
              "subcategory": builtin_virtual_sshd_account
              "desc": TBD
            - "end": "-501"
              "category": standard_account
              "subcategory": builtin_guest_account
              "desc": TBD
            - "not_end": "$"
              "category": standard_account
              "subcategory": standard_account
              "desc": TBD
```

Пример результата, если lookup без записей:

```
"initiator" : {
    "user" : {
        "id_details" : {
            "category" : "undefined_account_type",
            "subcategory" : "undefined_account_type",
            "desc" : "undefined_account_type"
        }
    }
}
```

13.10. Дополнительные функции

13.10.1. Tapping

Функция, которая помогает обрабатывать сложные непрогнозируемые данные на этапе предварительной обработки.

Пример использования функции:

```
tap: |
    tcp_flags = line.parsed['tcp']
    line.parsed['flow_tags'] = [
        f"tcp_{flag}"
        for flag in
            ("syn", "fin", "rst", "psh", "ack", "cwr", "ecn", "urg")
        if tcp_flags.get(flag, False)
    ]
```

14. Обогащение событий

В качестве источников обогащения событий в Платформе используются следующие типы обогащений:

- GeoIP
- DNS
- Threat Intelligence
- RVS
- Lookups

14.1. Настройка GeoIP обогащения

GeoIP обогащение работает на основе базы IP-адресов GeoLite от MaxMind's.

1. Для работы необходимо получить базу GeoLite2-City.mmdb и положить ее на экземпляр модуля обработки событий.(https://dev.maxmind.com/geoip/geolite2-free-geolocation-data?lang=en)
2. Далее в конфигурационном файле модуля обработки событий `/opt/pangeoradar/configs/termite/conf.yaml` необходимо добавить следующие записи:

```
geoip:
  enabled: true
  db-path: /etc/termite/GeoLite2-City.mmdb # путь до файла с базой ip-
адресов
```

3. Далее необходимо перезапустить сервис **pangeoradar-termite**.

В результате, события должны обогащаться GeoIP информацией, как изображено на рисунке 46.

initiator.host.geoip.city	[null]
initiator.host.geoip.continent	["North America"]
initiator.host.geoip.country	["United States"]
initiator.host.geoip.iso	["US"]
initiator.host.geoip.key	["66.249.64.137"]
initiator.host.geoip.location	[[-97.822, 37.751]]
initiator.host.geoip.timezone	["America/Chicago"]
initiator.host.hostname	[]
initiator.host.internal	false
initiator.host.ip	["66.249.64.137"]
initiator.network.node.host.hostname	[]
initiator.process.id	13353
initiator.socket.port	22758

Рисунок 46 - Обогащенное GeoIP событие

14.2. Настройка DNS обогащения

DNS обогащение может работать как от .csv файла с базой FQDN и IP-адресов, так и получая информацию от DNS сервера. Можно использовать оба способа единовременно.

14.2.1. DNS обогащение по сети

Для организации работы DNS обогащения по сети необходимо произвести настройку в конфигурационном файле модуля обработки событий `/opt/pangeoradar/configs/termite/conf.yaml` добавив туда следующие записи:

```
dns:
  enabled: true
  domains:
  - demo.local # Домены для dns обогащения
  nets: [192.168.0.0/16] # Сети для dns обогащения
  servers: [192.168.150.15] # Сервера для dns обогащения
  port: 53 # Порт для dns обогащения
  local: false # вкл/выкл только локальное dns обогащение
  in_memory: # dns-кэш
    enabled: true # вкл/выкл dns-кэш
    expire: 10800 # время (сек) через которое записи удаляются из кеша
```

14.2.2. Локальное DNS обогащение

Для организации работы DNS обогащения из файла необходимо произвести настройку в конфигурационном файле модуля обработки событий `/opt/pangeoradar/configs/termite/conf.yaml` добавив туда следующие записи:

```
dns:
  enabled: true
  domains:
  - demo.local # Домены для dns обогащения
  nets: [192.168.0.0/16] # Сети для dns обогащения
  local: true # вкл/выкл только локальное dns обогащение
  in_memory: # dns-кэш
    enabled: true # вкл/выкл dns-кэш
    expire: 10800 # время (сек) через которое записи удаляются из кеша
    preload_from_file: /opt/pangeoradar/configs/termite/demo.local-output.csv #
Путь до файла csv
```

Пример представления CSV файла с перечнем FQDN и IP-адресов:

```
'test1.demo.local','192.168.1.1'
'192.168.1.100','test3.demo.local'
```

В результате, события должны обогащаться DNS информацией, как изображено на рисунке 47.

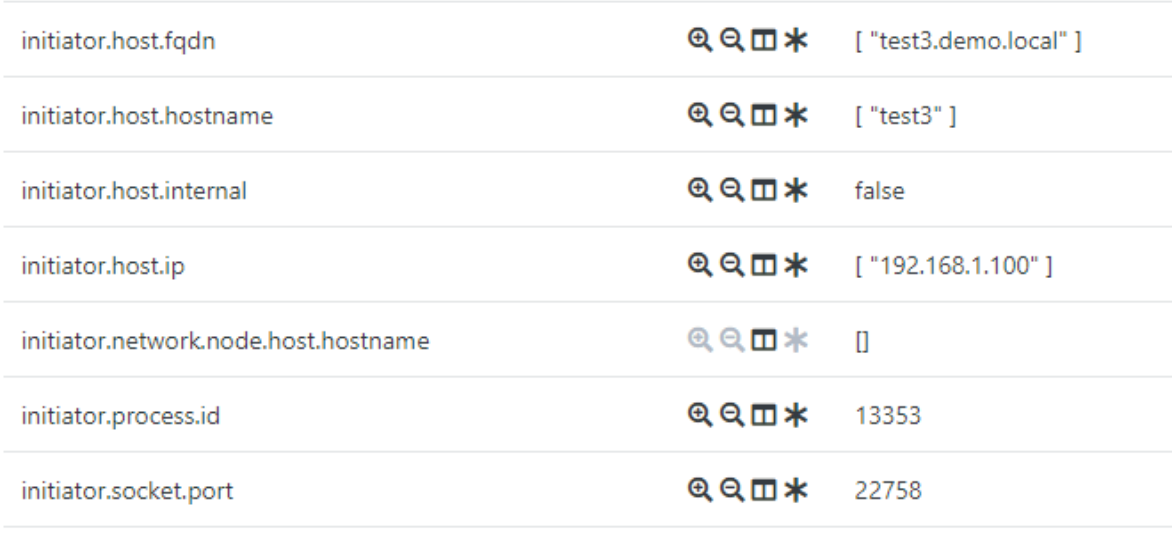

Рисунок 47 - Обогащенное DNS событие

14.3. Настройка Threat Intelligence обогащения

Threat Intelligence обогащение работает на основе баз угроз безопасности, получаемых Платформой различных поставщиков.

Для просмотра базы Threat Intelligence необходимо в интерфейсе Платформы перейти в раздел “Репутационные списки”. Раздел изображен на рисунке 48.

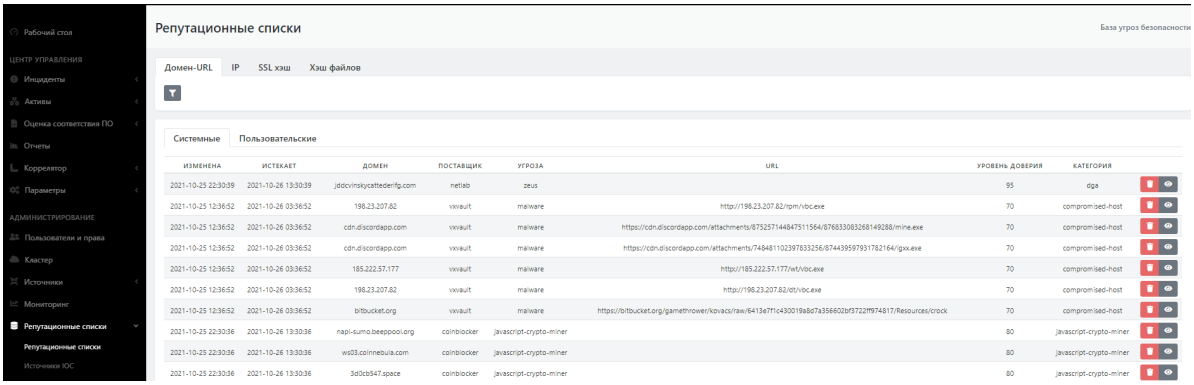

Рисунок 48 - Репутационные списки

TI обогащение позволяет наполнять дополнительной информацией события, содержащие: Домен-URL, IP - адрес, SSL хэш, Хэш файлов из базы угроз.

Для работы TI - обогащения необходимо в конфигурационном файле модуля обработки событий `/opt/pangeoradar/configs/termite/conf.yaml` добавить следующие записи:

```
#Пример настройки при Standalone инсталляции:

threatintel: # threat intelligence обогащение
  enabled: true
  service-url: http://localhost:8082/
  db-path: ./threat.db
```

В результате, события должны обогащаться TI информацией, как изображено на рисунке 49.

initiator.blacklist.ip.category	["compromised-host"]
initiator.blacklist.ip.confidence	[33]
initiator.blacklist.ip.ip	["119.236.128.231"]
initiator.blacklist.ip.port	[null]
initiator.blacklist.ip.protocol	["tcp"]
initiator.blacklist.ip.provider	["alienvault"]
initiator.blacklist.ip.threat	["compromised-host"]
initiator.host.fqdn	[]
initiator.host.hostname	[]
initiator.host.internal	false
initiator.host.ip	["119.236.128.231"]
initiator.network.node.host.hostname	[]
initiator.process.id	13353
initiator.socket.port	22758

Рисунок 49 - Обогащенное TI событие

14.4. Настройка RVS обогащения

RVS обогащение работает на основе табличных списков.

1. Для настройки RVS обогащения необходимо в табличном списке создать коллекцию (вручную или специальными средствами для обогащения).
   Работа с интерфейсом табличных списков представлена в руководстве по работе с RVS (табличные списки) ;
2. Далее, в созданной коллекции, необходимо добавить документ, пример которого изображен на рисунке 50.

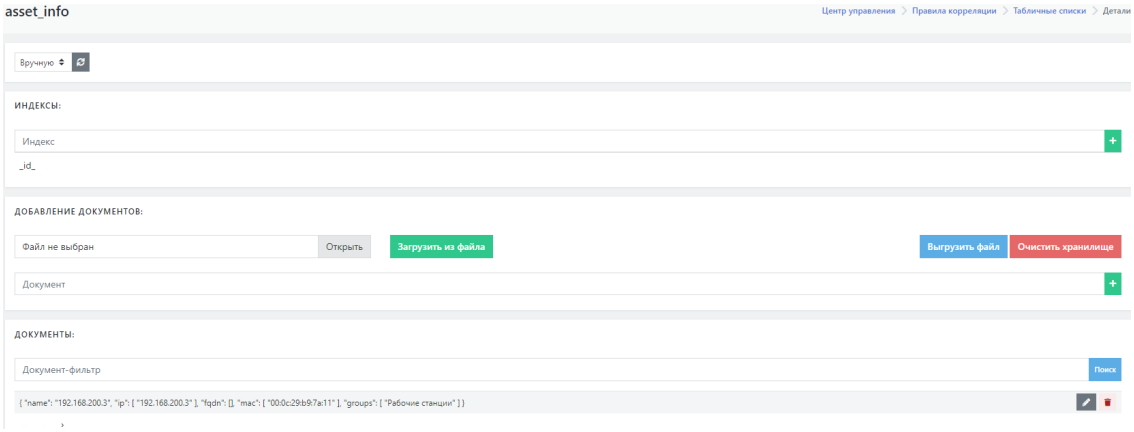

Рисунок 50 - Табличные списки

Созданный документ в json формате:

```
{
"name": "192.168.200.3",
"ip": [
    "192.168.200.3"
],
"fqdn": [],
"mac": [
    "00:0c:29:b9:7a:11"
],
"groups": [
    "Рабочие станции"
]
}
```

3. В конфигурационном файле модуля обработки событий `/opt/pangeoradar/configs/termite/conf.yaml` необходимо добавить следующие записи:

```
rvs:
  enabled: false                                          # включение
rvs обогащения
  host: <IP-адрес-Платформы>                              # адрес
MongoDB (роль: Мастер)
  inmemory_collection_size: 60                            # размер
коллекции, хранящийся в памяти
  mapping:                                                # настройка
сопоставления для обогащений
    2162-Cisco-NetFlow:                                   # название
источника (event.logsource.input)
      asset_info:                                         # название
коллекции
      - enrich_from:                                      # настройка
проверки совпадения полей коллекции и нормализованного события
          collection_field: ip                            # поле в
коллекции, сопоставляемое с полем в нормализованном событии
          normalized_field: initiator.host.ip             # поле в
нормализованном событии, сопоставляемое с полем коллекции
        enrich_to:                                        # настройка
обогащения полей из коллекции в поля нормализации
        - collection_field: mac                           # содержимое
поля в коллекции
          normalized_field: initiator.interface.mac       # целевое поле
в нормализованном событии
  no_verify: true                                         # не проверять
сертификат
  port: 27017                                             # порт MongoDB
  secure: true                                            #
использование SSL
  update_interval: 60                                     # частота
обновлений
```

Таким образом, обогащение производится по следующему принципу:
Если приходит событие, в котором значение поля нормализованного события **initiator.host.ip** равно значению поля **ip** одного из документов коллекции в RVS - заменять/добавлять в поле **initiator.interface.mac** нормализованного события, значение поля **mac** документа в RVS.

Пример:

В Платформу приходит событие:


```
"Version":9,"SysUpTimeMilisec":2549185408,"EventReceivedTime":"2022-02-
17T17:31:42+03:00","MessageSourceAddress":"192.168.200.1","SourceIPv4Address":"1
92.168.200.3","DestIPv4Address":"172.30.254.128","inputSNMPIface":3,"outputSNMPI
face":2,"InPackets":1,"InBytes":226,"postPacketDeltaCount":2,"postOctetDeltaCoun
t":78,"FlowStart":1645108153,"FlowEnd":1645108153,"SourcePort":53,"DestPort":344
20,"postNATSourceIPv4Address":"185.120.22.23","postNATDestinationIPv4Address":"1
92.168.150.15","postNAPTSourceTransportPort":53,"postNAPTDestinationTransportPor
t":61962,"privateEnterpriseNumber":2620,"natInstanceID":0,"natThresholdEvent":0,
"ProtocolIdentifier":17,"TCPFlags":0
```


В поле **initiator.host.ip** значение ['192.168.200.3'], что совпадает с полем **ip** в одной из записей в коллекции. В результате в поле **initiator.interface.mac** должно записаться значение поля **mac** из записи в коллекции, то есть значение '00:0c:29:b9:7a:11'.

Результат обогащенного табличным списком события представлен на рисунке 51.

Рисунок 51 - исунок 6. Обогащенное RVS событие

14.5. Lookup обогащение

Lookup обогащение происходит на этапе нормализации событий.

Описание Lookup представлено в разделе **Специальные функции для работы с полями нормализации**.

15. Фильтрация событий

Платформа поддерживает фильтрацию входящих событий на двух этапах:

1. Фильтрация на этапе сбора лог-коллектором;
2. Фильтрация на этапе принятия событий модулем обработки событий.

15.1. Фильтрация на этапе сбора лог-коллектором

Фильтрацию на этапе сбора лог-коллектором можно разбить на два типа:

15.1.1. Фильтрация структурированных данных

Применима для таких типов источников, как:

- WMI
- Event log
- ODBC
- ETW

Фильтрация в компонентах сбора со структурированными данными работает как blacklist и настраивается при описании источника, в секции filters.

Пример фильтрации, исключающий сбор событий с уровнем Information:

```
filters:
  created: ''
  event_id: ''
  qualifiers: ''
  record_id: ''
  process_id: ''
  thread_id: ''
  version: ''
  computer_name: ''
  msg: ''
  level_text: 'Information'
  task_text: ''
  opcode_text: ''
  channel_text: ''
  provider_text: ''
```

15.1.2. Фильтрация неструктурированных данных

Фильтры можно указать для каждого коллектора с неструктурированными данными.

Фильтры содержат белый список (whitelist) и черный список (blacklist) с массивом регулярных выражений.

Все регулярные выражения проверяются перед запуском приложения. Сначала проверяется белый список, а затем черный.

Фильтрация по регулярным выражениям может быть включена в любом коллекторе с неструктурированными данными.

Включается путем добавления секции filters. В данной секции указываются два массива — whitelist, blacklist.

Все события сначала проходят фильтры указанные в whitelist, т.к. его приоритет выше. Затем события проверяются фильтрами, указанными в blacklist.

Whitelist — события, которые соответствуют регулярному выражению, попадают в очередь на отправку.
Blacklist — события, которые соответствуют регулярному выражению, блокируются и не попадают в очередь на отправку.

Пример фильтрации для неструктурированных данных:

```
 filters:
     whitelist:
       - "^localhost.*$"
     blacklist:
       - "^.*[0-9]$"
```

15.2. Фильтрация на этапе принятия событий модулем обработки событий

Фильтрация на данном этапе осуществляется через файл `/etc/termite/processors.py`

Блок предварительной фильтрации - это простая функция Python, которая принимает сырое событие и определенное условие, в качестве входных данных, и возвращает логическое значение.

Значение False приводит к пропуску сырого события.

Пример использования, по которому, если в сыром событии присутствует слово "irrelevant" - событие пропускатеся:

```
@prefilter('drop-irrelevant')
def drop_irrelevant(raw):
    return 'irrelevant' not in raw
```

15.3. Настройка фильтрации поступающих событий

При получении сообщений от внешних источников можно настроить фильтрацию поступающих событий.

Для настройки фильтрации необходимо выполнить следующие действия:

1. Подключиться по SSH к узлу обработки событий (Worker) платформы.
2. Открыть на редактирование файл:

   nano /opt/pangeoradar/termite2/venv/lib/python3.7/site-packages/termite_spu/prefilters/prefilters.py
3. Добавить в конец файла следующую запись:

   `@prefilter('<имя>')`

   `def drop_demo_user(raw: str):`

   `return '"rrname":"<ключ>"' not in raw`

   где:

   <*имя*> - уникальное имя фильтра.

   <*ключ*> - строка из состава сырых данных события, по наличию которой будет происходить фильтрация поступающих событий.
4. Сохранить изменения в файле.
5. Открыть файл pipelines.yaml:

   nano /opt/pangeoradar/configs/termite/pipelines.yaml
6. Найти в файле раздел настроек источника, чьи поступающие сообщения необходимо фильтровать, и добавить в этот раздел следующую запись:

   `prefilters: ['<имя>']`,

   где: <*имя*> - имя фильтра, заданное на шаге 3.
7. Сохранить изменения в файле.

Для проверки работы фильтрации поступающих сообщений - перейти в веб-интерфейс платформы в раздел **"Просмотр событий"** и убедиться, что указанные для фильтрации события не поступают в платформу.

16. Агрегация событий

16.1. Настройка агрегации событий

В Платформе предусмотрена возможность агрегации событий. Настроить агрегацию можно либо при создании нового источника событий, либо при редактировании параметров ранее заведенного в Платформе источника.

Для настройки агрегации необходимо выполнить следующие действия (на примере редактирования):

1. В веб-интерфейсе Платформы зайти в подраздел "Источники"->"Управление источниками"->вкладка "Источники".
2. В текущем списке источников событий выбрать интересующий и нажать кнопку редактирования для данного источника  .
3. В форме редактирования источника в области параметров **Правила для termite** для параметра "**Агрегация**" выбрать в раскрывающемся списке одно или последовательно несколько полей, которые не должны меняться, и по которым будет происходить агрегация событий на данном источнике (см. Рисунок 52). Расшифровка полей для агрегации дана в разделе «Описание полей нормализации».
4. Для сохранения введенных изменений нажать кнопку **Сохранить**.
5. Синхронизировать источники, нажав кнопку **Синхронизировать** на вкладке "Источники".

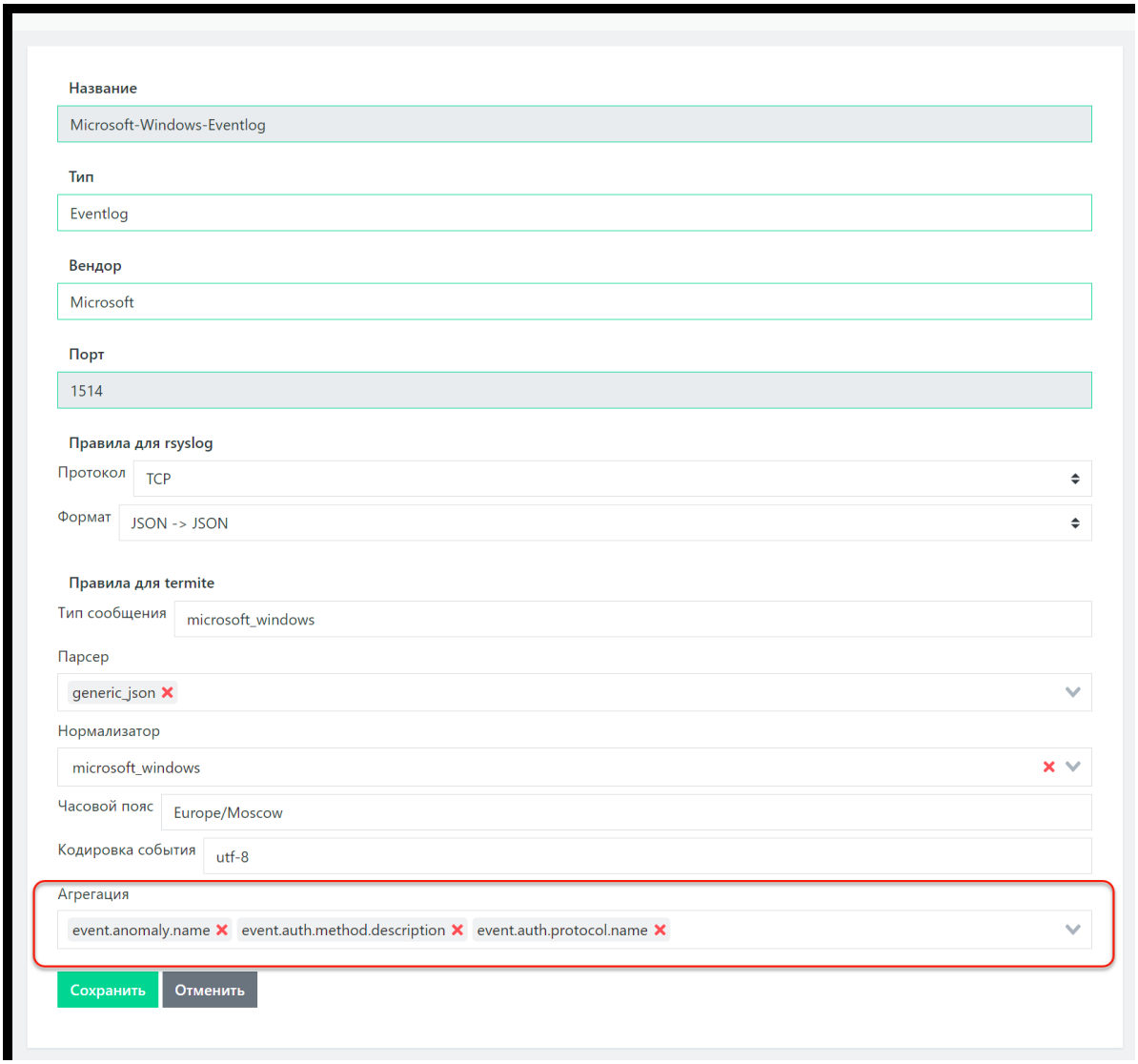

Рисунок 52 - Настройка агрегации событий

16.2. Просмотр результатов агрегации событий

Для просмотра результатов агрегации событий необходимо выполнить следующие действия:

1. В веб-интерфейсе Платформы зайти в раздел "Просмотр событий".
2. Задать временной интервал в поле **Время**.
3. Ввести или выбрать в раскрывающемся списке нужный индекс в поле **Индекс**.
4. Обновить данные на экране согласно заданным параметрам нажав .
5. В левой части экрана в области "**Доступные поля**" найти поле **grouped_occur_count** и нажать .

Поле **grouped_occur_count** добавится в область "**Выбранные поля**". Колонка поля **grouped_occur_count** добавится в табличный список событий (см. Рисунок 53). В данной колонке отображается количество агрегированных событий.

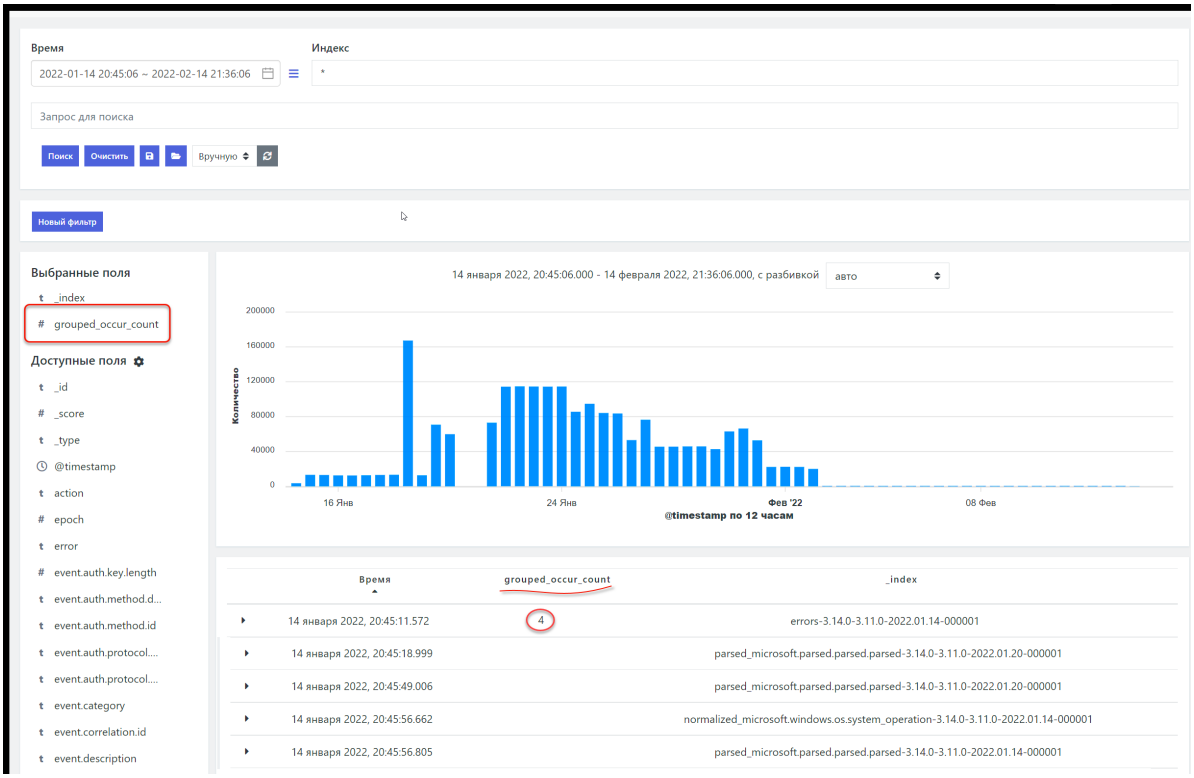

Рисунок 53 - Отображение агрегированных событий в табличном списке событий в разделе "Просмотр событий"

17. Руководство по настройке лог-коллектора. Активные источники событий

17.1. Радар лог-коллектор. Описание.

Радар лог-коллектор (RADAR LOG-COLLECTOR), далее лог-коллектор, предназначен для организации активного сбора событий от активов, не имеющих возможности отправки данных в сторонние системы. Лог-коллектор позволяет организовать различные схемы сбора событий от любых активов, участвующих в сетевом взаимодействии, создающих журналы событий.

Основные функции:

- сбор, локально и удалённо по различным протоколам;
- отправка событий в другие системы;
- обработка событий перед отправкой;
- пересылка событий в зашифрованном виде и со сжатием;
- отправка по расписанию;
- накопление событий при разрыве соединения и отправка после восстановления.

Поддерживаемые операционные системы следующих семейств:

- Windows XP, Windows 2003 Server, Vista+, 2008+;
- Linux Debian;
- Linux CentOS;
- Linux RedHat.

Поддерживаемые способы и протоколы отправки:

- TCP;
- SSL/TLS TCP;
- UDP;
- Kafka;
- Запись в локальный файл.

Сбор событий

1. Локальный:

- Event Tracing for Windows (ETW);
- File Read;
- Windows Event Log;
- Результаты работы скрипта (Python/CMD/PowerShell/Bash/Perl).

2. Удалённый:

- Windows Event Log via RPC;
- WMI;
- ODBC;
- File via SMB;
- File via SSH;
- File via FTP;
- File via SFTP;
- File via HTTP(S);
- Checkpoint OPSEC LEA.

3. Пассивный:

- TCP;
- UDP;
- Netflow v5, v9;
- Syslog;
- SNMP Trap;
- HTTP.

17.2. Основные характеристики

Программное обеспечение лог-коллектор обеспечивает решение перечисленных ниже основных задач:

- сбор/прием событий;
- обработка событий;
- отправка событий;
- временное хранение событий.

Сбор/прием *событий* может осуществляться в любом из трех режимов:

- **Локальный**. Лог-коллектор устанавливается в системе в виде агента, производит чтение файлов etw, eventlog, wmi, получение результатов выполнения скриптов (bash, perl, python, PowerShell) и их отправку в *Платформу* (либо на промежуточный агент).
- **Пассивный**. Лог-коллектор осуществляет прием *событий* от систем, которые могут самостоятельно отправлять данные.
- **Удаленный/активный.** Лог-коллектор устанавливается на выделенный сервер и осуществляет удаленный сбор *событий* по различным протоколам. Также может быть установлен на конечном *источнике событий*, и осуществлять сбор не только с этого *источника событий*, но и с других систем. Необходимо предусмотреть дополнительные ресурсы для работы Лог-коллектора.

Обработка событий позволяет обогатить события дополнительной технической информацией, изменить формат и кодировку перед их отправкой.

Отправка событий может осуществляться посредством их одновременной передачи в *Платформу* и несколько внешних систем, в зашифрованном виде со сжатием.

Временное хранение событий предотвращает потерю *событий* при разрыве соединения, накапливая *события* для их последующей отправки после восстановления соединения.

С учётной записью/без учётной записи. Лог-коллектор имеет возможность сбора событий как с использованием указанной служебной учетной записи для доступа к событиям, так и без учётной записи (для транспортов, где это технически возможно).

17.3. Архитектура

Лог-коллектор имеет компонентную архитектуру и включает в себя следующие программные *компоненты*:

- **Контроллер** (controller)— осуществляет управление всеми компонентами лог-коллектора
- **Компонент сбора метрик** (metric_server)— осуществляет сбор статистики по работе лог-коллектора
- **Компонент API управления** (api_server)— предоставляет возможность удаленного управления лог-коллектором и мониторинга
- **Компонент журналирования** (journal)— осуществляет ведение журнала работы лог-коллектора
- **Компоненты сбора/приема событий** (inputs) — осуществляют сбор событий
- **Компоненты отправки событий** (outputs) — осуществляют оправку событий

Управление настройками лог-коллектора и его *компонентов* может осуществляться как через основной конфигурационный файл на сервере, где установлен экземпляр Лог-коллектора, так и централизовано, из веб-интерфейса *Платформы Радар* Управление Лог-коллектором.

17.4. Установка лог-коллектора

17.4.1. Требования к техническому и программному обеспечению

Минимальные требования к ресурсам:

- 4 Core
- 4 GB RAM
- 60 GB HDD

Минимальные требования к ресурсам при установке в системе с настроенным сервисом Windows Event Collector:

- 4 Core
- 8 GB RAM
- 500 GB HDD

> **Важно!** Требования к объему дискового пространства представлены без учета промежуточного хранения.

Для нормального функционирования лог-коллектора требуется установка на выделенные ресурсы одной из нижеперечисленных операционных систем, являющихся средой функционирования Лог-коллектора:

- Windows Vista+, 2008+
- Windows XP, Windows 2003 Server
- Linux Debian

На приведенных выше минимальных требованиях к ресурсам лог-коллектор обеспечивает обработку потока 5000 событий в секунду.

17.4.2. Возможные схемы развертывания

- Установка на источнике для организации локального сбора *событий* с последующей передачей в *Платформу* или в промежуточный Лог-коллектор.
- Установка на выделенный сервер для организации удаленного сбора и пересылки *событий*.
- Установка цепочки лог-коллекторов для передачи *событий* в зашифрованном виде.

17.4.3. Установка лог-коллектора на различных ОС

> **Важно!** Установка лог-коллектора осуществляется под учетной записью с правами администратора.

Перед установкой необходимо скопировать на целевую систему архив с дистрибутивом, защищенный паролем. Пароль передается отдельно от архива.

17.4.3.1. Установка в ОС Windows

Разархивировать папку с дистрибутивом в корневой раздел диска C. При выполнении распаковки нужно будет ввести пароль от архива.

После распаковки в папке должны быть следующие файлы:

- log-collector.exe (дистрибутив лог-коллектора)
- config.yaml (файл конфигурации с минимально необходимыми для подключения настройками)
- example.yaml (пример общего файла конфигурации)

Необходимо запустить терминал от имени администратора и перейти в раздел, куда предварительно была распакована папка с дистрибутивом Лог-коллектора.

```
cd c:\log-collector
```

Выполнить установку с помощью команды:

```
log-collector-<версия релиза>-<тип архитектур>.exe winsvc
```

Данная команда установит лог-коллектор в качестве сервиса ОС.

Запуск сервиса

```
net start PangeoRadarLogCollector
```

Остановка сервиса

```
net stop PangeoRadarLogCollector
```

Так же можно выполнить запуск/остановку/перезапуск сервиса из оснастки «Сервисы»

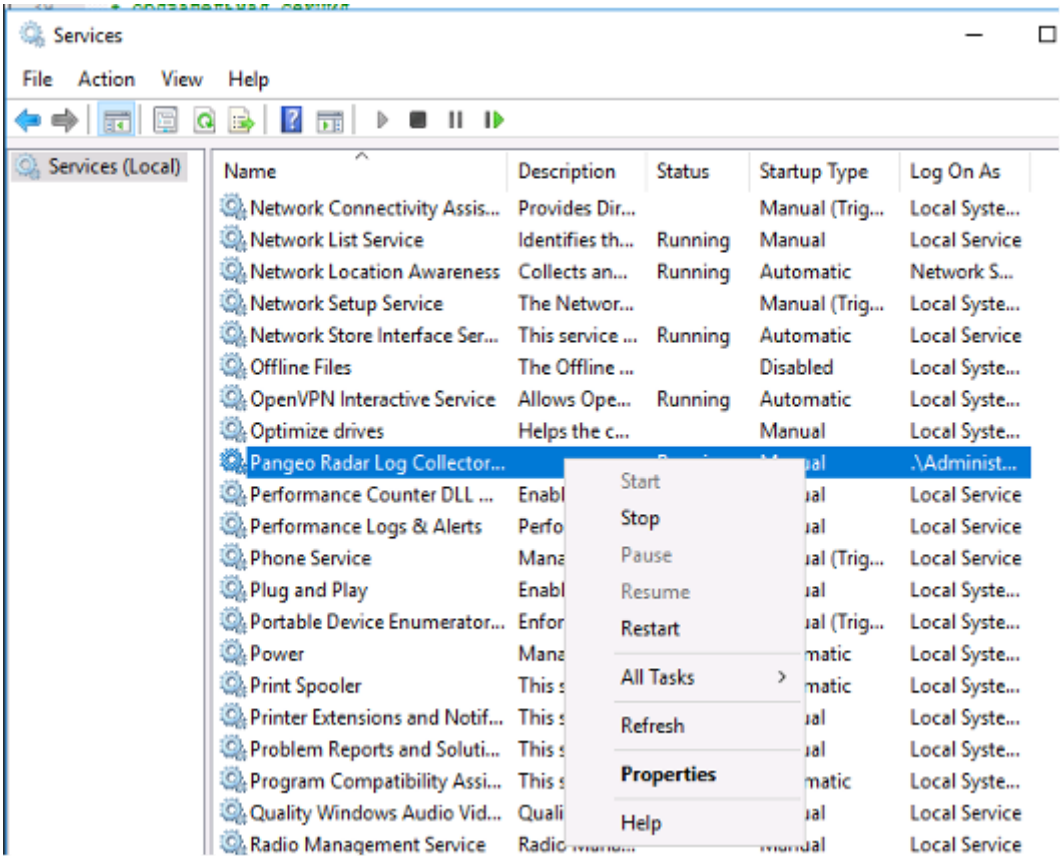

Рисунок 54 - Управление службой лог-коллектор в оснастке «Сервисы»

Если при запуске/перезапуске сервиса выводится ошибка, причину следует определять просмотром записей в журнале неудачных запусков Лог-коллектора **logcollector.crash.log** в папке, из которой была выполнена установка сервиса.

Конфигурационный файл лог-коллектора **config.yaml** находится в папке, откуда была выполнена установка сервиса.

17.4.3.2. Установка в ОС Linux Debian

Для выполнения установки необходимо распаковывать архив с дистрибутивами командой:

```
openssl enc -aes-256-cbc -d -in log-collector-<версия релиза>-<тип
архитектур>.tar.gz.enc | tar xz
```

Далее выполнить установку командой:

```
dpkg -iR ./
```

Проверка состояния сервиса

```
systemctl status log-collector
```

Перезапуск сервиса

```
systemctl restart log-collector
```

17.5. Основные настройки лог-коллектора

> **Важно!** Для успешного запуска лог-коллектора необходимо выполнить основные настройки в конфигурационном файле **config.yaml**

17.5.1. Настройка централизованного управления

Лог-коллектор может управляться через файл конфигурации непосредственно на сервере, где он развернут, или централизованно из веб-интерфейса Платформы.

Для подключения экземпляра к Платформе для управления из интерфейса, необходимо добавить в файл конфигурации следующие директивы:

```
# Централизованное управление
cluster:
  url: "http://<ip адрес Платформы Радар>:9000/cm/api/agent/"
  api_key: "<API ключ>"
```

> где:
> *<API ключ>* - ключ доступа к API, сгенерированный в веб-интерфейсе Платформы

Более подробное описание по настройке централизованного управления приведено в Управление лог-коллектором.

17.5.2. Настройка контроллера

Для настройки контроллера необходимо добавить в конфигурационный файл следующие директивы:

```
controller:
  # Порт компонента, обязательный параметр
  port: 48000
```

17.5.3. Настройка компонента сбора метрик

Для настройки сбора метрик и статистики необходимо добавить в конфигурационный файл следующие директивы:

```
metric_server:
  # Порт компонента, обязательный параметр
  port: 48005
```

17.5.4. Настройка размещения защищенного хранилища

Защищенной хранилище используется для хранения чувствительных данных, которые не должны храниться в открытом виде.

Для настройки размещения необходимо добавить в конфигурационный файл следующие директивы:

```
# Путь к файлу с секретом
secret_file: "C:\\log-collector\\secret"
# Путь к хранилищу секретов
secret_storage: "C:\\log-collector\\secret.storage"
```

Для создания секрета используется команда:

```
# <путь>log-collector secrets set <ключ> <значение> --config <путь до
конфигурационного файла>
```

> где:
> <*путь*> - путь до каталога, в котором находится исполняемый файл лог-коллектора
> <*ключ*> - имя значения
> <*значение*> - данные, которые нужно скрыть в конфигурационном файле

После создания секрета его можно подставить в виде конструкции - `{{.ключ}}`, вместо любой строки в конфигурационном файле. Таким образом, все учетные данные могут быть скрыты.

Пример создания секретов и использования их в конфигурационном файле:

```
log-collector secrets set user User --config /etc/log-collector/config.yaml
```

```
log-collector secrets set user_password $ecure_P@sswOrd --config /etc/log-
collector/config.yaml
```

Пример использования в конфигурационном файле:

```
ssh_collector: &ssh_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "ssh_collector"
  # Имя пользователя для удаленного подключения, обязательный параметр\
  user: "{{.user}}"
  # Список хостов для подключения, обязательный параметр
  remote_servers: ["<ip адрес удаленного узла>", "<имя удаленного узла>"]
```

```
  # Порт для подключения (default: 22)
  port: 22
  # Путь к файлу с ssh ключами, обязательный параметр
  rsa: "./ssh"
  # Пароль от файла с ключами
  password: "{{.user_password}}"
  # Команда для выполнения по ssh, обязательный параметр
  command: "tail -F -n +$$$line$$$ /var/log/sshfile.txt"
  # Если установлено - файл будет читаться с последней позиции в следующем тике
  # или после перезапуска
  read_from_last: true
  # Интервал между выполнением команд (в секундах)
  ticker: 30
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
```

Для просмотра созданных ключей используется команда:

```
# <путь>log-collector secrets list --config <путь до конфигурационногофайла>
```

> где:
> *<путь>* - путь до каталога, в котором находится исполняемый файл лог-коллектора

Для удаления ключей используется команда:

```
<путь>log-collector secrets remove <ключ> - -config <путь до конфигурационного
файла>
```

> где:
> *<путь>* - путь до каталога, в котором находится исполняемый файл лог-коллектора
> *<ключ>* - имя значения, которое нужно удалить.

> **Важно!** Прописывать ключи в файле конфигурации нужно в кавычках, как в примере выше **"{{.ключ}}"**, иначе при запуске приложения упадет с ошибкой из-за того что не сможет прочитать файл конфигурации. Соответствующее сообщение будет в журнале неудачных запусков Лог-коллектора **logcollector.crash.log**.

17.5.5. Настройка API

API необходимо для удаленного управления экземпляром лог-коллектора, получения журнала *событий* работы лог-коллектора и сбора статистики. Для настройки работы API необходимо добавить в конфигурационный файл следующие директивы:

```
api_server:
  # ip адрес, на котором будем слушать http сервер
  address: "<Внещний ip адрес сервера, на котором установлен лог-коллектор>"
  # порт, на котором будем слушать http сервер, обязательный параметр
  port: 8080
  # Таймаут чтения (получение запроса), обязательный параметр
  read_timeout: 60
  # Таймаут записи (отправка запроса), обязательный параметр
```

```
  write_timeout: 60
  # Время ожидания окончания обработки запроса при получении сигнала на остановку
приложения
  # Обязательный параметр
  wait: 5
  # Включение https (защищенное соединение)
  enable_tls: false
  # Путь для файла сертификатов, если enable_tls: false параметр не обязательный
  cert_file: "certs/server.crt"
  # Путь для файла ключей, если enable_tls: false параметр не обязательный
  key_file: "certs/server.key"
  # Пароль для расшифровки файла ключей, если не указан считаем, что файл не
зашифрован
  cert_key_pass: ""
  # Включение проверки клиентского сертификата, обязательный параметр
  require_client_cert: true
  # Путь до корневого сертификата, обязательный параметр
  ca_file: "certs/ca.crt"
  # Уровень логирования, если не указан используется указанный в компоненте
журналирования
  log_level: "INFO"
```

17.5.6. Настройка журналирования

В журнал лог-коллектора записываются все *события*, происходящие в *компонентах* лог-коллектора с уровнем логирования, указанным для каждого *компонента*. Если уровень логирования не указан для конкретного *компонента*, то логирование происходит с уровнем, выставленным в *компоненте* журналирования.

> **Важно**! Уровень логирования DEBUG используется только для отладки работы *компонентов*. В промышленной эксплуатации рекомендуется использовать уровень INFO.

```
journal:
  # Порт компонента, обязательный параметр
  port: 48004
  # Еровень логирования по умолчанию. Возможные значения - DEBUG, INFO, WARN,
ERROR.
  # Обязательный параметр
  log_level: "INFO"
  # Путь к файлу журнала, обязательный параметр
  log_path: "C:\\log-collector\\journal.log"
  # Порог ротации файла логов, указывается в мегабайтах, обязательный параметр
  rotation_size: 30
  # Порог количества файлов истории, если не указано файлы удаляться не будут
  max_backups: 7
  # Максимальное количество дней для хранения старых файлов журнала на основе
метки времени
  # Если не указано, файлы удаляться не будут (в днях).
  max_age: 7
```

17.6. Фильтрация событий

17.6.1. Структурированные данные

Фильтрация в *компонентах* сбора со структурированными данными работает как **blacklist** и применима к коллекторам wmi, eventlog, odbc, etw.

17.6.2. Неструктурированные данные

Фильтры можно указать для каждого коллектора с неструктурированными данными. Фильтры содержат белый список (**whitelist**) и черный список (**blacklist**) с массивом регулярных выражений. Все регулярные выражения проверяются перед запуском приложения. Сначала проверяется белый список, а затем черный.
Фильтрация по регулярным выражениям может быть включена в любом коллекторе с неструктурированными данными (кроме odbc, wmi, etw, eventlog). Включается путем добавления секции **filters**. В данной секции указываются два массива — **whitelist**, **blacklist**. Все события сначала проходят фильтры указанные в **whitelist**, т.к. его приоритет выше. Затем события проверяются фильтрами, указанными в **blacklist**.
Whitelist — события, которые соответствуют регулярному выражению, попадают в очередь на отправку.
Blacklist — события, которые соответствуют регулярному выражению, блокируются и не попадают в очередь на отправку.

```
ssh_collector: &ssh_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "ssh_collector"
  # Имя пользователя для удаленного подключения, обязательный параметр
  user: "user"
  # Список хостов для подключения, обязательный параметр
  remote_servers: ["<ip адрес удаленного узла>", "<имя удаленного узла>"]
  # Порт для подключения (default: 22)
  port: 22
  # Путь к файлу с ssh ключами, обязательный параметр
  rsa: "./ssh"
  # Пароль от файла с ключами
  password: "password"
  # Команда для выполнения по ssh, обязательный параметр
  command: "tail -F -n +$$$line$$$ /var/log/sshfile.txt"
  # Если установлено - файл будет читаться с последней позиции в следующем тике
  # или после перезапуска
  read_from_last: true
  # Интервал между выполнением команд (в секундах)
  ticker: 30
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  filters:
    whitelist:
      - "^localhost.*$"
    blacklist:
      - "^.*[0-9]$"
```

17.7. Настройка очереди отправки событий

Применимо к *компонентам* отправки событий TCP и KAFKA

```
# Максимальное количество сообщений в буфере
queue_length_limit: 1500
# максимальное время жизни событий в очереди (в секундах)
queue_time_limit: 300
```

17.8. Формат отправки данных

Применим ко всем неструктурированным *компонентам* сбора событий (кроме eventlog, wmi, odbc, etw)

```
# Формат отправки сообщения - как есть(raw), с обогащением(json)
format: "raw"
```

> **raw** - данные отправляются в том виде, в котором пришли
> **json** - пришедшие данные обогащаются дополнительной технической информацией и упаковываются в пакет json

17.9. Кодировка

Данная секция позволяет изменить кодировку входящих данных. Если не указывать original_encoding, лог-коллектор сам попытается определить кодировку.

```
# Опции смены кодировки
encoding:
  # Использовать кодировку в UTF-8
  change_to_utf8: false
  # Кодировка оригинала
  original_encoding: "cp1251"
```

17.10. Описание хранилища приложения

Хранилище ключей значений в файловой системе ОС (LevelDB). Используется для хранения ссылок и буферизации событий. Не предусмотрено стороннее редактирование.
Папка с файлами хранилища (.storage) располагается в рабочей директории лог-коллектора.

17.11. Компоненты лог-коллектора

Каждый компонент сбора и отправки может иметь один и более экземпляров в файле конфигурации.

> **Важно!** При настройке не использовать повторяющихся названий настроек компонентов, ссылок на них и уникальных идентификаторов.

Пример:

```
# Название конфигурации компонента и ссылка на него для запуска
<наименование настройки компонента>: &<уникальная ссылка на настройку>
# Уникальный идентификатор компонента, отображается в логах и метриках.
Обязательный параметр
  id: "<уникальный идентификатор>"
```

17.11.1. Компоненты сбора событий (inputs)

Для включения компонента сбора необходимо добавить в файл конфигурации его настройки.

17.11.1.1. Компонент Eventlog

> **Важно!** Работает только на ОС Windows.

Позволяет выполнить локальный или удаленный сбор событий Windows.

Пример настроек по умолчанию:

```
# Название конфигурации компонента и ссылка на него для запуска
eventlog_collector: &eventlog_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "eventlog_collector"
  # Имя канала (Security, Application, System, ForwardedEvents)
  # Используется если не указан путь к файлу
  channel: ['Security']
  # Запрос описывающий тип получаемого события. Может быть в формате
  # XPath 1.0 или структурированный XML запрос. Если XPath содержит более 20
параметров,
  # следует использовать структурированный XML запрос.
  # Чтобы получить все параметры укажите "*"
  query: "*"
  # Полный путь к файлу журналов событий
  # Поддерживаемые форматы: .evt, .evtx, .etl
  file: ""
  # Размер запроса
  batch_size: 50
  # Таймаут запроса в секундах
  timeout: 5
  # Интервал между запуском запроса в секундах
  poll_interval: 30
  # Чтение с последней сохраненной позиции
  read_from_last: true
  # Конвертировать SID в имя
  resolve_sid: false
  # формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования. Если не указан используется уровень компонента
журналирования
  log_level: "INFO"
  # Параметры удаленного подключения
  remote:
    # Включение удаленного соединения
    enabled: false
    # Имя пользователя, обязательно если enabled: true
```

```
    user: ""
    # Пароль пользователя, обязательно если enabled: true
    password: ""
    # Домен пользователя
    domain: "."
    # Адрес удаленного сервера
    remote_servers: ["<ip адрес удаленного узла>", "<имя удаленного узла>"]
    # Доступные методы авторизации: Negotiate, Kerberos, NTLM
    auth_method: "Negotiate"
    # Фильтрация по полям события, регулярные выражения
  filters:
    # Время
    # формат 2020-08-13 10:02:55.9689259 +0000 UTC
    created: ''
    # Числовые фильтры
    # Пример для числовых фильтров - ^([5-9]\d|\d{3,})$
    event_id: ''
    qualifiers: ''
    record_id: ''
    process_id: ''
    thread_id: ''
    version: ''
    # Строковые фильтры
    # пример: DESKTOP-IDCMV6G
    computer_name: ''
    msg: ''
    # Возможные значения: Information, Warning, Error
    level_text: ''
    # Пример: Service State Event
    task_text: ''
    # Пример: ServiceShutdown
    opcode_text: ''
    # Пример: System
    channel_text: ''
    # Пример: System
    provider_text: ''

    # Возможно применение опций смены кодировки
```

17.11.1.2. Компонент Eventlog_XP

> **Важно!** Работает только на ОС Windows XP, 2003

```
eventlog_xp_collector: &eventlog_xp_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "eventlog_xp_collector"
  # Имя канала
  channel: ['Security']
  # Интервал между запуском запроса в секундах
  poll_interval: 5
  # Чтение с последней сохраненной позиции
  read_from_last: false
  # Уровень логирования, если не указан используется уровень компонента
журналирования
```

```
  log_level: "INFO"
  # Фильтрация по полям события, регулярные выражения
  filters:
    # Время
    # формат 2020-08-13 10:02:55.9689259 +0000 UTC
    created: ''
    # Числовые фильтры
    # Пример для числовых фильтров - ^([5-9]\d|\d{3,})$
    event_id: ''
    qualifiers: ''
    record_id: ''
    process_id: ''
    thread_id: ''
    version: ''
    # Строковые фильтры
    # пример: DESKTOP-IDCMV6G
    computer_name: ''
    msg: ''
    # Возможные значения: Information, Warning, Error
    level_text: ''
    # Пример: Service State Event
    task_text: ''
    # Пример: ServiceShutdown
    opcode_text: ''
    # Пример: System
    channel_text: ''
    # Пример: System
    provider_text: ''

    # Возможно применение опций смены кодировки
```

17.11.1.3. Компонент ODBC

Позволяет осуществлять сбор событий из баз данных.

```
odbc_collector: &odbc_collector
  # Уникальный идентификатор компонента, отображается в журналах иметриках.
  # Обязательный параметр
  id: "odbc_collector"
  # Интервал между запуском запроса в секундах
  poll_interval: 5
  # Чтение с последней сохраненной позиции
  read_from_last: true
  # Строка подключения, обязательный параметр
  connection_string: "server=<ip адрес удаленного узла> или <имя удаленного
узла>;port=<порт>;driver=<название драйвера>;database=<название базы данных>;Uid=
<пользователь>;Pwd=<пароль>"
  # SQL запрос, обязательный параметр
  sql: >
    SELECT id, name, dsc
    FROM test WHERE id > ?;
    # Поле, которое будет использоваться как закладка для сохранения позиции
    bookmark_field: "id"

    # Возможно применение опций смены кодировки
```

Примеры строк подключения:

> **PostgreSQL**:
> Driver={PostgreSQL};Server=IP address;Port=5432;Database=myDataBase;Uid=myUsername;Pwd=myPassword;

> **MSSQL**:
> Driver={ODBC Driver 17 for SQL Server};Server=myServerAddress;Database=myDataBase;UID=myUsername;PWD=myPassword;Driver={ODBC Driver 17 for SQL Server};Server=myServerAddress;Database=myDataBase;Trusted_Connection=yes;

> **Oracle**:
> Driver={Oracle ODBC Driver};UID=Kotzwinkle;PWD=whatever;DBQ=instl_alias;DBA=W;

17.11.1.4. Компонент WMI

> **Важно!** Работает только на ОС Windows

```
wmi_collector: &wmi_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "wmi_collector"
  # Интервал между запуском запроса в секундах
  poll_interval: 5
  # Список серверов к которым уйдет wmi запрос, обязательный параметр
  remote_servers:
    - "localhost"
    - "имя удаленного узла"
    - "ip адрес удаленного узла"
  # Имя пользователя, обязательно если это не локальный сбор
  # Для сбора в домене "domain\\user"
  user: "user"
  # Пароль пользователя, обязательно если это не локальный сбор
  password: ""
  # Чтение с последней сохраненной позиции
  read_from_last: true
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
  # Блэклист фильтры по полям события, используются регулярные выражения
  wmi_filters:
    # Числовые поля
    category: ''
    event_code: ''
    event_identifier: ''
    event_type: ''
    record_number: ''
    # Строковые поля
    computer_name: ''
    message: ''
    source_name: ''
    type: ''
    user: ''
    time_generated: ''
    time_written: ''
```

```
  # Возможно применение опций смены кодировки
```

17.11.1.5. Компонент ETW

> **Важно!** Работает только на ОС Windows

```
etw_collector: &etw_collector
  # Имя провайдера или GUID
  # Формат GUID должен быть "{9E814AAD-3204-11D2-9A82-006008A86939}"
  id: "etw_collector"
  provider: "{A68CA8B7-004F-D7B6-A698-07E2DE0F1F5D}"
  kernel_args: [ "ALPC", "CSWITCH", "DBGPRINT", "DISK_FILE_IO", "DISK_IO",
"DISK_IO_INIT", "DISPATCHER", "DPC", "DRIVER", "FILE_IO", "FILE_IO_INIT",
"IMAGE_LOAD", "INTERRUPT", "MEMORY_HARD_FAULTS", "MEMORY_PAGE_FAULTS",
"NETWORK_TCPIP", "NO_SYSCONFIG", "PROCESS", "PROCESS_COUNTERS", "PROFILE",
"REGISTRY", "SPLIT_IO", "SYSTEMCALL", "THREAD", "VAMAP", "VIRTUAL_ALLOC" ]
  provider_level: "Information"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.1.6. Компонент OPSEC LEA

> **Важно!** Работает только на ОС Linux

```
opsec_lea_collector: &opsec_lea_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "opsec_lea_collector"
  # Директория расположения утилиты lea_client
  exec_path: "opsec"
  # Периодичность проверки наличия новых записей в журналах
  poll_interval: 5
  # Сохранение позиции последнего чтения из журнала (сохранение на диск),
возобновление чтения с последней сохраненной позиции.
  read_from_last: false
  # Сервер для сбора событий.
  remote_server: "<ip адрес или имя удаленного узла>"
  # Порт для аутентификации.
  auth_port: 18184
  # Аутентификация для OPSEC
  auth_type: "sslca"
  # Параметры авторизации
  opsec_sic_name: "CN=lea_logger,O=vmfw..ktz7qd"
  opsec_sslca_file: "/home/lea/lea_client/opsec.p12"
  opsec_entity_sic_name: "cn=cp_mgmt,o=vmfw..ktz7qd"
  opsec_sic_policy_file: ""
  # Название собираемого журнала.
  log_filename: "fw.log"
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
```

```
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
```

17.11.1.7. Компонент SSH

```
ssh_collector: &ssh_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "ssh_collector"
  # Имя пользователя для удаленного подключения, обязательный параметр
  user: "user"
  # Список хостов для подключения, обязательный параметр
  remote_servers: ["<ip адрес удаленного узла>", "<имя удаленного узла>"]
  # Порт для подключения (default: 22)
  port: 22
  # Путь к файлу с ssh ключами, обязательный параметр
  rsa: "./ssh"
  # Пароль от файла с ключами
  password: ""
  # Команда для выполнения по ssh, обязательный параметр
  command: "tail -F -n +$$$line$$$ /var/log/sshfile.txt"
  # Если установлено - файл будет читаться с последней позиции в следующем тике
или после перезапуска
  read_from_last: true
  # Интервал между выполнением команд (в секундах)
  ticker: 30
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.1.8. Компонент SMB

```
smb_collector: &smb_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "smb_collector"
  # Список хостов для подключения, обязательный параметр
  remote_servers: ["<ip адрес удаленного узла>", "<имя удаленного узла>"]
  # Порт подключения
  port: 445
  # SMB share. sharename должен соответсвовать формату `<share>` или `\\\\
<server>\\<share>`, обязательный параметр
  share: "<путь к общему ресурсу>"
  # Домен
  domain: "."
  # NTLMv2 пользователь, обязательный параметр
  user: "user"
  # NTLMv2 пароль(или hash), обязательный параметр
  password: "password"
```

```
  # настройки аутентификации kerberos
  kerberos:
    # включение авторизации kerberos
    enabled: false
    # имя целевого сервиса (service principal name)
    target_spn: "pdc"
    # kerberos realm
    realm: "TEST.TEST"
    # путь до конфигурации kerberos
    config_path: "assets/krb5/krb5.conf"
  # Интервал между запуском сканирования файлов в секундах
  poll_interval: 5
  # Список файлов для чтения, обязательный параметр
  files: [ "smbfile.txt" ]
  # Если установлено - использовать регулярное выражение для поиска файлов
  using_regexp: true
  # Начальный каталог для поиска файлов
  regexp_starting_dir: "."
  # Регулярное выражение для поиска файлов
  regexp_expression: ".(?:txt|log)$"
  # Интервал проверки файлов (в секундах) в дереве каталогов
  dir_check_interval: 5
  # Если установлено - файл будет читаться с последней позиции в следующем тике
  # или после перезапуска
  read_from_last: true
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.1.9. Компонент FTP

```
ftp_collector: &ftp_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "ftp_collector"
  # Список хостов для подключения, обязательный параметр
  remote_servers: ["<ip адрес удаленного узла>", "<имя удаленного узла>"]
  # Порт для ftp запросов, обязательный параметр
  port: 21
  # ftp пользователь, обязательный параметр
  user: ""
  # ftp пароль, обязательный параметр
  password: ""
  # Интервал между сканированием файла в секундах
  poll_interval: 5
  # Список файлов для чтения, обязательный параметр
  files: ["ftpfile.txt"]
  # Если установлено - использовать регулярное выражение для поиска файлов
  using_regexp: true
  # Начальный каталог для поиска файлов
  regexp_starting_dir: "."
```

```
  # Регулярное выражение для поиска файлов
  regexp_expression: ".(?:txt|log)$" #"^.*\_logs$"
  # Интервал проверки файлов (в секундах) в дереве каталогов
  dir_check_interval: 5
  # Если установлено - файл будет читаться с последней позиции в следующем тике
  # или после перезапуска
  read_from_last: true
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.1.10. Компонент SFTP

```
sftp_collector: &sftp_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "sftp_collector"
  # Список хостов для подключения, обязательный параметр
  remote_servers: ["<ip адрес удаленного узла>", "<имя удаленного узла>"]
  # Порт для sftp запросов, обязательный параметр
  port: 22
  # Пользователь ssh, обязательный параметр
  user: "user"
  # Пароль ssh, обязательный параметр
  password: "password"
  # Интервал между сканированием файла в секундах
  poll_interval: 5
  # Список файлов для чтения, обязательный параметр
  files: ["sftptest.txt"]
  # Если установлено - использовать регулярное выражение для поиска файлов
  using_regexp: false
  # Начальный каталог для поиска файлов
  regexp_starting_dir: "upload"
  # Регулярное выражение для поиска файлов
  regexp_expression: ".(?:txt|log)$" #"^.*_logs$"
  # Интервал проверки файлов (в секундах) в дереве каталогов
  dir_check_interval: 5
  # Если установлено - файл будет читаться с последней позиции при следующем тике
или после перезапуска
  read_from_last: true
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.1.11. Компонент NetFlow

```
netflow_input: &netflow_input
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "netflow_input"
  # Хост на каком запустится сервер (default: localhost)
  host: "<ip адрес лог-коллектора>"
  # Порт на каком запустится сервер (обязательное)
  port: 2162
  # Размер буфера сообщений (если не задано то берется из SO_RCVBUF)
  sock_buf_size: 0
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень сообщений в логах, могут быть значения - DEBUG, INFO, WARN, ERROR
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.1.12. Компонент TCP

```
tcp_input: &tcp_input\
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "tcp_input"
  # Хост на каком запустится сервер
  host: "<ip адрес лог-коллектора>"
  # Порт на каком запустится сервер
  port: <порт для приема соединений>
  # Включение TLS соединения на сервере
  enable_tls: false
  # Файл с приватным ключом (обязательное поле при включенном TLS)
  key_file: "certs/server.key"
  # Файл с сертификатом должно быть подписанным сертификатом CA (обязательное
поле при включенном TLS)
  cert_file: "certs/server.crt"
  # Файл с паролем если сертификат подписывался с паролем
  cert_key_pass: ""
  # Файл с сертификатом CA (обязательное поле при включенном TLS)
  ca_file: "certs/ca.crt"
  # Проверять ли сертификаты клиента
  require_client_cert: false
  # Нужна ли распаковка тела запроса, ожидается, что клиент упаковал тело запроса
в архив (default: false)
  compression_enabled: false
  # Количество соединений, которые может принять сервер
  connections_limit: 10
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.1.13. Компонент UDP

```
udp_input: &udp_input
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "udp_input"
  # Хост на каком запустится сервер
  host: "<ip адрес лог-коллектора>"
  # Порт на каком запустится сервер
  port: <порт для приема соединений>
  # Размер буфера сообщений (если незаданно то берется из SO_RCVBUF)
  sock_buf_size: 0
  # формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.1.14. Компонент HTTP приемник

```
http_input: &http_input
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "http_input"
  # Хост на каком запустится сервер (default: localhost)
  host: "<ip адрес лог-коллектора>"
  # Порт на каком запустится сервер (обязательное)
  port: <порт для приема соединений>
  # Включение TLS соединения на сервере (default: false)
  enable_tls: false
  # Файл с приватным ключом (обязательное поле при включенном TLS)
  key_file: "certs/server.key"
  # Файл с сертификатом должно быть подписанным сертификатом CA (обязательное
поле при ключенном TLS)
  cert_file: "certs/server.crt"
  # Файл с паролем если сертификат подписывался с паролем
  cert_key_pass: ""
  # Файл с сертификатом CA (обязательное поле при включенном TLS)
  ca_file: "certs/ca.crt"
  # Проверять ли сертификаты клиента (default: false)
  require_client_cert: false
  # Количество соединений, которые может принять сервер
  connections_limit: 10
  # формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.1.15. Компонент HTTP сборщик

```
http_collector: &http_collector\
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "http_collector"
  # Удаленный адрес для вызовов http (обязательное)
  remote_server: "<ip адрес или имя удаленного узла>"
  # Удаленный порт (default: 80)
  port: <порт на целевой системе>
  # Имя пользователя для базовой авторизации, если пустое, считаем, что
авторизация выключена
  basic_auth_user: ""
  # Пароль для базовой авторизации
  basic_auth_password: ""
  # Ограничение по времени для запросов, сделанных http-клиентом в секундах
(default: 10)\
  timeout: 10
  # Если установлено - будет использоваться tls клиент
  enable_tls: false
  # Путь к .key файлу, обязательно если enable_tls: true
  key_file: "certs/server.key"
  # Путь к .crt файлу, обязательно если enable_tls: true
  cert_file: "certs/server.crt"
  # Пароль к файлу сертификатов
  cert_key_pass: ""
  # Путь к файлу с набором корневых центров сертификации, обязательно
  если enable_tls: true
  ca_file: "certs/ca.crt"
  # Имя файла для получения по http
  file: "httptest.txt"
  # Если установлено - файл будет читаться с последней позиции в следующем тике
или после перезапуска (default: false)
  read_from_last: true
  # Интервал между http-вызовами в секундах
  poll_interval: 5
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.1.16. Компонент File

```
file_input: &file_input
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "file_input"
  # Интервал между чтениями файла, в секундах)
  poll_interval: 5
  # Список файлов для чтения
  files: ["logfile.txt"]
```

```
  # Использовать regexp для поиска файлов
  using_regexp: false
  # Начальный каталог для поиска файлов
  regexp_starting_dir: "."
  # regexp для поиска файлов
  regexp_expression: "^.*_logs$"
  # Интервал поиска файлов в секундах, в дереве каталогов
  dir_check_interval: 5
  # Если установлено - файл будет читаться с последней позиции в следующем тике
или после перезапуска\
  read_from_last: true
  # Создает file watchers для всех файлов
  enable_watcher: true
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.1.17. Компонент External Command

```
external_command_input: &external_command_input
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "external_command_input"
  # Интервал между выполнениями команд
  poll_interval: 5
  # Команда bash/cmd
  command: "<команда выполнения скрипта>"
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.1.18. Компонент SNMP Trap

```
snmp_trap: &snmp_trap
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "snmp_trap"
  # Адресс snmp менеджера
  host: "<ip адрес лог-коллектора>"
  # Порт для запуска snmp менеджера
  port: 162
  # Принимать только аутентифицированные SNMP v3 Traps
  allow_authenticated_only: false
  # Список директорий с .mib файлами для конвертации oid
  # Если не указаны, oid будут передаваться в сыром виде
  mib_dirs:
```

```
    - dir1
    - dir2
    - dir3
  # Параметры безопасности
  # Методы аутентификации. Возможные значения:
  # - MD5
  # - SHA
  auth_proto: "SHA"
  # Методы шифрования. Поддерживается только DES.
  encrypt_proto: "DES"
  # Имя SNMP пользователя
  user_name: "user"
  # Пароль аутентификации. Используется с MD5 или SHA
  authentication_passphrase: "user_pass"
  # Пароль шифрования для DES
  privacy_passphrase: "priv_user_pass"
  # Используется в SNMPv3 для идентификации сущностей.
  authoritative_engine_id: "880000009fe71969bdd782bbc691c06b524d70324abe96c0755"
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

17.11.2. Компоненты отправки событий (outputs)

Для включения компонента отправки событий необходимо добавить в файл конфигурации его настройки.

17.11.2.1. Компонент отправки событий по протоколу TCP

Позволяет отправлять данные по протоколу TCP. Так же есть возможность отправки в зашифрованном виде, со сжатием, и настроить буферизацию.

```
tcp_output: &tcp_output
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "tcp_output"
  # Адрес куда отправлять события, обязательный параметр
  target_host: "<ip адрес или имя удаленного узла>"
  # Порт куда отправлять события, обязательный параметр
  port: <порт на целевой системе>
  # Включение batch режима
  batch_mode_enable: false
  # Период отправки пакета в секунундах при включенном batch режиме
  batch_flush_interval: 5
  # Количество сообщений, которые попадут в пакет при включенном batch режиме
  batch_flush_limit: 500
  # Включение сжатия, включение при выключенном batch режиме ощутимо замедляет
отправку
  ssl_compression: false
  # Включение проверки сертификата
  require_cert: false
```

```
  # Включение ssl\
  ssl_enable: false
  # Путь для файла сертификатов, если enable_tls: false параметр не обязательный
  cert_file: "client-cert.pem"
  # Путь для файла ключей, если enable_tls: false параметр не обязательный
  key_file: "client-key.pem"
  # Пароль для расшифровки файла ключей, если не указан считаем, что файл не
зашифрован
  cert_key_pass: ""
  # Путь до корневого сертификата, если enable_tls: false не обязательный
параметр
  ca_file: "ca.pem"
  # уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
  # максимальное количество сообщений в буфере
  #queue_length_limit: 1500
  # максимальное время жизни событий в очереди. в секундах
  #queue_time_limit: 300
```

17.11.2.2. Компонент отправки событий по протоколу UDP

Позволяет отправлять данные по протоколу UDP.

```
udp_output: &udp_output
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "udp_output"
  # Адрес куда отправлять события
  target_host: "<ip адрес или имя удаленного узла>"
  # Порт, на который отправлять события. Обязательный параметр
  port: <порт на целевой системе>
  # Размер буфера для отправки, если не указан или равен нулю используется
системное значение
  sock_buf_size: 0
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
```

17.11.2.3. Компонент отправки событий в KAFKA

Позволяет отправлять данные в KAFKA.

```
kafka_output: &kafka_output
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "kafka_output"
  # Включение проверки сертификата (default: false)
  require_cert: false
  # Включение ssl (default: false)
  ssl_enable: false
  # Путь для файла сертификатов, если ssl_enable: false параметр не обязательный
  cert_file: "certs/server.crt"
  # Путь для файла ключей, если ssl_enable: false параметр не обязательный
```

```
  key_file: "certs/server.key"
  # Пароль для расшифровки файла ключей, если не указан считаем, что файл не
зашифрован
  cert_key_pass: ""
  # Путь до корневого сертификата, если ssl_enable: false параметр не
обязательный
  ca_file: "certs/ca.crt"
  # Таймауту отправки события в секундах, обязательный параметр
  timeout: 10
  # Топик в который попадет событие, обязательный параметр
  topic: "<наименование топика>"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
  # Kafka брокеры, обязательный параметр
  brokers:
    - "<ip адрес или имя удаленного узла>:9092"
  # Максимальное количество сообщений в буфере
  #queue_length_limit: 1500
  # Максимальное время жизни событий в очереди (в секундах)
  #queue_time_limit: 300
```

17.11.2.4. Компонент записи событий в локальный файл

Позволяет записать входящие события в локальный файл.

```
out_file: &out_file
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "file_output"
  # Путь до файла куда будут записываться события, обязательный параметр
  file: "ouput_file.txt"
  # Порог ротации в мегабайтах, если указан ноль ил не указан совсем ротация не
происходит
  rotation_size: 10
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
```

17.12. Включение компонентов

Для того чтобы включить *компоненты* сбора или отправки, необходимо добавить их в разделах настроек collectors для *компонентов* сбора или senders для *компонентов* отправки в конфигурационном файле Лог-коллектора.

17.12.1. Включение компонентов сбора (collectors)

В разделе collectors необходимо прописать следующие настройки (пример для *компонента* сбора eventlog):

```
collectors:
  # Уровень логирования, если не указан используется уровень логирования
компонента журналирования\
  log_level: "INFO"
  # eventlog коллектор, работает только на windows vista и старше
  event_log:
  - <<: *eventlog_collector
```

17.12.2. Включение компонентов отправки (senders)

В разделе senders необходимо прописать следующие настройки (пример для *компонента* отправки по TCP):

```
senders:
  # Порт компонента, обязательный параметр
  port: 48002
  # Уровень логирования, если не указан используется уровень логирования
компонента журналирования
  log_level: "INFO"
  # Отправка по протоколу tcp
  tcp:
    - <<: *tcp_output
```

17.13. Маршрутизация событий

Для организации маршрутизации необходимо выполнить следующие действия для связывания *компонентов* сбора и *компонентов* отправки:

1. Настроить маршруты взаимодействия между *компонентами* сбора событий и *компонентами* отправки событий.

Пример настройки маршрута:

```
route_1: &route_1
  collector_id:
    - "eventlog_collector"
    - "tcp_input"
  sender_id:
    - "tcp_output"
```

2. включить маршрут в разделе конфигурационного файла routers.

Пример включения маршрута:

```
routers:
  - <<: *route_1
```

> **Важно!** *Компоненты*, используемые в маршрутах, обязательно должны быть включены в разделах **collectors** и **senders** и настроены.

18. Управление лог-коллектором из веб-интерфейса Платформы

Управление экземплярами Лог-коллектора, которые были настроены для централизованного управления с помощью директивы cluster в конфигурационном файле, осуществляется в интерфейсе Платформы в разделе «**Администрирование**» — «**Кластер**» на вкладке «**Узлы системы**» пункт «**Проверка**» (Рисунок 1).

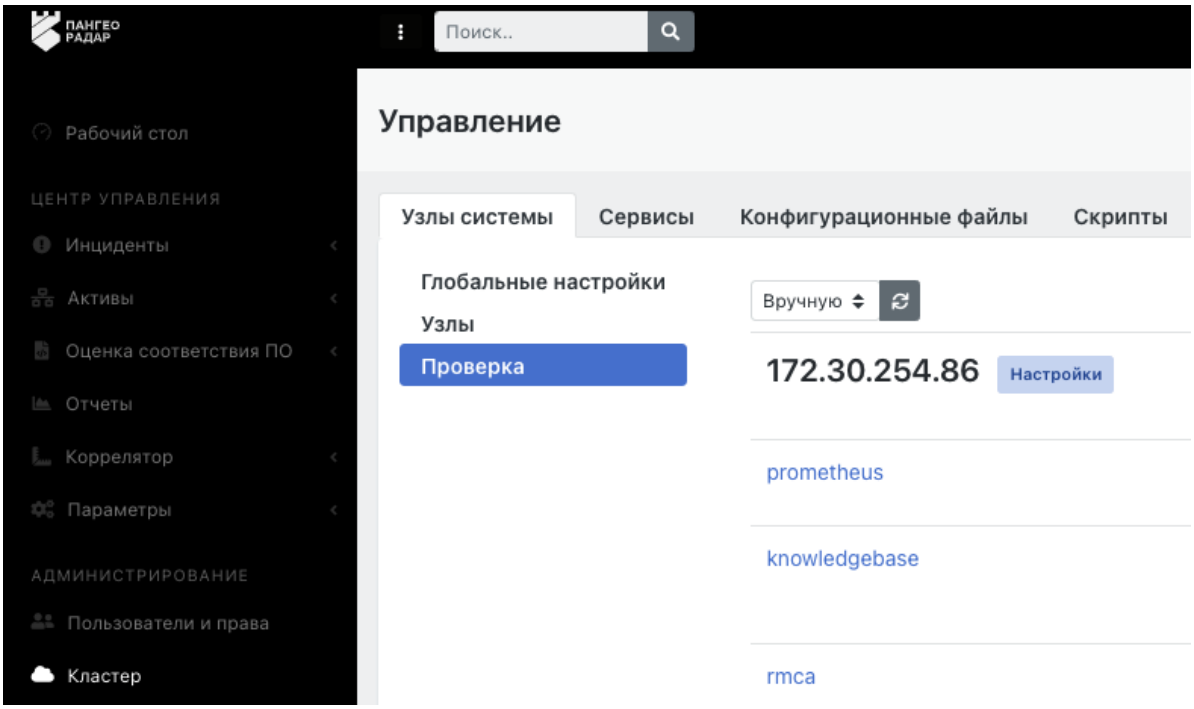

Рисунок 55

Необходимо выбрать нужный узел и нажать кнопку «**Настройка**»

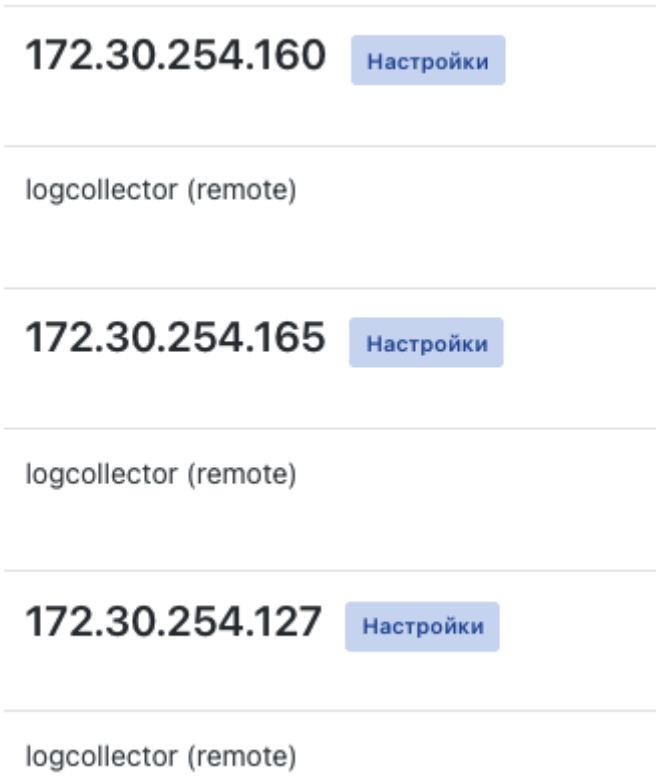

Рисунок 56

Откроется страница управления узлом

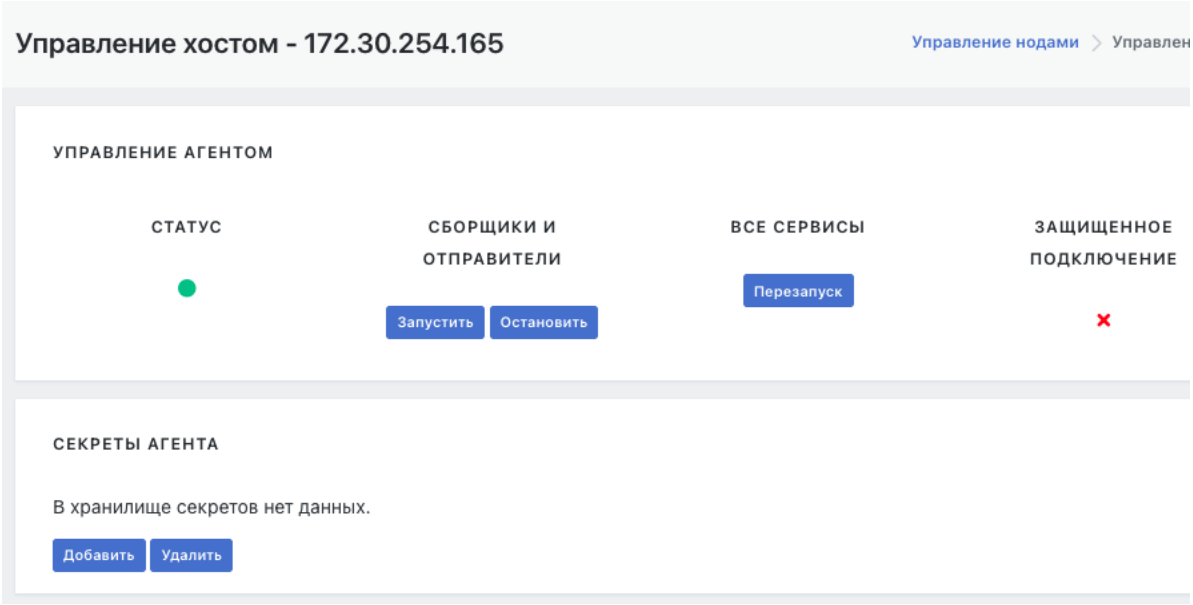

Рисунок 57

На странице Управление лог-коллектором доступны следующие действия:

- Остановка и запуск компонентов сбора и отправки
- Перезапуск всех сервисов
- Загрузить и отредактировать конфигурационный файл удаленного лог-коллектора
- Сохранить конфигурационный файл на удаленный лог-коллектор

Для загрузки конфигурационного файла необходимо нажать кнопку «**Загрузить**»

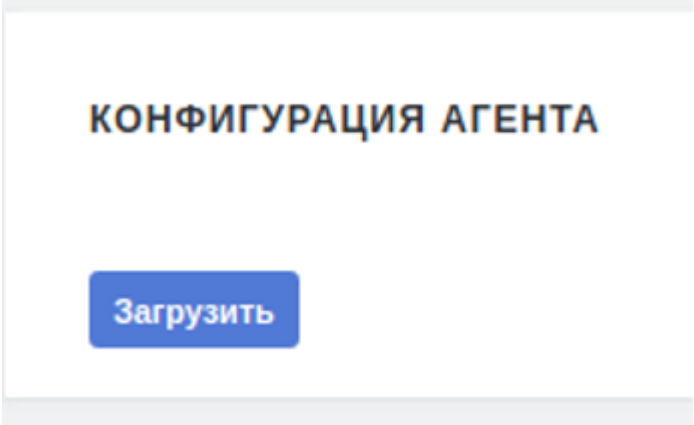

Рисунок 58

После нажатия кнопки Загрузить появится возможность редактирования конфигурационного файла на удаленном лог-коллекторе.

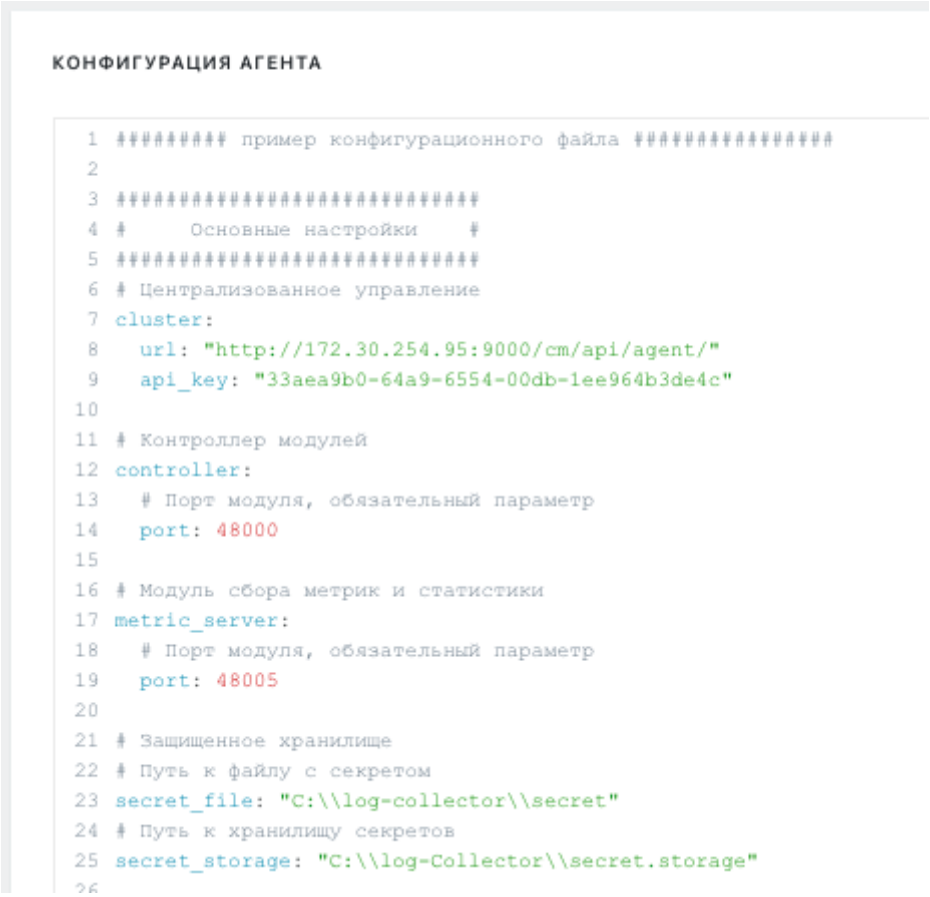

Рисунок 59

После внесения изменений в конфигурационный файл, его необходимо загрузить на лог-коллектор с помощью нажатия кнопки «**Сохранить**».

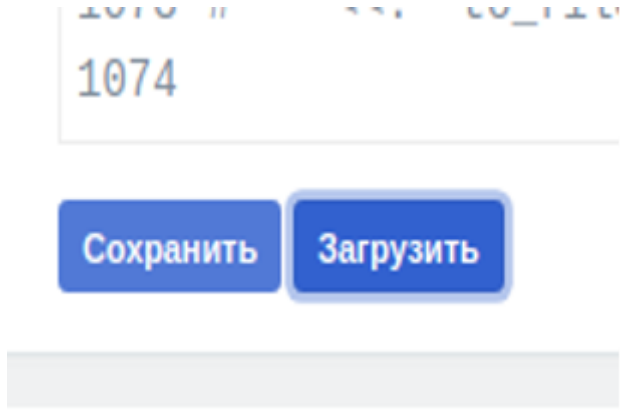

Рисунок 60

После чего необходимо перезапустить все сервисы удаленного лог-коллектора нажав на кнопку «**Перезапуск**»

19. Пример конфигурационного файла лог-коллектора

```
######################
# Основные настройки #
######################
# Централизованное управление
cluster:
```

```
  url: "http://<ip адрес Платформы Радар>:9000/cm/api/agent/"
  api_key: "<ключ API>"

# Контроллер компонентов
controller:
  # Порт компонента, обязательный параметр
  port: 48000

# Компонент сбора метрик и статистики
metric_server:
  # Порт компонента, обязательный параметр
  port: 48005

# Защищенное хранилище
# Путь к файлу с секретом
secret_file: "C:\\log-collector\\secret"
# Путь к хранилищу секретов
secret_storage: "C:\\log-Collector\\secret.storage"

# Компонент API
api_server:   # ip адрес, на котором будем слушать http сервер
  address: "<Внешний ip адрес сервера, на котором установлен лог-коллектор>"
  # Порт на котором будем слушать http сервер, обязательный параметр
  port: 8080
  # Таймаут чтения (получение запроса), обязательный параметр
  read_timeout: 60
  # Таймаут записи (отправка запроса), обязательный параметр
  write_timeout: 60
  # Время ожидания окончания обработки запроса при получении сигнала на остановку
приложения
  # Обязательный параметр
  wait: 5
  # Включение https (защищенного соединения)
  enable_tls: false
  # Путь для файла сертификатов, если enable_tls: false параметр не обязательный
  cert_file: "certs/server.crt"
  # Путь для файла ключей, если enable_tls: false параметр не обязательный
  key_file: "certs/server.key"
  # Пароль для расшифровки файла ключей, если не указан считаем, что файл не
зашифрован
  cert_key_pass: ""
  # Включение проверки клиентского сертификата, обязательный параметр
  require_client_cert: true
  # Путь до корневого сертификата, обязательный параметр
  ca_file: "certs/ca.crt"
  # Уровень логирования, если не указан используется указанный в компоненте
журналирования
  log_level: "INFO"

# Компонент журналирования
journal:\
  # Порт компонента, обязательный параметр\
  port: 48004
  # Еровень логирования по умолчанию. Возможные значения - DEBUG, INFO, WARN,
ERROR.
```

```
  #Обязательный параметр
  log_level: "INFO"
  # Путь к файлу журнала, обязательный параметр
  log_path: "C:\\log-collector\\journal.log"
  # Порог ротации файла логов, указывается в мегабайтах, обязательный параметр
  rotation_size: 30
  # Порог количества файлов истории, если не указано файлы удаляться не будут
  max_backups: 7
  # Максимальное количество дней для хранения старых файлов журнала на основе
метки времени
  # Если не указано, файлы удаляться не будут (в днях).
  max_age: 7

############
# outputs #
############
  # Отправки событий в KAFKA
  kafka_output: &kafka_output
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "kafka_output"
  # Включение проверки сертификата (default: false)
  require_cert: fals
  # Включение ssl (default: false)
  ssl_enable: false
  # Путь для файла сертификатов, если ssl_enable: false параметр не обязательный
  cert_file: "certs/server.crt"
  # Путь для файла ключей, если ssl_enable: false параметр не обязательный
  key_file: "certs/server.key"
  # Пароль для расшифровки файла ключей, если не указан считаем что файл не
зашифрован
  cert_key_pass: ""
  # Путь до корневого сертификата, если ssl_enable: false параметр не
обязательный
  ca_file: "certs/ca.crt"
  # Таймауту отправки события в секундах, обязательный параметр
  timeout: 10
  # Топик в который попадет событие, обязательный параметр
  topic: "<наименование топика>"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
  # Kafka брокеры, обязательный параметр
  brokers:
    - "<ip адрес или имя удаленного узла>:9092"
  # Максимальное количество сообщений в буфере
  queue_length_limit: 3
  # Максимальное время жизни событий в очереди (в секундах)
  queue_time_limit: 3

# Вывод в файл
out_file: &out_file
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "file_output"
```

```
  # Путь до файла куда будут записываться события, обязательный параметр
  file: "ouput_file.txt"
  # Порог ротации в мегабайтах, если указан ноль ил не указан совсем ротация не
происходит (default: 0)
  rotation_size: 10
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

# Отправка по протоколу udp
  udp_output: &udp_output
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "udp_output"
  # Адрес куда отправлять события
  target_host: "<ip адрес или имя удаленного узла>"
  # Порт, на который отправлять события. Обязательный параметр
  port: <порт на целевой системе>
  # Размер буфера для отправки, если не указан или равен нулю используется
системное значение (default: 0)
  sock_buf_size: 0
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

# Отправка по протоколу tcp
tcp_output: &tcp_output
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "tcp_output"
  # Адрес куда отправлять события, обязательный параметр
  target_host: "<ip адрес или имя удаленного узла>"
  # Порт куда отправлять события, обязательный параметр
  port: <порт на целевой системе>
  # Включение batch режима (default: false)
  batch_mode_enable: false
  # Период отправки пакета в секунундах при включенном batch режиме (default: 5)
  batch_flush_interval: 5
  # Количество сообщений которые попадут в пакет при включенном batch режиме
(default: 500)
  batch_flush_limit: 500
  # Включение сжатия, включение при выключенном batch режиме ощутимо замедляет
отправку (default: false)
  ssl_compression: false
  # Включение проверки сертификата (default: false)
  require_cert: false
  # Включение ssl (default: false)
  ssl_enable: false
  # Путь для файла сертификатов, если enable_tls: false параметр не обязательный
  cert_file: "client-cert.pem"
  # Путь для файла ключей, если enable_tls: false параметр не обязательный
  key_file: "client-key.pem"
  # Пароль для расшифровки файла ключей, если не указан считаем что файл не
зашифрован
  cert_key_pass: ""
```

```
  # Путь до корневого сертификата, если enable_tls: false не обязательный
параметр
  ca_file: "ca.pem"
  # уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
  # максимальное количество сообщений в буфере
  queue_length_limit: 1500
  # максимальное время жизни событий в очереди. в секундах
  queue_time_limit: 300

##########
# inputs #
##########
# Работает только на ОС Windows
eventlog_collector: &eventlog_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "eventlog_collector"
  # Имя канала (Security, Application, System, ForwardedEvents)
  # Используется если не указан путь к файлу
  channel: ['Security']
  # Запрос описывающий тип получаемого события. Может быть в формате
  # XPath 1.0 или структурированный XML запрос. Если XPath содержит более 20
параметров,   # следует использовать структурированный XML запрос.
  # Чтобы получить все параметры укажите "*"
  query: "*"
  # Полный путь к файлу журналов событий
  # Поддерживаемые форматы: .evt, .evtx, .etl
  file: ""
  # Размер запроса
  batch_size: 50
  # Таймаут запроса в секунда
  timeout: 5
  # Интервал между запуском запроса в секундах
  poll_interval: 30
  # Чтение с последней сохраненной позиции
  read_from_last: true
  # Конвертировать SID в имя
  resolve_sid: false
  # формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования. Если не указан используется уровень компонента
журналирования
  log_level: "INFO"
  # Параметры удаленного подключения
  remote:
    # Включение удаленного соединения
    enabled: false
    # Имя пользователя, обязательно если enabled: true
    user: ""
    # Пароль пользователя, обязательно если enabled: true
    password: ""
    # Домен пользователя
    domain: "."
```

```
    # Адрес удаленного сервера
    remote_servers: ["<ip адрес удаленного узла>", "<имя удаленного узла>"]
    # Доступные методы авторизации: Negotiate, Kerberos, NTLM
    auth_method: "Negotiate"
  # Фильтрация по полям события, регулярные выражения
  filters:
    # Время. Формат 2020-08-13 10:02:55.9689259 +0000 UTC
    created: ''
    # Числовые фильтры
    # Пример для числовых фильтров - ^([5-9]\d|\d{3,})$
    event_id: ''
    qualifiers: ''
    record_id: ''
    process_id: ''
    thread_id: ''
    version: ''
    # Строковые фильтры
    # пример: DESKTOP-IDCMV6G
    computer_name: ''
    msg: ''
    # Возможные значения: Information, Warning, Error
    level_text: ''
    # Пример: Service State Event
    task_text: ''
    # Пример: ServiceShutdown
    opcode_text: ''
    # Пример: System
    channel_text: ''
    # Пример: System
    provider_text: ''

  # Возможно применение опций смены кодировки
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

# Работает только на ОС Windows XP, 2003
eventlog_xp_collector: &eventlog_xp_collector
  # Название компонента, отображается в логах и метриках, уникальный
  # Обязательный параметр
  id: "eventlog_xp_collector"
  # Имя канала
  channel: ['Security']
  # Интервал между запуском запроса в секундах
  poll_interval: 5
  # Чтение с последней сохраненной позиции
  read_from_last: false
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
  # Фильтрация по полям события, регулярные выражения
  filters:
```

```
    # Время
    # формат 2020-08-13 10:02:55.9689259 +0000 UTC
    created: ''
    # Числовые фильтры
    # Пример для числовых фильтров - ^([5-9]\d|\d{3,})$
    event_id: ''
    qualifiers: ''
    record_id: ''
    process_id: ''
    thread_id: ''
    version: ''
    # Строковые фильтры
    # пример: DESKTOP-IDCMV6G
    computer_name: ''
    msg: ''
    # Возможные значения: Information, Warning, Error
    level_text: ''
    # Пример: Service State Event
    task_text: ''
    # Пример: ServiceShutdown
    opcode_text: ''
    # Пример: System
    channel_text: ''
    # Пример: System
    provider_text: ''

  # Возможно применение опций смены кодировки
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

odbc_collector: &odbc_collector
  # Названия компонента, отображается в логах и метриках, уникальный обязательный
параметр
  id: "odbc_collector"
  # Интервал между запуском запроса в секундах
poll_interval: 5
  # Чтение с последней сохраненной позиции (default: false)
read_from_last: true
  # Строка подключения, обязательный параметр
connection_string: "server=<ip адрес или имя удаленного узла>;port=<порт>;driver=
<названиедрайвера>;database=<название базыданных>;Uid=<пользователь>;Pwd=
<пароль>"
  # SQL запрос, обязательный параметр
  sql: \>
    SELECT id, name, dsc
    FROM test WHERE id \> ?;
  # Поле, которое будет использоваться как закладка для сохраненияпозиции,
обязательный параметр
  bookmark_field: "id"

  # Возможно применение опций смены кодировки
```

```
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

# Работает только на ОС Windows
wmi_collector: &wmi_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "wmi_collector"
  # интервал между запуском запроса в секундах
  poll_interval: 5
  # Список серверов к которым уйдет wmi запрос, обязательный параметр
  remote_servers:
    - "localhost"
    - "<имя удаленного узла>"
    - "<ip адрес удаленного узла>"
  # Имя пользователя, обязательно если это не локальный сбор
  # Для сбора в домене "domain\\user"
  user: "user"
  # Пароль пользователя, обязательно если это не локальный сбор
  password: ""
  # Чтение с последней сохраненной позиции
  read_from_last: true
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
  # Блэклист фильтры по полям события, используются регулярные выражения
  wmi_filters:
    # Числовые поля
    category: ''
    event_code: ''
    event_identifier: ''
    event_type: ''
    record_number: ''
    # Строковые поля
    computer_name: ''
    message: ''
    source_name: ''
    type: ''
    user: ''
    time_generated: ''
    time_written: ''

  # Возможно применение опций смены кодировки
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

# Работает только на ОС Windows
```

```
etw_collector: &etw_collector
  # Уникальный идентификатор компонента, отображается в журналах иметриках.
  # Обязательный параметр
  id: "etw_collector"
  # Имя провайдера или GUID
  # Формат GUID должен быть "{9E814AAD-3204-11D2-9A82-006008A86939}"
  provider: "{A68CA8B7-004F-D7B6-A698-07E2DE0F1F5D}"
  kernel_args: [ "ALPC", "CSWITCH", "DBGPRINT", "DISK_FILE_IO", "DISK_IO",
"DISK_IO_INIT", "DISPATCHER", "DPC", "DRIVER", "FILE_IO", "FILE_IO_INIT",
"IMAGE_LOAD","INTERRUPT", "MEMORY_HARD_FAULTS", "MEMORY_PAGE_FAULTS",
"NETWORK_TCPIP", "NO_SYSCONFIG", "PROCESS", "PROCESS_COUNTERS", "PROFILE",
"REGISTRY", "SPLIT_IO", "SYSTEMCALL", "THREAD", "VAMAP", "VIRTUAL_ALLOC" ]
  provider_level: "Information"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
  # Возможно применение опций смены кодировки
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

# Работает только на ОС Linux
opsec_lea_collector: &opsec_lea_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "opsec_lea_collector"
  # Директория расположения утилиты lea_client.
  exec_path: "opsec"
  # Периодичность проверки наличия новых записей в журналах.
  poll_interval: 5
  # Сохранение позиции последнего чтения из журнала (сохранение на диск),
возобновление чтения с последней сохраненной позиции.
  read_from_last: false
  # Сервер для сбора событий.
  remote_server: "<ip адрес или имя удаленного узла>"
  # Порт для аутентификации.
  auth_port: 18184
  # Аутентификация для OPSEC.
  auth_type: "sslca"
  # Параметры авторизации.
  opsec_sic_name: "CN=lea_logger,O=vmfw..ktz7qd"
  opsec_sslca_file: "/home/lea/lea_client/opsec.p12"
  opsec_entity_sic_name: "cn=cp_mgmt,o=vmfw..ktz7qd"
  opsec_sic_policy_file: ""
  # Название собираемого журнала.
  log_filename: "fw.log"
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
```

```
ssh_collector: &ssh_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "ssh_collector"
  # Имя пользователя для удаленного подключения, обязательный параметр
  user: "user"
  # Список хостов для подключения, обязательный параметр
  remote_servers: ["<ip адрес удаленного узла>", "<имя удаленного узла>"]
  # Порт для подключения (default: 22)
  port: 22
  # Путь к файлу с ssh ключами, обязательный параметр
  rsa: "./ssh"
  # Пароль от файла с ключами
  password: ""
  # Команда для выполнения по ssh, обязательный параметр
  command: "tail -F -n +$$$line$$$ /var/log/sshfile.txt"
  # Если установлено - файл будет читаться с последней позиции при следующем тике
или после перезапуска
  read_from_last: true
  # Интервал между выполнением команд(в секундах)
  ticker: 30
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
  # Возможно применение опций смены кодировки
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

smb_collector: &smb_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "smb_collector"
  # Список хостов для подключения, обязательный параметр
  remote_servers: ["<ip адрес удаленного узла>", "<имя удаленного узла>"]
  # Порт подключения
  port: 445
  # SMB share. sharename должен соответсвовать формату `<share>` или `\\\\
<server>\\<share>`, обязательный параметр
  share: "<путь к общему ресурсу>"
  # Домен
  domain: "."
  # NTLMv2 пользователь, обязательный параметр
  user: "user"
  # NTLMv2 пароль(или hash), обязательный параметр
  password: "password"
  # настройки аутентификации kerberos
  kerberos:
    # Включение авторизации kerberos
    enabled: false
```

```
    # имя целевого сервиса (service principal name)
    target_spn: "pdc"
    # kerberos realm
    realm: "PANGEO.LOCAL"
     # путь до конфигурации kerberos
    config_path: "assets/krb5/krb5.conf"
  # Интервал между запуском сканирования файлов в секундах
  poll_interval: 5
  # Список файлов для чтения, обязательный параметр
  files: [ "smbfile.txt" ]
  # Если установлено - использовать регулярное выражение для поиска файлов
  using_regexp: true
  # Начальный каталог для поиска файлов
  regexp_starting_dir: "."
  # Регулярное выражение для поиска файлов
  regexp_expression: ".(?:txt\|log)$"
  # Интервал проверки файлов (в секундах) в дереве каталогов
  dir_check_interval: 5
  # Если установлено - файл будет читаться с последней позиции в
  следующем тике или после перезапуска
  read_from_last: true
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

ftp_collector: &ftp_collector
  # Уникальный идентификатор компонента, отображается в журналах иметриках.
  # Обязательный параметр
  id: "ftp_collector"
  # Список хостов для подключения, обязательный параметр
  remote_servers: ["<ip адрес удаленного узла>", "<имя удаленного узла>"]
  # Порт для ftp запросов, обязательный параметр
  port: 21
  # ftp пользователь, обязательный параметр
  user: ""
  # ftp пароль, обязательный параметр
  password: ""
  # Интервал между сканированием файла в секундах
  poll_interval: 5
  # Список файлов для чтения, обязательный параметр
  files: ["ftpfile.txt"]
  # Если установлено - использовать регулярное выражение для поиска файлов
  using_regexp: true
  # Начальный каталог для поиска файлов
  regexp_starting_dir: "."
```

```
  # Регулярное выражение для поиска файлов
  regexp_expression: ".(?:txt|log)$" #"^.*_logs$"
  # Интервал проверки файлов (в секундах) в дереве каталогов (default: 5)
  dir_check_interval: 5
  # Если установлено - файл будет читаться с последней позиции в следующем тике
или после перезапуска
  read_from_last: true
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

sftp_collector: &sftp_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "sftp_collector"
  # Список хостов для подключения, обязательный параметр
  remote_servers: ["<ip адрес удаленного узла>", "<имя удаленного узла>"]
  # Порт для sftp запросов, обязательный параметр
  port: 22
  # Пользователь ssh, обязательный параметр
  user: "user"
  # Пароль ssh, обязательный параметр
  password: "password"
  # Интервал между сканированием файла в секундах
  poll_interval: 5
  # Список файлов для чтения, обязательный параметр
  files: ["sftptest.txt"]
  # Если установлено - использовать регулярное выражение для поиска файлов
  using_regexp: false
  # Начальный каталог для поиска файлов
  regexp_starting_dir: "upload"
  # Регулярное выражение для поиска файлов
  regexp_expression: ".(?:txt\|log)\$" #"\^.*\_logs\$"
  # Интервал проверки файлов (в секундах) в дереве каталогов
  dir_check_interval: 5
  # Если установлено - файл будет читаться с последней позиции в следующем тике
или после перезапуска
  read_from_last: true
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

```
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

netflow_input: &netflow_input
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "netflow_input"
  # Хост на каком запустится сервер (default: localhost)
  host: "<ip адрес лог-коллектора>"
  # Порт на каком запустится сервер (обязательное)
  port: 2162
  # Размер буфера сообщений (если не задано то берется из SO_RCVBUF)
  sock_buf_size: 0
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень сообщений в логах, могут быть значения - DEBUG, INFO, WARN, ERROR
  log_level: "INFO"

tcp_input: &tcp_input
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "tcp_input"
  # Хост на каком запустится сервер
  host: "<ip адрес лог-коллектора>"
  # Порт на каком запустится сервер
  port: <порт для приема соединений>
  # Включение TLS соединения на сервере
  enable_tls: false
  # Файл с приватным ключом (обязательное поле при включенном TLS)
  key_file: "certs/server.key"
  # Файл с сертификатом должно быть подписанным сертификатом CA (обязательное
поле при включенном TLS)
  cert_file: "certs/server.crt"
  # Файл с паролем если сертификат подписывался с паролем
  cert_key_pass: ""
  # Файл с сертификатом CA (обязательное поле при включенном TLS)
  ca_file: "certs/ca.crt"
  # Проверять ли сертификаты клиента
  require_client_cert: false
  # Нужна ли распаковка тела запроса, ожидается, что клиент упаковал тело запроса
в архив (default: false)
  compression_enabled: false
  # Количество соединений которые может принять сервер
  connections_limit: 10
  # формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
```

```
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

udp_input: &udp_input
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "udp_input"
  # Хост на каком запустится сервер
  host: "<ip адрес лог-коллектора>"
  # Порт на каком запустится сервер
  port: <порт для приема соединений>
  # Размер буфера сообщений (если незаданно то берется из SO_RCVBUF)
  sock_buf_size: 0
  # формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

http_input: &http_input
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "http_input"
  # Хост на каком запустится сервер (default: localhost)
  host: "<ip адрес лог-коллектора>"
  # Порт на каком запустится сервер (обязательное)
  port: <порт для приема соединений>
  # Включение TLS соединения на сервере (default: false)
  enable_tls: false
  # Файл с приватным ключом (обязательное поле при включенном TLS)
  key_file: "certs/server.key"
  # Файл с сертификатом должно быть подписанным сертификатом CA (обязательное
поле при включенном TLS)
  cert_file: "certs/server.crt"
  # Файл с паролем если сертификат подписывался с паролем
  cert_key_pass: ""
  # Файл с сертификатом CA (обязательное поле при включенном TLS)
  ca_file: "certs/ca.crt"
  # Проверять ли сертификаты клиента (default: false)
  require_client_cert: false
  # Количество соединений которые может принять сервер
  connections_limit: 10
  # формат отправки сообщения - как есть(raw), с обогащением(json)
```

```
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

http_collector: &http_collector
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "http_collector"
  # Удаленный адрес для вызовов http (обязательное)
  remote_server: "<ip адрес или имя удаленного узла>"
  # Удаленный порт (default: 80)
  port: <порт на целевой системе>
  # Имя пользователя для базовой авторизации, если пустое, считаем, что
авторизация выключена
  basic_auth_user: ""
  # Пароль для базовой авторизации
  basic_auth_password: ""
  # Ограничение по времени для запросов, сделанных http-клиентом в секундах
(default: 10)
  timeout: 10
  # Если установлено - будет использоваться tls клиент
  enable_tls: false
  # Путь к .key файлу, обязательно если enable_tls: true
  key_file: "certs/server.key"
  # Путь к .crt файлу, обязательно если enable_tls: true
  cert_file: "certs/server.crt"
  # Пароль к файлу сертификатов
  cert_key_pass: ""
  # Путь к файлу с набором корневых центров сертификации. обязательный параметр,
если enable_tls: true
  ca_file: "certs/ca.crt"
  # Имя файла для получения по http
  file: "httptest.txt"
  # Если установлено - файл будет читаться с последней позиции в следующем тике
или после перезапуска (default: false)
  read_from_last: true
  # Интервал между http-вызовами в секундах
  poll_interval: 5
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
  # Опции смены кодировки
```

```
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

file_input: &file_input
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "file_input"
  # Интервал между чтениями файла, в секундах
  poll_interval: 5
  # Список файлов для чтения
  files: ["logfile.txt"]
  # Использовать regexp для поиска файлов
  using_regexp: false
  # Начальный каталог для поиска файлов
  regexp_starting_dir: "."
  # regexp для поиска файлов
  regexp_expression: "\^.*\_logs\$"
  # Интервал поиска файлов в секундах, в дереве каталогов (default: 2)
  dir_check_interval: 2
  # Если установлено - файл будет читаться с последней позиции в следующем тике
или после перезапуска
  read_from_last: true
  # Создает file watchers для всех файлов
  enable_watcher: true
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: true
    # Кодировка оригинала
    original_encoding: "cp1251"

external_command_input: &external_command_input
  # Уникальный идентификатор компонента, отображается в журналах и метриках.
  # Обязательный параметр
  id: "external_command_input"
  # Интервал между выполнениями команд
  poll_interval: 5
  # Команда bash/cmd
  command: "<команда выполнения скрипта>"
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"
```

```
  # Возможно применение опций смены кодировки
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: true
    # Кодировка оригинала
    original_encoding: "cp1251"

snmp_trap: &snmp_trap
  # Уникальный идентификатор компонента, отображается в журналах и метиках.
  # Обязательный параметр
  id: "snmp_trap"
  # Адресс snmp менеджера
  host: "<ip адрес лог-коллектора>"
  # Порт для запуска snmp менеджера
  port: 162
  # Принимать только аутентифицированные SNMP v3 Traps
  allow_authenticated_only: false
  # Список директорий с .mib файлами для конвертации oid
  # Если не указаны, oid будут передаваться в сыром виде
  mib_dirs:
    - dir1
    - dir2
    - dir3
  # Параметры безопасности
  # Методы аутентификации. Возможные значения:
  # - MD5
  # - SHA
  auth_proto: "SHA"
  # Методы шифрования. Поддерживается только DES.
  encrypt_proto: "DES"
  # Имя SNMP пользователя
  user_name: "admin"
  # Пароль аутентификации. Используется с MD5 или SHA
  authentication_passphrase: "admin"
  # Пароль шифрования для DES
  privacy_passphrase: "admin"
  # Используется в SNMPv3 для идентификации сущностей.
  authoritative_engine_id: "880000009fe71969bdd782bbc691c06b524d70324abe96c0755"
  # Формат отправки сообщения - как есть(raw), с обогащением(json)
  format: "raw"
  # Уровень логирования, если не указан используется уровень компонента
журналирования
  log_level: "INFO"

  # Возможно применение опций смены кодировки
  # Опции смены кодировки
  encoding:
    # Использовать кодировку в UTF-8
    change_to_utf8: false
    # Кодировка оригинала
    original_encoding: "cp1251"

###############
# collectors #
```

```
##############
# Включение компонентов сбора событий
# Обязательная секция
collectors:
  log_level: "INFO"
  # opsec_lea коллектор
  #opsec_lea:
    # - <<: *opsec_lea
  # etw коллектор, работает только на windows
  #etw:
    # - <<: *etw_collector
  # чтение из локального файла
  #files:
    # - <<: *file_input
  # коллектор выполняющий сторонней командой
  #external_command:
    # - <<: *external_command_collector
  # wmi коллектор, работает только на windows
  #wmi:
    # - <<: *wmi_collector
  # event_log коллектор, работает только на windows vista и старше
  #event_log:
    # - <<: *eventlog_collector
  # event_log_xp коллектор, работает только на windows xp
  #event_log_xp:
    # - <<: *eventlog_xp_collector
  # odbc коллектор
  #odbc:
    # - <<: *odbc_collector
  # ssh коллектор
  #ssh:
    # - <<: *ssh_collector
  # smb коллектор
  #smb:
    # - <<: *smb_collector
  # ftp коллектор
  #ftp:
    # - <<: *ftp_collector
  # sftp коллектор
  #sftp:
    # - <<: *sftp_collector
  # tcp коллектор, пассивный прием
  #tcp_receiver:
    # - <<: *tcp_input
  # udp коллектор, пассивный прием
  #udp_receiver:
    # - <<: *udp_input
  # netflow коллектор, пассивный прием
  #nf_receiver:
    # - <<: *netflow_input
  # http коллектор, пассивный прием
  #http_receiver:
    # - <<: *http_input
  # http коллектор, удаленный сбор событий
  #http_collector:
```

```
    # - <<: *http_collector
  # snmp trap коллектор, пассивный прием
  #snmp_trap:
    # - <<: *snmp_trap

############
# senders #
############
# Включение компонентов отправки событий
# Обязательная секция
senders:
  # Порт компонента, обязательный параметр
  port: 48002
  # Уровень логирования, если не указан используется уровень логирования
компонента журналирования
  log_level: "INFO"
  # Отправка в журнал
  #stdout:
    # Названия компонента, отображается в логах и метриках, уникальный
обязательный параметр
    # - id: "stdout"
  # Запись в файл
  #out_file:
    # - <<: *out_file
  # Отправка в kafka
  #kafka:
    # - <<: *kafka_output
  # Отправка по протоколу tcp
  #tcp:
    # - <<: *tcp_output
  # Отправка по протоколу udp
  #udp:
    # - <<: *udp_output

###########
# routes #
###########
# Настройка маршрутов
# Обязательная секция
route_1: &route_1
  collector_id:
    - "eventlog_collector"
    - "tcp_input"
  sender_id:
    - "tcp_output"

route_2: &route_2
  collector_id:
    - "udp_input"
  sender_id:
    - "tcp_output"
    - "udp_output"
    - "kafka_output"

############
```

```
# routers #
############
# Включение маршрутов
# Обязательная секция
routers:
  - <<: *route_1
  - <<: *route_2
```