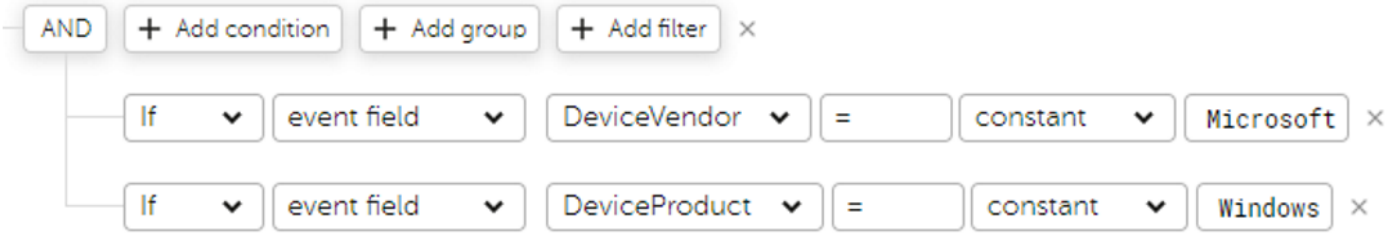
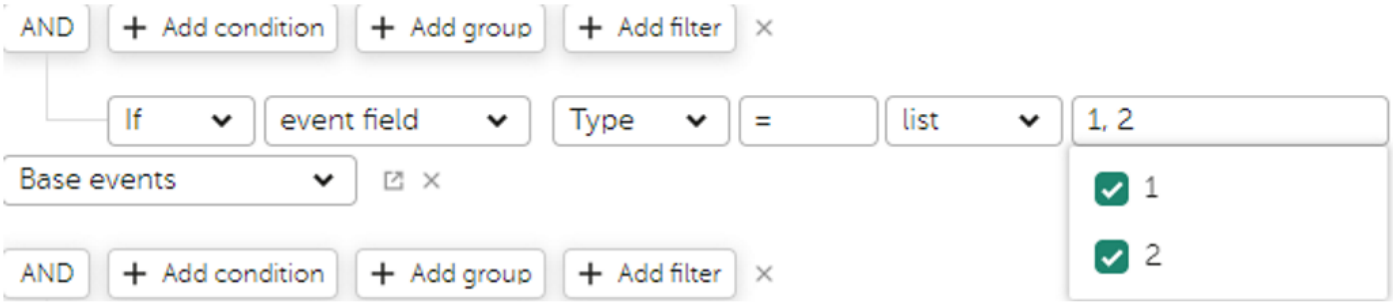
# Приемы в правилах корреляции

## Сравнение с константой



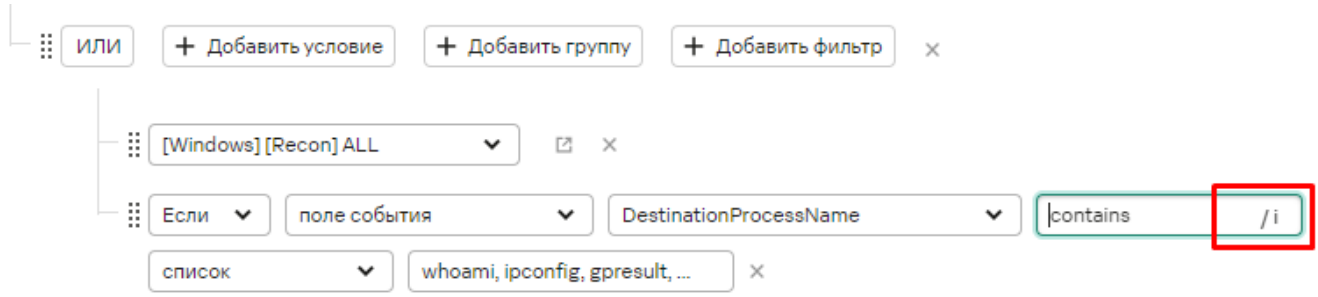

## Сравнение с листом/списком

Аналогично =константе **ИЛИ** =константе




## Содержит список констант регистронезависимый

Ищется заданная подстрока *“whoami” или "ipconfig" и др* в занчении поля *DestinationProcessName*




# Соответствие регулярному выражению (REGEX)

Должно быть условие регулярного выражения в формате RE2


# Работа с подсетями


# Содержит любое значение (не пустое)

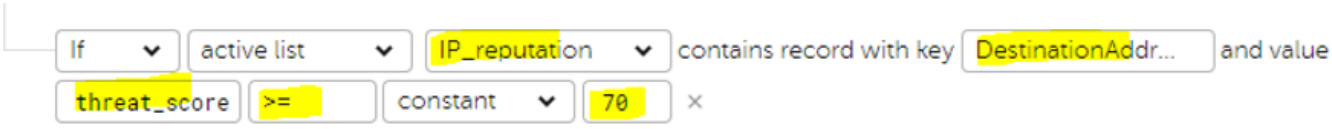
# Активный список / лист

## Активный лист содержит ключ

Ключ листа должен совпадать со значением поля *Message*

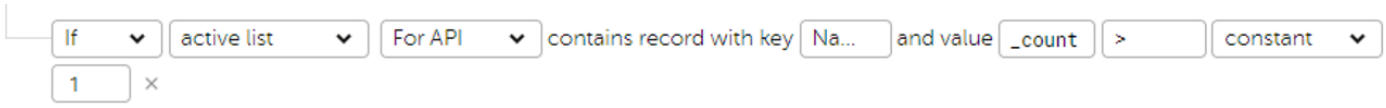

## Сравнение по экстра данным в активном листе (неключевому полю)

Где *threat_score* > 70




## Фильтр по количеству записей по ключу в активном листе

Где количество записей > 1, используется служебная переменная *_count*

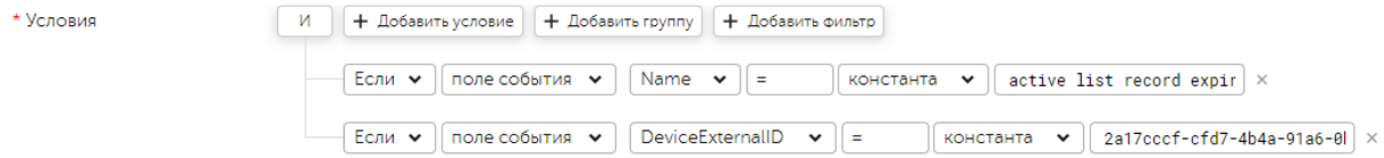



Другие служебные поля активных листов:

- _count (счетчик количества записей)
- _created (время создания записи UnixTime, в наносекундах)
- _updated (время обновления записи UnixTime, в наносекундах)
- _expires (время окончания жизни записи UnixTime, в наносекундах)
- _key (значение ключевой записи)

## Событие истечения времени жизни в активном листе

Возникает служебное событие *active list record expired*, помимо этого необходимо указать UUID активного листа в поле *DeviceExternalID*




Значение ключевого поля передатся в *DevicePayloadID* служебного события.

## Работа с активным листом не по его ID

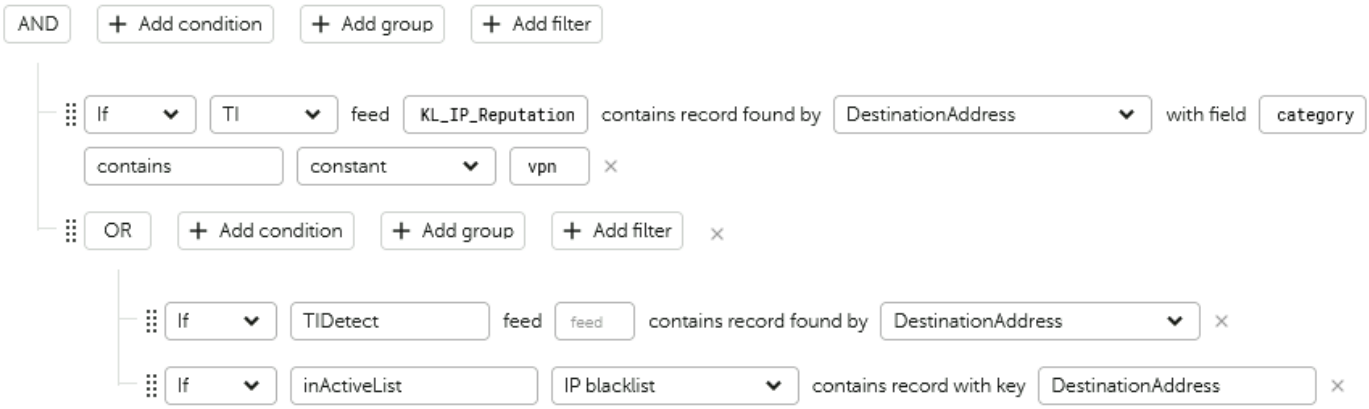
Если указывать ID листа неудобно (а это обычно так), то можно в ключевое поле писать уникальный префикс типа "failed login attempts|username|1.1.1.1" и в события ловить по полю devicePayloadID функцией startsWith "failed login attempts" такой вариант реализации правила не зависит от инсталляции.

> Такие события существуют только в рамках коррелятора (служебные события) и не сохраняются в сторадже, их можно поймать только правилом корреляции.

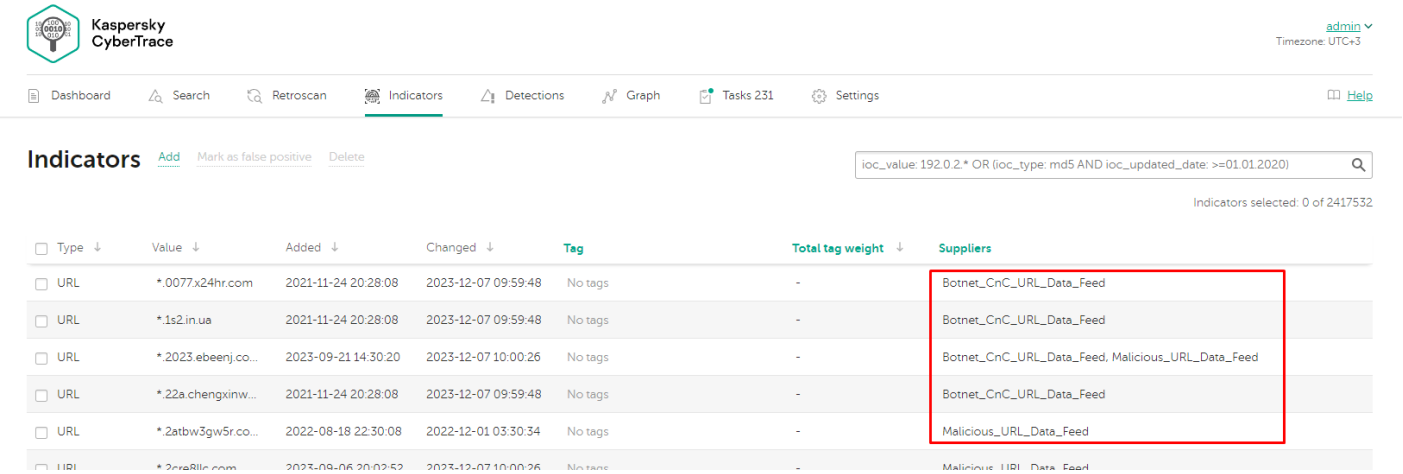
# Сравнение переменной с числовым значением

Из строки с REGEX вырезается число и кладется его в переменную, например, для получения SID пользователя из "SID: 1-21-1231-500", получаем "500", кладем в переменную $temp, чтобы сделать сравнение необходимо $temp привести к числовому типу это можно сделать новой переменной `$usersid = $temp + 0` и далее сравнивать, например, `$usersid > 1000`

# Условие с полем TI, сравнение с категорией

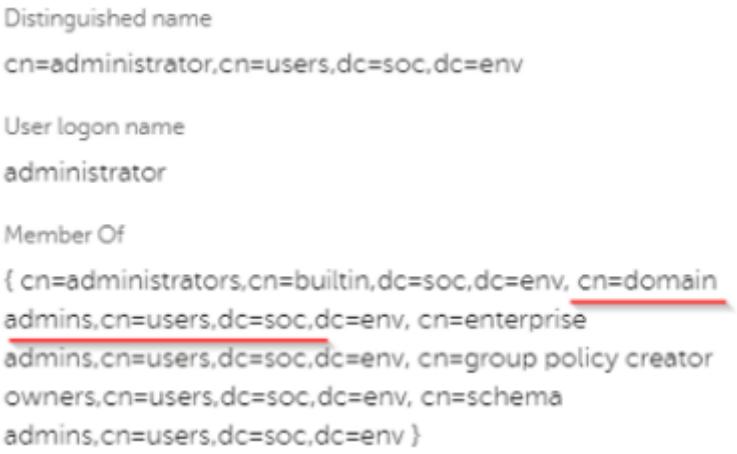



Значение feed это именование фидов из CyberTrace, например для фидов Kaspersky бывают такие значения (зависит какие фиды приобретены/подключены):

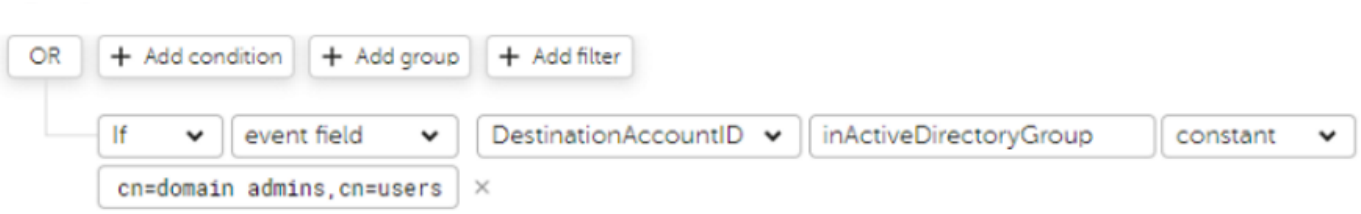



# Работа с группами AD

Необходимо указывать полный DN, пример:

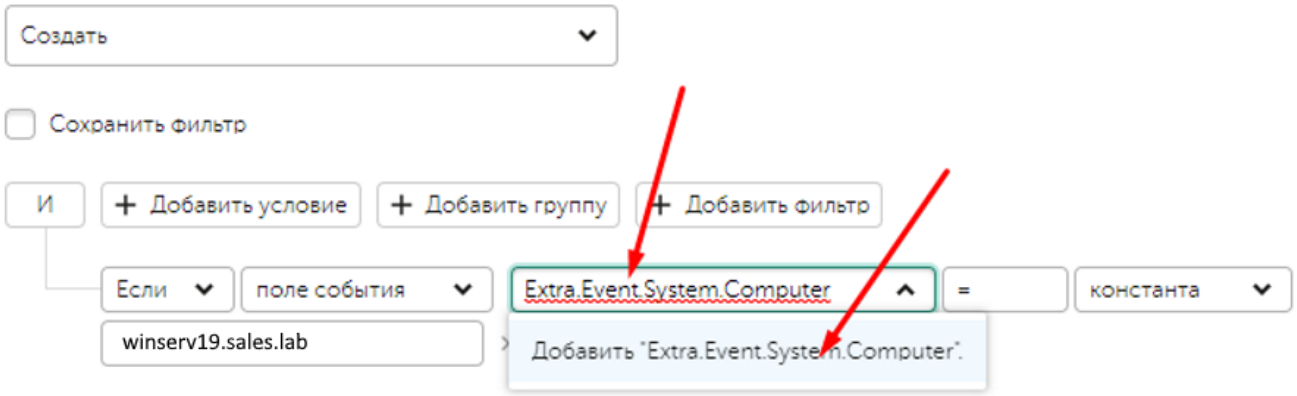



Условие фильтра:




# Актив находится в определенной категории


# Работа с полем Extra в селекторе



# Работа с полями типа SA (массив строк) KUMA 3.0+

При операции **match** к полям типа SA применяется как к строке, т.е. массив представляется в виде строки **["a1", "b2", "c3"]** и т.д. Т.е. ко всему массиву сразу, а не к отдельным его элементам по очереди.

При операции **contains** применяется именно к элементам массива. Т.е. **contains [** для массива вернет false, как и **contains "** или **'**. Но при этом же, если в массиве есть элемент abc, то contains abc вернет true и contains ab тоже вернет true

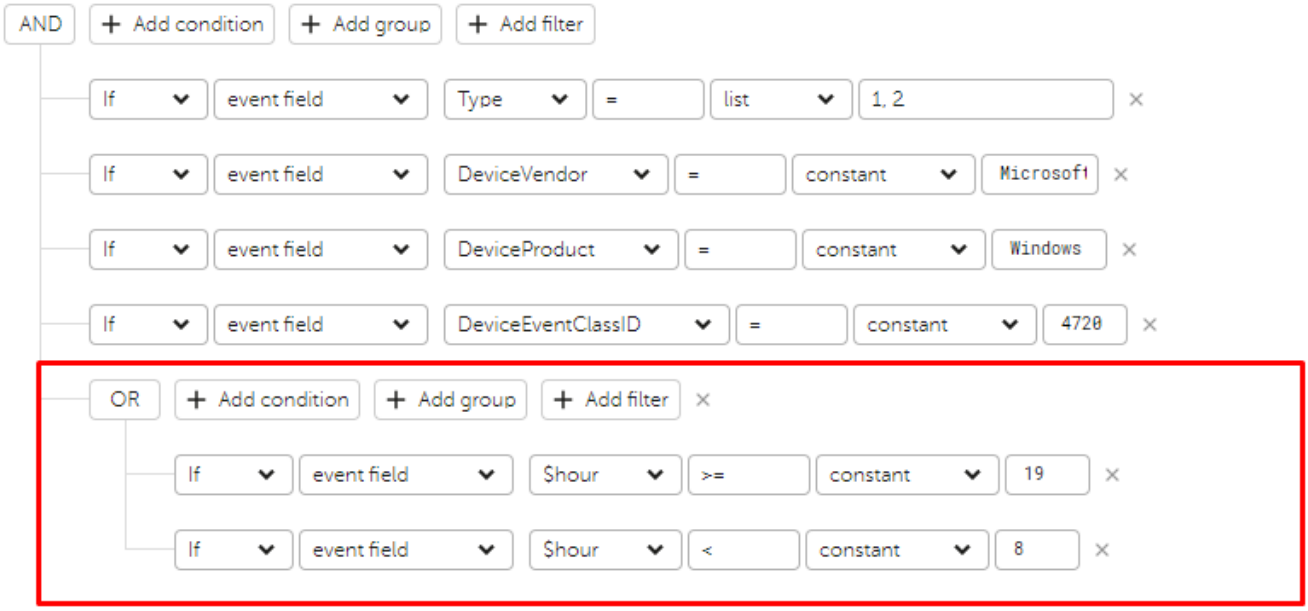
# Работа с переменными (KUMA 2.1+)

## Вход в рабочее время

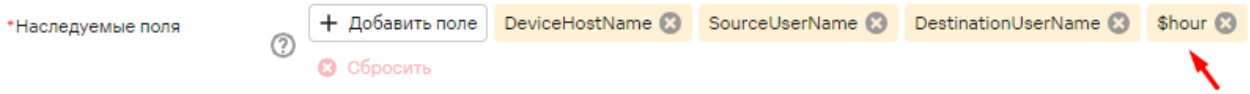
**Документация** по функциям переменных (локальные переменные).

Сначала извлекается час из таймштемпа, с помощью функции: *extract_from_timestamp(Timestamp, 'h', 'Europe/Moscow')*


Условие в селекторе:

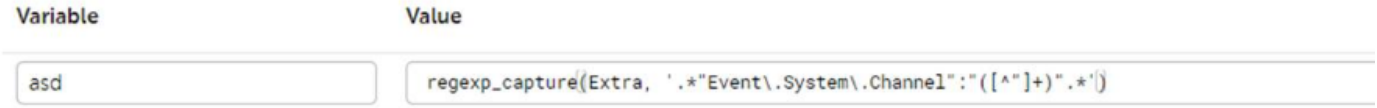



Переменные необходимо указывать в группирующих полях:




# Работа с Extra

Здесь вместо *Event.System.Channel* нужно указать интересующее вас поле экстра. Регулярка: `."Event\.System\.Channel":"([^"]+)".`



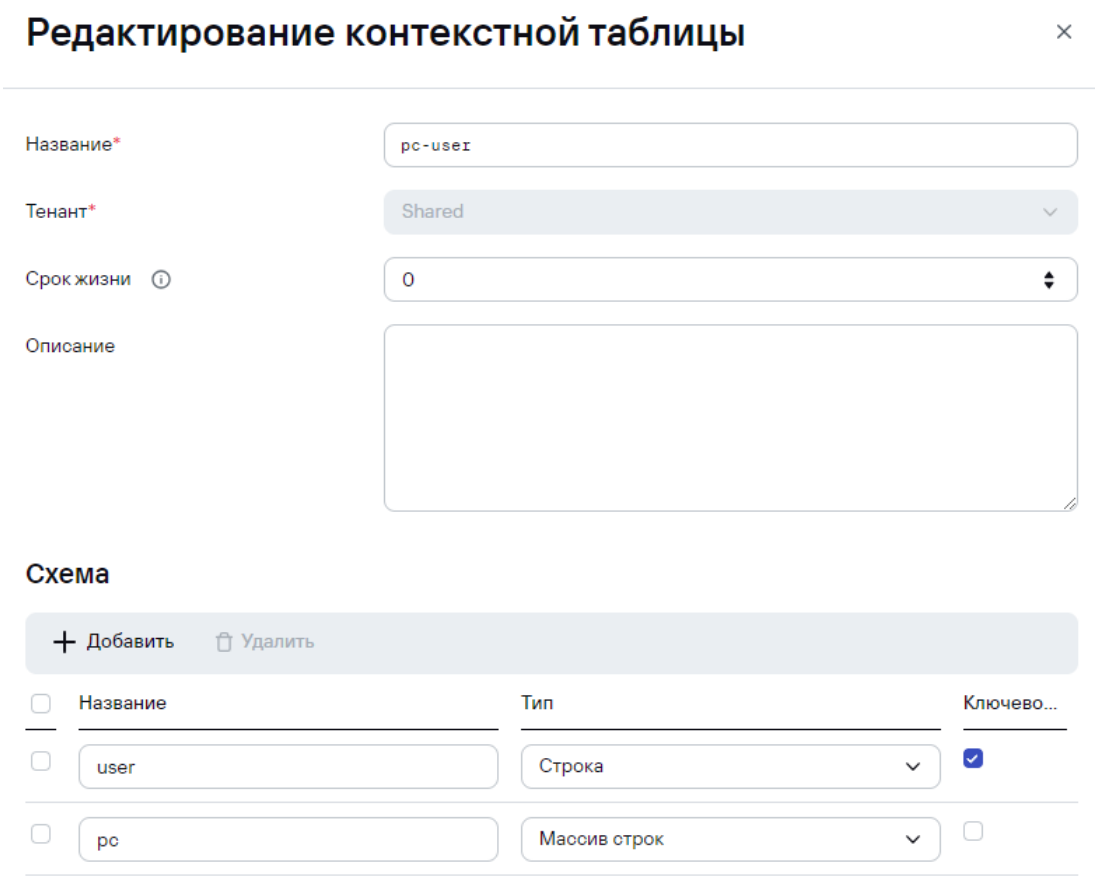

В некоторых случаях производительней использовать не регулярки, а функции:

- $firstIndex=index_of('(', fieldName)
- $lastIndex=index_of('.', fieldName)
- $getSubstring=substring(fieldName, $firstIndex, $lastIndex)

# Работа с контекстной таблицей (KUMA 3.0+)

Исходная задача: Необходимо отслеживать, на каких отличных друг от друга устройствах производится вход одной УЗ. (активный лист будет менее удобен т.к. различных устройств может быть > 1)

Пример Контестной таблицы:




Примеры переменных:

Извлечение содержимого поля с массивом `pc` из контекстной таблицы в тенанте `Shared` (только для этого тенанта нужно в переменной это указывать) по ключевому полю `user` и его значением `DestinationUserName`, назовем переменную `ct_value`:

```
context_table('pc-user@Shared', 'pc', 'user', DestinationUserName)
```

В событии выглядит это так:

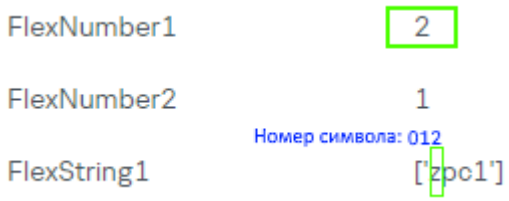
| | |
|---|---|
| FlexString1 | ['zpc1'] |

Получение индекса (номер символа) по содержимому поля массива `pc` по значению поля `DestinationHostName` из контекстной таблицы в тенанте Shared (только для этого тенанта нужно в переменной это указывать) по ключевому полю user и его значением DestinationUserName, назовем переменную `ct_contains`:

```
index_of(DestinationHostName, $ct_value)
```

Возвращает первую позицию символа или подстроки в строке, расчет индекса начинается с 0. Если в результате работы функции подстрока не была найдена, функция вернёт значение -9223372036854775808

Вот как выглядит это в событии, значение `ct_contains` в поле FlexNumber1.




Получение количества элементов в поле с массивом `pc` из контекстной таблицы (пример в собтии на рисунке выыше в поле FlexNumber2), назовем переменную `ct_len`:

```
len($ct_value)
```

Если в содержимом поля с массивом `pc` из контекстной таблицы есть подстока см. выше описание переменной `ct_contains`, то вернуть `true` назовем переменную `ct_item_exist`:

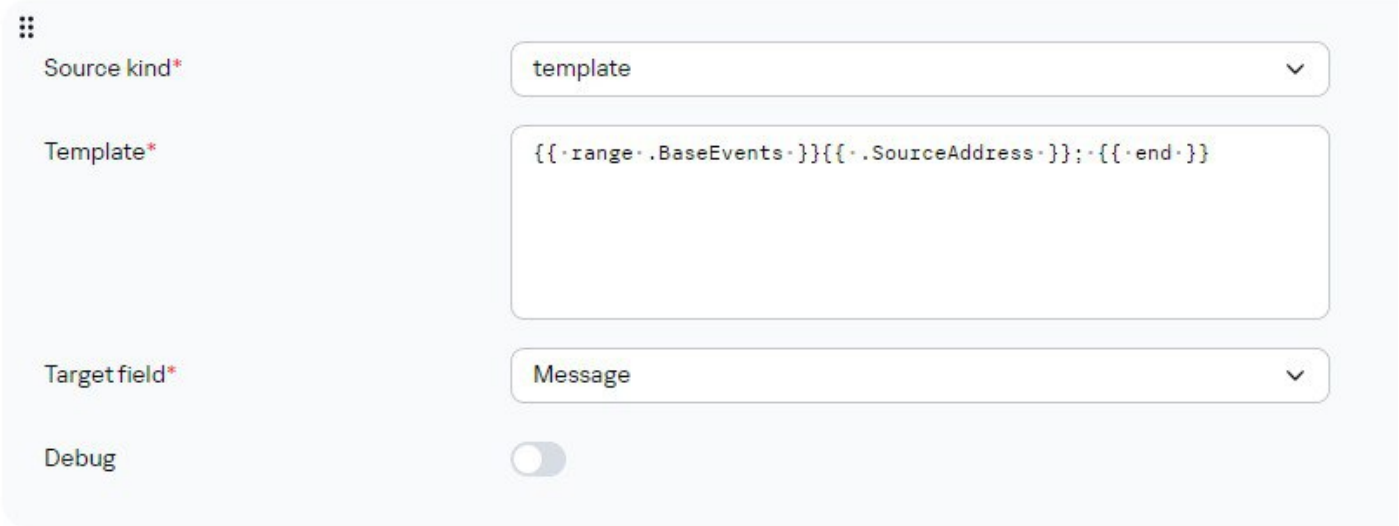
```
conditional(`$ct_contains LIKE '-.*'`, 'false', 'true')
```

Вот как выглядит это в событии, значение `ct_item_exist` в поле FlexString2.


# Как записать значение из нескольких событий в одно поле в правиле корреляции типа Standard

В обогащении можнонаписать шаблон, в котором можно пройтись по всем подсобытиям и получить список всех полей. Дальше уже с помощью функционала Go template можете сделать что хотите. Для правильной работы метода keep event policy в правиле корреляции должна иметь значение all.

В данном примере можно получить в строку все значения SourceAddress из базовых событий через ";" в корреляционном событии.




На выходе получается примерно следующее


Revision #20
Created 9 August 2023 13:06:47 by Boris Rzr
Updated 12 May 2025 08:41:54 by Boris Rzr