

# Стандартное правило (standard)

**“Обновить параметры”** нужно делать в корреляторе, когда какое-либо правило меняется, чтобы подтянулись актуальные изменения в правилах в коррелятор.

<https://www.youtube.com/embed/TauttDGugBc?si=jw05NxFfgxOuyioY>

**Стандартное правило (standard)** — срабатывает при достижении определенного порогового значения группы событий, которые удовлетворяют условиям селектора, полей группировки событий (на основе значений поля создается группа) и времени жизни контейнера для группы.

Если частота срабатывания (Rate limiting) явна не указана, то устанавливается лимит умолчанию - 100 срабатываний в секунду. При превышении лимита правило ничего не делает.

Политика хранения базовых событий (Base events keep policy) - указание, какие из базовых событий должны сохраняться в корреляционном. Возможно указать одно из значений:

- first (по умолчанию) - сохранять только первое базовое событие от каждого селектора в корреляционном событии
- last - сохранять только последнее базовое событие от каждого селектора в корреляционном событии
- all - сохранять все базовые события в корреляционном событии

Типовой пример правила:

Общие

Селекторы

Действия

Название

[general] Обнаружено сканирование портов [B]

Тенант

test2

Тип

standard

Группирующие поля

+ Добавить поле

SourceAddress

DestinationAddress

Очистить

Уникальные поля

+ Добавить поле

DestinationPort

Очистить

Частота срабатываний

1000

Время жизни контейнера, сек.

60

Политика хранения базовых событий

all

Уровень важности

Средний

Сортировать по

Описание

Правило обнаруживает перебор различных 30 портов по одному хосту в течение 60 сек.

Общие

Селекторы

Действия

Селектор "More than 30 ports"

Параметры

Локальные переменные

Название

More than 30 ports

Порог срабатывания селектора (количество событий)

30

Фильтр

Создать

Сохранить фильтр

Условия

И

+ Добавить условие

+ Добавить группу

+ Добавить фильтр

Если

поле события

Device/Vendor

=

/1

список

×

Если

поле события

DestinationPort

<=

константа

1824

×

Если не

поле события

Type

=

константа

3

×

Если не

inActiveList

[Exclusion] General port Scan (Vertical)

SourceAddress

×

Если не

inActiveList

[Detection] General port Scan (Vertical)

SourceAddress

×

Обнуление

Общие

Селекторы

Действия

Действия

> На первом срабатывании правила

> На последующих срабатываниях правила

> На каждом срабатывании правила

> По истечении времени жизни контейнера

Возможные действия (допускается указать одно или более) правила:

- On first threshold — создавать корреляционное событие только после первого превышения порога, а двукратное, трехкратное и т.д. превышение порога за время жизни группы игнорировать.
- On every threshold — создавать корреляционное событие после каждого превышения порога за время жизни группы.
- On subsequent threshold — создавать корреляционные событие при всех превышениях порога, кроме первого.
- On timeout — в стандартных правилах есть еще возможность настройки действий по окончании времени жизни группы. Это действие используется в связке с опцией Recovery (Обнуление) в настройках селектора, в каких случаях это уместно и как именно это работает рассматривается ниже.

Необходимо указывать все поля и переменные участвующие в селекторах в группирующих/уникальных полях.

Можно также использовать **несколько селекторов**. Например, несколько неудачных попыток брутфорса (ловится на основе сработки другого правила корреляции) и успешный вход.

Пример правила с несколькими селекторами:

Общие

Селекторы

Действия

Название

[Linux] обнаружен успешный брутфорс

Тенант

test2

Тип

standard

Группирующие поля

+ Добавить поле DeviceHostName DestinationUserName Сбросить

Уникальные поля

+ Добавить поле

Частота срабатываний

0

Время жизни контейнера, сек.

900

Политика хранения базовых событий

all

Уровень важности

Высокий

Сортировать по

Описание

Правило обнаруживает события успешного входа в систему после большого количества неудачных попыток авторизации от 10.

Селектор "BruteForce"

Параметры

Локальные переменные

Название

BruteForce

Порог срабатывания селектора (количество событий)

1

Фильтр

Создать

Сохранить фильтр

Условия

И + Добавить условие + Добавить группу + Добавить фильтр

Если поле события Name = константа

обнаружена попытка брутфорс

Если поле события Type = константа 3

Обучение

Удалить селектор

Селектор "Login\_Success"

Параметры

Локальные переменные

Название

Login\_Success

Порог срабатывания селектора (количество событий)

1

Фильтр

Создать

Сохранить фильтр

Условия

И + Добавить условие + Добавить группу + Добавить фильтр

(Filter) Linux AuditD base events

Если поле события DeviceEventClassID = список

USER\_LOGIN\_USER\_AUTH

Если не поле события DestinationUserName = список

splunk, zabbix, (unknown)

Если поле события EventOutcome = константа success

Обучение

Удалить селектор

Добавить селектор

Действия

На первом срабатывании правила

На последующих срабатываниях правила

На каждом срабатывании правила

Отправить событие на дальнейшую обработку

Отправить событие снова в коррелятор

Не создавать алерт

Обогащение

Добавить обогащение

Обновление активных листов

Добавить действие с активным листом

Изменение категорий

Добавить категоризацию

По истечении времени жизни контейнера

Можно также **рекавери правило**. Например, когда событие типа «Вредоносное ПО удалено» не обнаружено в течение 5 минут после получения события «Вредоносное ПО обнаружено».

Общие

Селекторы

Действия

Название

[KSC] обнаружено ВПО, невозможно удалить

Тенант

test2

Тип

standard

Группирующие поля

+ Добавить поле DeviceCustomString1 DeviceCustomStringLabel DeviceProduct DeviceVendor Filename Filepath DestinationHostName Сбросить

Уникальные поля

+ Добавить поле

Частота срабатываний

0

Время жизни контейнера, сек.

300

Политика хранения базовых событий

all

Уровень важности

Высокий

Сортировать по

Описание

Неудачная попытка лечения или удаления вредоносного объекта. Срабатывает когда событие типа «Вредоносное ПО удалено» не обнаружено в течение 5 минут после получения события «Вредоносное ПО обнаружено».

Селектор "KSC Virus Found"

Параметры

Локальные переменные

Название

KSC Virus Found

Порог срабатывания селектора (количество событий)

1

Фильтр

(Filter) KSC Virus Found

Сохранить фильтр

Условия

И + Добавить условие + Добавить группу + Добавить фильтр

(Filter) KSC Base Events

Если не поле события DeviceCustomString1 startsWith константа

not-a-virus

Если поле события DeviceEventClassID startsWith константа

WML\_FX\_VIRUS\_FOUND

Обучение

Удалить селектор

Селектор "KSC Virus Deleted"

Параметры

Локальные переменные

Название

KSC Virus Deleted

Порог срабатывания селектора (количество событий)

1

Фильтр

(Filter) KSC Object Deleted

Сохранить фильтр

Условия

И + Добавить условие + Добавить группу + Добавить фильтр

Если поле события DeviceEventClassID = константа

WML\_FX\_OBJECT\_DELETED

(Filter) KSC Base Events

Обучение

Удалить селектор

Добавить селектор

Действия

На первом срабатывании правила

На последующих срабатываниях правила

На каждом срабатывании правила

По истечении времени жизни контейнера

Отправить событие на дальнейшую обработку

Отправить событие снова в коррелятор

Не создавать алерт

Обогащение

Обогащение №1

Тип источника данных

шаблон

Шаблон

ВПО было обнаружено и не было удалено на хосте ({{DestinationHostName}}).

Целевое поле

Message

Отладка

Включено

Удалить обогащение

Добавить обогащение

# Бакет (Окно корреляции)

1. Бакет открывается на событие из любого селектора, не важно в каком они порядке в правиле, порядок проверяется после наполнения бакета!
2. Для каждого набора Identical Fields создается свой бакет.
3. Когда событие подпадает под селектор, коррелятор смотрит, есть ли уже бакет с нужным набором полей Identical Fields, если нет - создает, если есть - событие отправляется в существующий.
4. Когда под селектор с Unique fields подпадает событие, то проверяется, есть ли уже в бакете события с таким же набором значений для Unique Fields, если есть, то событие не учитывается.

## Recovery селектор (Обнуление)

1. Бакет открывается только на событие из обычного селектора, на событие из recovery-селектора бакет не открывается никогда!
2. Место нахождения селектора с recovery не имеет значения, как только в бакет попадут все нужные recovery-события бакет будет закрыт!
3. На recovery-селектор не влияет настройка фильтра Order By.
4. Если нужно, чтобы произошло событие А, и не произошло событие Б, при этом событие Б может произойти раньше А, нужно использовать активные листы, т.к. с помощью recovery-селектора такой логики не достичь (см п.1).

---

Revision #7

Created 9 August 2023 12:44:28 by Boris RZR

Updated 7 July 2024 08:21:47 by Koala