Плейбуки

- Триггеры в плейбуках
- Действия в плейбуках

Триггеры в плейбуках

> Конструктор jq для OSMP: https://koalapower.github.io/jq_builder/

> Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

> Официальная документация по данному разделу приведена в Онлайн-справке на продукт: https://support.kaspersky.com/help/XDR/1.4/ru-RU/273327.htm

В SMP в плейбуках для описания триггеров, правил сегментации и агрегации, а также для фильтров в действиях плейбуков используется язык jq. В этой статье приведены выражения на языке jq, которые будут полезны при создании триггеров плейбуков.

> Для тренировки в написании выражений на jq можно воспользоваться специальными платформами, например такой: https://www.devtoolsdaily.com/jq_playground/. Также, по опыту, LLM помощники хороши справляются с этой задачей.

Т.к. описание триггера плейбуков отличается для Алертов и Инцидентов данная статья разделена на две соответствующие части.

Алерты

Свойства алерта

Ниже приведены примеры триггеров для алертов, по их базовым свойствам

1. Имя алерта начинается с `Test`

```
.Name | startswith("Test")
```

2. Имя алерта `Test`

```
.Name == "Test"
```

3. Тактика `TA0003`

```
[.MITRETactics[] | .ID == " TA0003"] | any
```

4. Техника `T1098`

```
[.MITRETechniques[] | .ID == " T1098"] | any
```

5. Уровень важности алерта высокий (`high`). Возможные значения: `low`, `medium`, `high`, `critical`.

```
.Severity =="high"
```

6. Статус алерта - новый (`new`). Возможные значения: `new`, `inProgress`, `inIncident`, `closed`.

```
.Status == "new"
```

Правила корреляции

Ниже приведены примеры триггеров для алертов, по свойствам правил корреляции

1. Правило начинается с `Test`

```
[.Rules[] | .Name | startswith("Test")] | any
```

2. Название правила `Test`

```
[.Rules[] | .Name=="Test"] | any
```

3. Название правила содержит `Test`

```
[.Rules[] | .Name | contains("Test")] | any
```

3. Правило соответствует регулярному выражению `^R\d{3}.+` (характерно для коробочного контента)

```
[.Rules[]| .Name | test("R\\d{3}.+") ] | any
```

4. Уровень важности правила - критический (`critical`). Возможные значения: `low`, `medium`, `high`, `critical`.

```
[.Rules[] | .Severity == "critical"] | any
```

5. Алерт относится к КИИ (содержит КИИ активы)

```
.IsCII
```

Активы

В разделе приведены примеры триггеров для активов, связанных с алертами. Т.к. в SMP активы объединяют в себе устройства и пользователей, раздел разделен на две соответствующие части.

Устройства

1. Имя устройства `TEST-PC`

```
[.Assets[] | select(.Type == "host") | .Name=="TEST-PC"] | any
```

2. ID устройства `00000000-0000-0000-0000-000000000000`

```
[.Assets[] | select(.Type == "host") | .ID=="00000000-0000-0000-0000-000000000000"] | any
```

3. По категории критичности актива

3.1 Актив относится к КИИ

```
[.Assets[] | select(.Type == "host") | .CIICategory == "CII"] | any
```

3.2 Актив не относится к КИИ

```
[.Assets[] | select(.Type == "host") | .CIICategory == "notCII"] | any
```

Пользователи

1. Имя учетной записи `test`

```
[.Assets[] | select(.Type == "user") | .Name=="test"] | any
```

2. ID учетной записи (внутренний ID в платформе) `00000000-0000-0000-0000-000000000000`

```
[.Assets[] | select(.Type == "user") | .ID=="00000000-0000-0000-0000-000000000000"] | any
```

Наблюдаемые объекты

В разделе приведены примеры триггеров для наблюдаемых объектов, связанных с алертами.

> Наблюдаемые объекты - это индикаторы, в т.ч. хэши, ip-адреса, url, домены, имена пользователей и т.п., которые автоматически извлекаются из базовых и корреляционных событий по внутренней логике SMP и добавляются в карточку алерта/ицидента в соответствующий раздел.

1. Домен `DEMO.LAB`

```
[.Observables[] | select(.Type == "domain") | .Value=="DEMO.LAB"] | any
```

2. URL `https://abc.demo.lab`

```
[.Observables[] | select(.Type == "url") | .Value=="https://abc.demo.lab"] | any
```

3. Хэш

3.1 MD5 хэш `00000000000000000000000000000000`

```
[.Observables[] | select(.Type == "md5") | .Value=="00000000000000000000000000000000"] | any
```

3.2 SHA256 хэш `0000000000000000000000000000000000000000000000000000000000000000`

```
[.Observables[] | select(.Type == "sha256") |
.Value=="0000000000000000000000000000000000000000000000000000000000000000"] | any
```

4. IP-адрес

4.1 IP-адрес равен `192.168.100.100`

```
[.Observables[] | select(.Type == "ip") | .Value=="192.168.100.100"] | any
```

4.2 IP-адрес находится в подсети `192.168.100.0/24` (грубый пример)

```
[.Observables[] | select(.Type == "ip") | .Value | (test("^192\\.168\\.100\\.")] | any
```

> В jq нет встроенной функции проверки адреса в подсети, поэтому предлагается использовать обходные пути через регулярные выражения или contains.

5. Имя пользователя `Alice`

```
[.Observables[] | select(.Type == "userName") | .Value=="Alice"] | any
```

6. Устройство

6.1 Имя устройства `abc.demo.lab`

```
[.Observables[] | select(.Type == "hostName") | .Value=="abc.demo.lab"] | any
```

6.1 Домен устройства `demo.lab` (грубый пример)

```
[.Observables[] | select(.Type == "hostName") | .Value | endswith(".demo.lab")] | any
```

События

Помимо прочего, в триггерах можно обращаться к базовым и корреляционным событиям и извлекать для проверки их них любую информацию. В виду большого числа вариантов ниже будет приведено лишь несколько примеров такого использования.

1. По SID пользователя из базового события в поле `SourceUserID` равный `S-1-5-18`

```
[.OriginalEvents[] | .BaseEvents[] | .SourceUserID == "S-1-5-18"] | any
```

2. По домену из базового события в поле `SourceNtDomain` равный `DEMO.LAB`

```
[.OriginalEvents[] | .BaseEvents[] .SourceNtDomain=="DEMO.LAB" ] | any
```

Инциденты

Свойства инцидента

Ниже приведены примеры триггеров для инцидентов, по их базовым свойствам

1. Имя инцидента начинается с `Test`

```
  .Name | startswith("Test")
```

2. Имя инцидента `Test`

```
  .Name == "Test"
```

3. Тактика `TA0003`

```
[ .Alerts[] | .MITRETactics[] | .ID == " TA0003"] | any
```

4. Техника `T1098`

```
[ .Alerts[] | .MITRETechniques[] | .ID == " T1098" ] | any
```

5. Уровень важности инцидента высокий (`high`). Возможные значения: `low`, `medium`, `high`, `critical`.

```
.Severity =="high"
```

6. Уровень приоритета инцидента высокий (`high`). Возможные значения: `low`, `medium`, `high`, `critical`.

```
.Priority =="high"
```

7. Статус инциденнта - открыт (`open`). Возможные значения: `open`, `inProgress`, `hold`, `closed`.

```
.Status == "open"
```

8. Имя рабочего процесса инцидента `standard`.

```
.WorkflowName == "standard"
```

9. Инцидент относится к КИИ (содержит КИИ активы)

```
.IsCII
```

Правила корреляции

Ниже приведены примеры триггеров для инцидентов, по свойствам правил корреляции

1. Правило начинается с `Test`

```
[ .Alerts[] | .Rules[] | .Name | startswith("Test")] | any
```

2. Название правила `Test`

```
[ .Alerts[] | .Rules[] | .Name=="Test"] | any
```

3. Название правила содержит `Test`

```
[ .Alerts[] | .Rules[] | .Name | contains("Test")] | any
```

3. Правило соответствует регулярному выражению `^R\d{3}.+` (характерно для коробочного контента)

```
[ .Alerts[] | .Rules[]| .Name | test("R\\d{3}.+") ] | any
```

4. Уровень важности правила - критический (`critical`). Возможные значения: `low`, `medium`, `high`, `critical`.

```
[ .Alerts[] | .Rules[] | .Severity == "critical"] | any
```

Активы

В разделе приведены примеры триггеров для активов, связанных с инцидентами. Т.к. в SMP активы объединяют в себе устройства и пользователей, раздел разделен на две соответствующие части.

Устройства

1. Имя устройства `TEST-PC`

```
[ .Alerts[] | .Assets[] | select(.Type == "host") | .Name=="TEST-PC"] | any
```

2. ID устройства `00000000-0000-0000-0000-000000000000`

```
[ .Alerts[] | .Assets[] | select(.Type == "host") | .ID=="00000000-0000-0000-0000-
000000000000"] | any
```

3. По категории критичности актива

3.1 Актив относится к КИИ

```
[ .Alerts[] | .Assets[] | select(.Type == "host") | .CIICategory == "CII"] | any
```

3.2 Актив не относится к КИИ

```
[ .Alerts[] | .Assets[] | select(.Type == "host") | .CIICategory == "notCII"] | any
```

Пользователи

1. Имя учетной записи `test`

```
[ .Alerts[] | .Assets[] | select(.Type == "user") | .Name=="test"] | any
```

2. ID учетной записи (внутренний ID в платформе) `00000000-0000-0000-0000-000000000000`

```
[ .Alerts[] | .Assets[] | select(.Type == "user") | .ID=="00000000-0000-0000-0000-
000000000000"] | any
```

Наблюдаемые объекты

В разделе приведены примеры триггеров для наблюдаемых объектов, связанных с инцидентами.

> Наблюдаемые объекты - это индикаторы, в т.ч. хэши, ip-адреса, url, домены, имена пользователей и т.п., которые автоматически извлекаются из базовых и корреляционных событий по внутренней логике SMP и добавляются в карточку алерта/ицидента в соответствующий раздел.

1. Домен `DEMO.LAB`

```
[ .Alerts[] | .Observables[] | select(.Type == "domain") | .Value=="DEMO.LAB"] | any
```

2. URL `https://abc.demo.lab`

```
[ .Alerts[] | .Observables[] | select(.Type == "url") | .Value=="https://abc.demo.lab"] | any
```

3. Хэш

3.1 MD5 хэш `00000000000000000000000000000000`

```
[ .Alerts[] | .Observables[] | select(.Type == "md5") |
.Value=="00000000000000000000000000000000"] | any
```

3.2 SHA256 хэш `0000000000000000000000000000000000000000000000000000000000000000`

```
[ .Alerts[] | .Observables[] | select(.Type == "sha256") |
.Value=="0000000000000000000000000000000000000000000000000000000000000000"] | any
```

4. IP-адрес

4.1 IP-адрес равен `192.168.100.100`

```
[ .Alerts[] | .Observables[] | select(.Type == "ip") | .Value=="192.168.100.100"] | any
```

4.2 IP-адрес находится в подсети `192.168.100.0/24` (грубый пример)

```
[ .Alerts[] | .Observables[] | select(.Type == "ip") | .Value | test("^192\\.168\\.100\\.")] | any
```

> В jq нет встроенной функции проверки адреса в подсети, поэтому предлагается использовать обходные пути через регулярные выражения или contains.

5. Имя пользователя `Alice`

```
[ .Alerts[] | .Observables[] | select(.Type == "userName") | .Value=="Alice"] | any
```

6. Устройство

6.1 Имя устройства `abc.demo.lab`

```
[ .Alerts[] | .Observables[] | select(.Type == "hostName") | .Value=="abc.demo.lab"] | any
```

6.1 Домен устройства `demo.lab` (грубый пример)

```
[ .Alerts[] | .Observables[] | select(.Type == "hostName") | .Value | endswith(".demo.lab")] | any
```

События

Помимо прочего, в триггерах можно обращаться к базовым и корреляционным событиям и извлекать для проверки их них любую информацию. В виду большого числа вариантов ниже будет приведено лишь несколько примеров такого использования.

1. По SID пользователя из базового события в поле `SourceUserID` равный `S-1-5-18`

```
[ .Alerts[] | .OriginalEvents[] | .BaseEvents[] | .SourceUserID == "S-1-5-18"] | any
```

2. По домену из базового события в поле `SourceNtDomain` равный `DEMO.LAB`

```
[ .Alerts[] | .OriginalEvents[] | .BaseEvents[] .SourceNtDomain=="DEMO.LAB" ] | any
```

Объединение условий

Приведенные выше условия можно объединять между собой используя различные логические операторы, такие как or, and, not.

Например, выражение для "Имя инцидента начинается с `Test`, статус инцидента - открыт (`open`) и инцидент не относится к КИИ

```
( .Name | startswith("Test")) and (.Status == "open") and (.IsCII | not)
```

В примерах с массивами выше часто встречается условие с `any`, которое означает "совпадение как минимум одного объекта", если для условия требуется совпадение всех объектов, то условие следует заменить на `all`.

Например, для инцидента, у которого все связанные правила корреляции имеют важность критические

```
[ .Alerts[] | .Rules[] | .Severity == "critical"] | all
```

Также для сравнения не с одним, но с массивом значений можно использовать оператор `IN`.

Например, триггер по нескольким именам хостов

```
[ .Alerts[] | .Assets[] | select(.Type == "host") | .Name | IN("TEST-PC1", "TEST-PC2", "ADMIN-PC")] | any
```

Заключение

Данная статья содержит базовый набор выражений, которые пригодятся при описании триггеров плейбуков. Если у вас есть замечания или предложения по содержимому статьи, а также собственные идеи для выражений - пишите в наш телеграм-канал (ссылка и инструкция на главной странице).

Полезные ссылки

Модель данных алерта - https://support.kaspersky.com/help/XDR/1.4/ru-RU/269125.htm

Модель данных инцидента - https://support.kaspersky.com/help/XDR/1.4/ru-RU/269168.htm

Действия в плейбуках

Конструктор jq для OSMP: https://koalapower.github.io/jq_builder/

Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

Официальная документация по данному разделу приведена в Онлайн-справке на продукт: https://support.kaspersky.com/help/XDR/1.4/ru-RU/270357.htm

В SMP в плейбуках для описания триггеров, правил сегментации и агрегации, а также для фильтров в действиях плейбуков используется язык jq. В этой статье приведены выражения на языке jq, которые будут полезны при создании действий плейбуков.

Для тренировки в написании выражений на jq можно воспользоваться специальными платформами, например такой: https://www.devtoolsdaily.com/jq_playground/. Также, по опыту, LLM помощники хороши справляются с этой задачей.

В зависимости от выбранного типа действия в плейбуке потребуется задать выражение, которое извлечет из алерта/инцидента активы (пользователи и устройства), хэш, PID или путь к файлу, IP-адрес и наблюдаемые объекты любых типов. По этой причине вместо создания готовых плейбуков было принято решение опубликовать примеры таких выражений, которые могут быть использованы сразу в большом количестве плейбуков и покрывать множество кейсов.

Также от объекта над которым выполняются действия - Алерт, Инцидент, Выбранные объекты - зависит jq-запрос. Поэтому запросы ниже разделены на три соответствующие группы.

Алерты

В разделе приведены примеры фильтров для раличных типов действий, выполняемых над Алертом.

Активы

Ниже приведены примеры jq-выражений для извлечения активов. Выражение в общем смысле должно возвращать массив ID активов.

Устройства

1. Имя устройства `TEST`

```
[ alert.Assets[] | select(.Name == "TEST") | .ID]
```

2. Тип актива - устройство (host)

```
[ alert.Assets[] | select(.Type == "host") | .ID]
```

3. Тип актива - устройство (host) и имеет признаки атакующего (IsAttacker)

```
[ alert.Assets[] | select(.Type == "host" and .IsAttacker) | .ID]
```

Тип актива - устройство (host) и имеет признаки жертвы (IsVictim)

```
[ alert.Assets[] | select(.Type == "host" and .IsVictim) | .ID]
```

> Признак "атакующего" и признак "жертвы" настраиваются в правиле корреляции путем обогащения поля AttackerHostID и VictimHostID соответственно. В данные поля могут быть записаны значения из Device/Source/DestinationAssetID.

Пользователи

Имя пользователя `Alice`

```
[ alert.Assets[] | select(.Name == "Alice") | .ID]
```

Тип актива - пользователь (user)

```
[ alert.Assets[] | select(.Type == "user") | .ID]
```

Тип актива - пользователь (user) и имеет признаки атакующего (IsAttacker)

```
[ alert.Assets[] | select(.Type == "user" and .IsAttacker) | .ID]
```

Тип актива - пользователь и имеет признаки жертвы (IsVictim)

```
[ alert.Assets[] | select(.Type == "user" and .IsVictim) | .ID]
```

>

> Признак "атакующего" и признак "жертвы" настраиваются в правиле корреляции путем обогащения поля AttackerUserID и VictimUserID соответственно. В данные поля могут быть записаны значения из Source/DestinationAccountID.

Наблюдаемые объекты

Ниже приведены примеры jq-выражений для извлечения наблюдаемых объектов. Выражение в общем смысле должно возвращать массив значений наблюдаемых объектов.

1. Хэши

1.1 MD5

```
[ alert.Observables[] | select(.Type == "md5") | .Value ]
```

1.2 SHA256

```
[ alert.Observables[] | select(.Type == "sha256") | .Value ]
```

2. IP-адреса

```
[ alert.Observables[] | select(.Type == "ip") | .Value ]
```

3. URL

```
[ alert.Observables[] | select(.Type == "url") | .Value ]
```

4. Все объекты (например, для действия по обогащению)

```
[ alert.Observables[] | .Value ]
```

Информация из событий

Для действий также может быть полезна информация из событий, например, путь к файлу.

```
[ alert.OriginalEvents[] | .BaseEvents[] | select(.DeviceEventClassID == "4688") |
.DestinationProcessName ]
```

Инциденты

В разделе приведены примеры фильтров для раличных типов действий, выполняемых над Инцидентом.

Активы

Ниже приведены примеры jq-выражений для извлечения активов. Выражение в общем смысле должно возвращать массив ID активов.

Устройства

1. Имя устройства `TEST`

```
[ incident.Alerts[] | .Assets[] | select(.Name == "TEST") | .ID]
```

2. Тип актива - устройство (host)

```
[ incident.Alerts[] | .Assets[] | select(.Type == "host") | .ID]
```

3. Тип актива - устройство (host) и имеет признаки атакующего (IsAttacker)

```
[ incident.Alerts[] | .Assets[] | select(.Type == "host" and .IsAttacker) | .ID]
```

Тип актива - устройство (host) и имеет признаки жертвы (IsVictim)

```
[ incident.Alerts[] | .Assets[] | select(.Type == "host" and .IsVictim) | .ID]
```

> Признак "атакующего" и признак "жертвы" настраиваются в правиле корреляции путем обогащения поля AttackerHostID и VictimHostID соответственно. В данные поля могут быть записаны значения из Device/Source/DestinationAssetID.

Пользователи

Имя пользователя `Alice`

```
[ incident.Alerts[] | .Assets[] | select(.Name == "Alice") | .ID]
```

Тип актива - пользователь (user)

```
[ incident.Alerts[] | .Assets[] | select(.Type == "user") | .ID]
```

Тип актива - пользователь (user) и имеет признаки атакующего (IsAttacker)

```
[ incident.Alerts[] | .Assets[] | select(.Type == "user" and .IsAttacker) | .ID]
```

Тип актива - пользователь и имеет признаки жертвы (IsVictim)

```
[ incident.Alerts[] | .Assets[] | select(.Type == "user" and .IsVictim) | .ID]
```

> Признак "атакующего" и признак "жертвы" настраиваются в правиле корреляции путем обогащения поля AttackerUserID и VictimUserID соответственно. В данные поля могут быть записаны значения из Source/DestinationAccountID.

Наблюдаемые объекты

Ниже приведены примеры jq-выражений для извлечения наблюдаемых объектов. Выражение в общем смысле должно возвращать массив значений наблюдаемых объектов.

1. Хэши

1.1 MD5

```
[ incident.Alerts[] | .Observables[] | select(.Type == "md5") | .Value ]
```

1.2 SHA256

```
[ incident.Alerts[] | .Observables[] | select(.Type == "sha256") | .Value ]
```

2. IP-адреса

```
[ incident.Alerts[] | .Observables[] | select(.Type == "ip") | .Value ]
```

3. URL

```
[ incident.Alerts[] | .Observables[] | select(.Type == "url") | .Value ]
```

4. Все объекты (например, для действия по обогащению)

```
[ incident.Alerts[] | .Observables[] | .Value ]
```

Информация из событий

Для действий также может быть полезна информация из событий, например, путь к файлу.

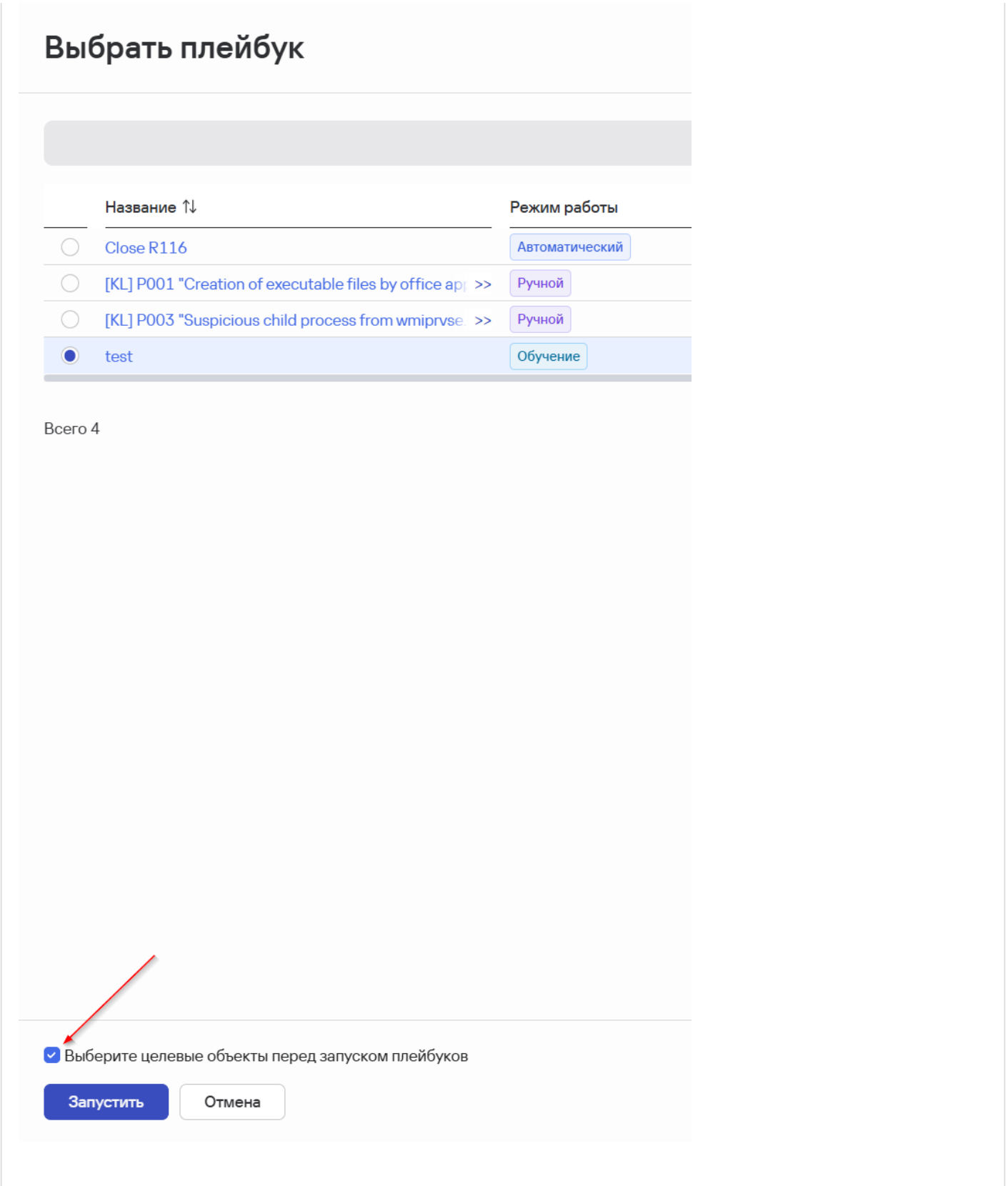
```
[ incident.Alerts[] | .OriginalEvents[] | .BaseEvents[] | select(.DeviceEventClassID == "4688") | .DestinationProcessName ]
```

Выбранные объекты

Помимо возможности автоматического выбора активов и индикаторов для реагирования, их можно также задавать вручную при реагировании. В таком случае синтаксис будет отличаться от используемого в Алертах и Инцидентах при автоматическом выборе, но при этом будет одинаков независимо от места применения плейбука.

Выбор объектов для запуска плейбуков

Выбор объектов для запуска доступен при ручном запуске плейбука. Для этого надо поставить галочку у соответствующей настройки перед запуском плейбука.



Т.к. объекты перед запуском выбирают вручную, то тонкая дополнительная фильтрация лишена смысла. Ниже будут приведены примеры для для указания активов и наблюдаемых объектов, разделенных только по их типам.

Активы

Ниже приведены примеры jq-выражений для извлечения активов. Выражение в общем смысле должно возвращать массив ID активов.

Устройства

```
[ .input.assets[] | select(.Type=="host") | .ID ]
```

Пользователи

```
[ .input.assets[] | select(.Type=="user") | .ID ]
```

Наблюдаемые объекты

Ниже приведены примеры jq-выражений для извлечения наблюдаемых объектов. Выражение в общем смысле должно возвращать массив значений наблюдаемых объектов.

1. Хэши

1.1 MD5

```
[ .input.observables[] | select(.Type == "md5") | .Value ]
```

1.2 SHA256

```
[ .input.observables[] | select(.Type == "sha256") | .Value ]
```

2. IP-адреса

```
[ .input.observables[] | select(.Type == "ip") | .Value ]
```

3. URL

```
[ .input.observables[] | select(.Type == "url") | .Value ]
```

Заключение

Данная статья содержит базовый набор выражений, которые пригодятся при описании действий плейбуков. Если у вас есть замечания или предложения по содержимому статьи, а также собственные идеи для выражений - пишите в наш телеграм-канал (ссылка и инструкция на главной странице).

Полезные ссылки

Модель данных алерта - https://support.kaspersky.com/help/XDR/1.4/ru-RU/269125.htm

Модель данных инцидента - https://support.kaspersky.com/help/XDR/1.4/ru-RU/269168.htm