NTechnology | SIEM

Руководство по написанию правил

НЦОТ

Содержание

1.Общая информация о системе

1.1 О документе

Этот документ содержит информацию о принципах написания правил в системе, предназначенной для сбора и анализа событий информационной безопасности (Security Information and Event Management system) «NTechnology SIEM» (далее – NT SIEM). Руководство содержит рекомендации по созданию правил нормализации, корреляции, агрегации в NT SIEM.

Комплект документации NT SIEM включает в себя следующие документы:

– Этот документ;

– Руководство по установке – содержит информацию для внедрения продукта в инфраструктуре организации: инструкции по установке, первоначальной настройке и удалению продукта;

– Руководство пользователя – содержит справочную информацию и инструкции по настройке и администрированию продукта. Содержит сценарии использования продукта для управления информационными активами организации и событиями информационной безопасности;

– Руководство по созданию запросов – содержит описание наборов запросов и результаты применения этих запросов.

1.2 О NT SIEM

NT SIEM – это система, которая осуществляет сбор, хранение и анализ событий, исходящих от сетевых устройств, средств защиты информации, баз данных, ключевых корпоративных ресурсов, инфраструктуры систем и приложений.

1.3 Краткое описание возможностей системы

Система NT SIEM предоставляет следующие основные функциональные возможности:

- Сбор журналов событий с различных источников;
- Визуализация данных в виде графиков, диаграмм в форме дашбордов;
- Анализ журналов событий в соответствии с правилами нормализации, корреляции, агрегации и обогащения;

- Формирование инцидентов на основе процессов агрегации, обогащения и корреляции;
- Управление инцидентами информационной безопасности;
- Хранение событий и инцидентов информационной безопасности;
- Фильтрация по различным параметрам событий и инцидентов, в том числе с использованием избранных запросов для быстрого доступа к фильтрам по событиям;
- Использование готовой базы правил, а также возможность создания собственных правил и табличных списков;
- Мониторинг состояния системы;
- Отправка уведомлений пользователям в рамках веб-приложения и по электронной почте;
- Формирование и выгрузка отчетов за определенный период времени;
- Осуществление интеграций, в том числе и с SOAR-системами;
- Мониторинг активов.

2.Этапы обработки

NT SIEM осуществляет сбор, хранение и анализ событий из разных источников (рис.1). Источниками событий может выступать любой элемент из IT-инфраструктуры: сетевые устройства, средства защиты информации, базы данных, ключевые корпоративные ресурсы, инфраструктуры систем и приложений.

Сбор данных, необработанных событий, в NT SIEM может осуществлять как в пассивном режиме, так и в активном.

Активный сбор – сбор данных при помощи сторонних программных продуктов – агентов, которые устанавливаются на конечные устройства и отправляют все необходимые логи в систему.

Пассивный сбор – сбор данных безагентским способом, при котором на системе открывается порт под новый источник событий и по протоколу syslog направляются логи в систему.

Для событий, собранных с разных источников, могут отличаться формат и стандарт записи. Для анализа потока событий требуется преобразовать все события к единому виду.

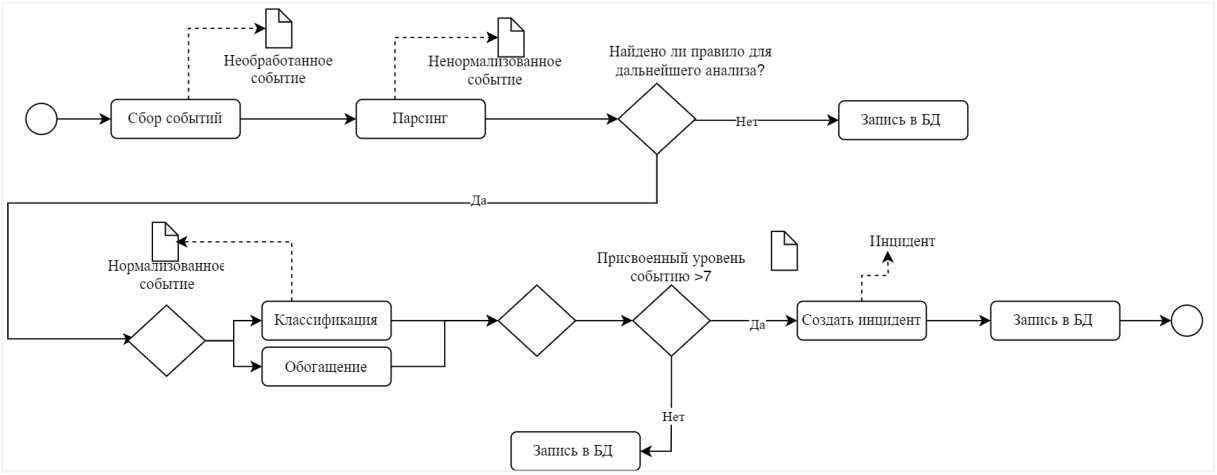

Рисунок 1 – Схема процесса возникновения инцидента

Процесс парсинга (нормализации) – процесс анализа текста события с целью извлечения нужной информации и структурирования по определенным правилам и шаблонам.

Парсеры извлекают информацию из полученных необработанных событий. Парсеры последовательно производят распознавание текста необработанного события, разбиение его на составные элементы и проверка выполнения условий отбора, если таковое условие имеется.

При анализе необработанных событий необходимо использовать язык регулярных выражений (см. пункт 3.1) для того, чтобы извлекать из необработанных событий значения нужных полей (рис 2). Для каждого полученного события система подбирает свой парсер. Глобально парсеры деляться на родительские и дочерние. У родительского может быть неограниченное количество дочерних парсеров, но дочерний не может стать родительским.

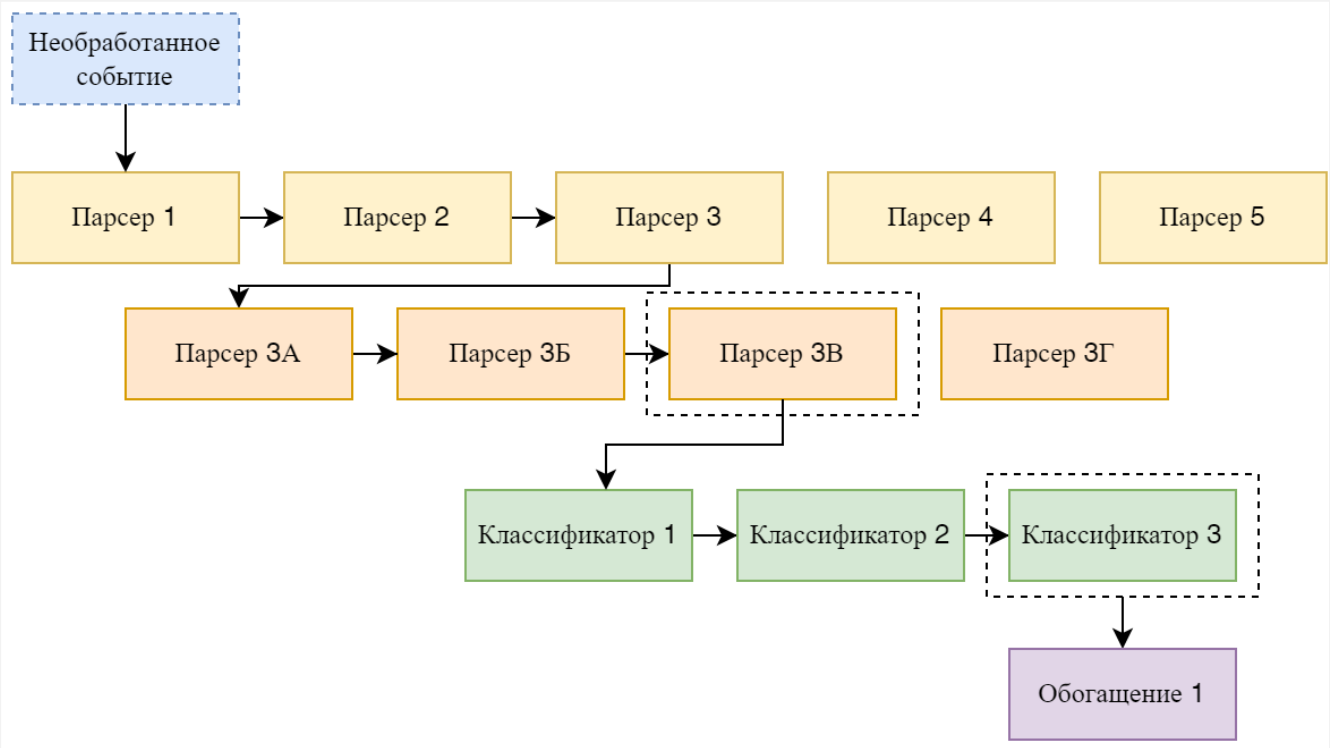

Рисунок 2 – Схема процесса анализа события

Процесс классификации состоит из нескольких подпроцессов: процесс выявления взаимосвязи между разнородными событиями из различных источников в соответствии с правилами корреляции (рис.2), присвоение определенного уровня критичности (табл. 1), процесс агрегации. Параллельно идёт процесс обогащения – добавление информации в событие.

Если в ходе процесса классификации присваивается уровень критичности от 7 и выше, то создается инцидент. Количественный уровень критичности преобразуется в качественный для удобства:

- Низкий (уровень критичность 7-9);
- Средний (уровень критичность 10-12);
- Высокий (уровень критичность 13-15).

Поток событий может содержать однотипные события, отличающиеся значением одного или нескольких полей. Для сокращения количества таких

событий выполняется агрегация событий. Агрегация – процесс объединения нескольких однотипных событий, которые удовлетворяют условию заранее настроенного правила агрегации, в одно событие.

Табличный список – совокупность данных, хранящихся в NT SIEM и используемых в процессе классификации.

Таблица 1 – Классификация уровней

Уро-вень	Название группы	Описание
0-1	Не учитывать	Данный уровень используется во избежание ложных срабатываний. Никаких действий не требуется. Правила корреляции с уровнем 0 сканируются раньше всех остальных. Рекомендуется использовать для событий, не имеющие отношения к безопасности.
2	Системные уведомления низкого приоритета	Данный уровень используется для системных уведомлений или сообщений о состоянии системы.
3	События об успешном получении доступа	К таким типам событий могут относится: успешные попытки входа в систему, события о разрешениях брандмауэра и т.д.
4	Системная ошибка низкого приоритета	Ошибки, связанные с неверными конфигурациями или неиспользуемыми устройствами/ приложениями. Рекомендуется использовать для событий, не имеющие отношения к безопасности, так как обычно такие ошибки возникают при установке или тестировании программного обеспечения.
5	Информация о пользовательской ошибке	К таким типам информации могут относится: пропущенные пароли, запрещенные действия и т. д.

Уро-вень	Название группы	Описание
6	Атака с низкой степенью релевантности	Данный уровень используется для обозначения червей или вирусов, которые не оказывают никакого влияния на систему. К ним также относятся частые события от системы обнаружения вторжений и частые ошибки. Например, красный код для серверов Apache.
7	Соответствие «BAD_WORDS»	К таким словам могут относится: «bad», «error» и т.д. Эти события в большинстве случаев не классифицированы и могут иметь определенное значение для безопасности.
8	Замеченная впервые	Рекомендуется использовать для событий, которые были замечены впервые. При первом запуске системы обнаружения вторжений или при первом входе пользователя в систему.
9	События об ошибках из неизвестного источника	Рекомендуется использовать для событий, связанных с попытками входа в систему от имени неизвестного пользователя или из неизвестного источника. К ним также относятся ошибки, связанные с учетной записью «admin» (root).
10	Многократные пользовательские ошибки	Рекомендуется использовать для событий, связанных с многократными случаями неверно введенных паролей, неудачных входов в систему и т. д.
11	Предупреждение о проверке целостности	К таким предупреждениям относятся сообщения об изменении двоичных файлов или наличии руткитов. Также стоит включать события от системы обнаружения вторжений, которые будут игнорироваться (большое количество повторений). Это может указывать на успешную атаку

Уро-вень	Название группы	Описание
12	Событие с высокой степенью важности	К ним относятся сообщения об ошибках или предупреждения от системы, ядра. Может указывать на атаку на конкретное приложение.
13	Аномалия	В большинстве случаев это соответствует общей схеме атаки.
14	Событие безопасности с высокой степенью важности	В большинстве случаев это связано с корреляцией, и это указывает на атаку.
15	Атака	В случае, когда необходимо немедленное внимание.

3. Создание правил по категориям

3.1 Регулярные выражения

Язык регулярных выражений (RegEx) – это формальный язык, используемый в компьютерных программах, работающих с текстом, для поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов. Для поиска используется строка-образец, состоящая из символов и метасимволов и задающая, правило поиска.

В системе регулярные выражения записываются в атрибут regex, существует 3 типа: regex (OS_Regex), sregex (OS_Match) and PCRE2.

1.Синтаксис регулярных выражений OS_Regex.

Ограничения:

1. Метасимволы «*» и «+» (табл. 3) можно применять только к выражениям с «\», а не к простым символам. Например:
   a. \d+ поддерживается;
   b. 0+ не поддерживается.
2. Нельзя использовать чередование в группе. Например:
   a. (foo|bar)
3. Сложный возврат не поддерживается. Например:

a. \p*\d*\s*\w*: не соответствует ни одному двоеточию, поскольку «\p*» использует двоеточие

4. Нет синтаксиса для поиска символов «^», «*» или «+», вариант с использованием «\» для поиска данных символов невозможен. Обратите внимание, «\p» будет соответствовать «*» или «+», а также некоторым другим символам (табл.2).

Таблица 2 – Синтаксис OS_Regex, представления

Предста-вление	Эквивалент	Значение
\w	[A-Za-z0-9],[-, @, _]	Символы, образующие «слово»
\d	[0-9]	Цифра
\s	Пробел	Соответствует только пробелу ASCII (32), а не другим пробелам, например, табуляции.
\t	TAB	
\p	()*+,-.:;<=>?[]!'"#$%&|{}	
\W	[^\w]	Символы, не образующие «слово»
\D	[^\d]	Любой символ, кроме цифры
\S	[^\s]	Любой символ, кроме пробела

Таблица 3 – Синтаксис OS_Regex, метасимволы

Метасим-вол	Описание	Примеры
\.	Соответствует любому символу	a.b – найдет любой символ между, например «abc»,« a-b», «a2b»
.	Соответствует точке	example. – найдет «example.xml», «example.csv»
^	Для обозначения начала строки	^[а-я] – абзац, начинающийся со строчной буквы

Метасим-вол	Описание	Примеры
$	Для обозначения конца строки	^$ – пустая строка
[ ]	Набор символов	0x[0a-c] – числа, начинающиеся с 0×0, 0xa, 0xb или 0xc
|	Для создания логического «или» между несколькими выражениями	рабоч(его|ее) – ищем «рабочего» или «рабочее»
*	Для совпадения 0 или более раз	^.* – выделение строк (пустых и не пустых)
+	Для совпадения 1 или более раз	^.+ – выделение непустых строк (абзацев)
\	Для того чтобы искать символ, совпадающий с метасимволом, его необходимо отделить обратной косой чертой	\$ \(\) \\ | \<

2. Синтаксис регулярных выражений Sregex (OS_Match).

Данный тип регулярных выражений обрабатывается быстрее, но поддерживает только простое сопоставление строк и некоторые специальные метасимволы (табл. 4).

Таблица 4 – Синтаксис OS_Match, метасимволы

Метасимвол	Описание	Пример
^	Для обозначения начала строки	^example – текст начинается с данного выражения
$	Для обозначения конца строки	example$ – конец текста заканчивается на данное выражение

Метасимвол	Описание	Пример
|	Для создания логического “или” между несколькими выражениями	example|test – найдет «example» или «test», или и то, и то
!	Для отрицания, исключения выражения.	!example – найдет все, кроме «example»

3. Синтаксис регулярных выражений PCRE2.

Perl Compatible Regular Expressions (далее – PCRE) предоставляет такие функции, как рекурсивные шаблоны, утверждения просмотра вперед и назад, группы без захвата, нежадные квантификаторы, расширенный синтаксис для метасимволов и классов метасимволов и многие другие (табл.5-8).

Таблица 5 – Синтаксис PCRE, представления

Предста-вление	Эквивалент	Значение
\w	[A-Za-z0-9],[-, @, _]	Символы, образующие «слово»
\d	[0-9]	Цифра
\h	Любой горизонтальный пробельный символ	
\s	[\t\r\n\f]	Любой пробельный символ
\p	()*+,-.:;<=>?[]!'"#$%&|{}	
\S	[^\s]	Любой символ, кроме пробела
\W	[^\w]	Символы, не образующие «слово»
\D	[^\d]	Любой символ, кроме цифры
\H	[^\h]	Любой символ, кроме \h

В синтаксисе PCRE есть метасимволы – специальные символы, которые используются для составления шаблонов и определений правил поиска.

Таблица 6 – Синтаксис PCRE, метасимволы

Метасим-вол	Описание	Примеры
.	Соответствует любому символу, кроме новой строки	a.b – найдет любой символ между, например «abc»,« a-b», «a2b»
^	Для обозначения начала строки	^[а-я] – абзац, начинающийся со строчной буквы
$	Для обозначения конца строки	^$ – пустая строка
[ ]	Набор символов	0x[0a-c] – числа, начинающиеся с 0×0, 0xa, 0xb или 0xc
()	Группировка	рабоч(его|ее) – ищем «рабочего» или «рабочее»
|	Для создания логического “или” между несколькими выражениями	рабоч(его|ее) – ищем «рабочего» или «рабочее»
\	Для того чтобы искать символ, совпадающий с метасимволом, его необходимо отделить обратной косой чертой	\$ \(\) \\ | \<

Метасимволы имеют квантификаторы (пишутся после метасимвола). Они могут быть greedy («жадными»), possessive («промежуточный, между жадный и ленивый»), lazy («ленивый»).

Таблица 7 – Синтаксис PCRE, метасимволы

Квантификаторы	Описание	Качество
?	Повторяется 1 или 0 раз	greedy

Квантификаторы	Описание	Качество
?+	Повторяется 0 или 1 раз	possessive
??	Повторяется 0 или 1 раз	
*	Повторяется 0 или более раз	greedy
*+	Повторяется 0 или более раз	possessive
*?	Повторяется 0 или более раз	lazy
+	Повторяется 1 или более раз	greedy
++	Повторяется 1 или более раз	possessive
+?	Повторяется 1 или более раз	lazy
{n}	Повторяется точное n раз	
{n,m}	Как минимум n раз, не более чем m раз	greedy
{n,m}+	Как минимум n раз, не более чем m раз	possessive
{n,m}?	Как минимум n раз, не более чем m раз	lazy
{n,}	По меньшей мере 1 раз	greedy
{n,}+	Повторяется n или более раз	possessive
{n,}?	Повторяется n или более раз	lazy

«Жадный» (greedy). Квантификатор сопоставляет с максимальным количеством символов, удовлетворяющих шаблону, и пытается расширить свою область до максимальной длины, при этом все еще удовлетворяя шаблону.

Например, в выражении .*foo, .* сначала «поглощает» всю входную строку. Затем, если общее выражение не может выполниться, поскольку последние три буквы («f», «o», «o») уже были «поглощены», то сопоставитель медленно отступает на одну букву за раз, пока не найдет правое вхождение «foo».

«Неохотные» (Lazy или Reluctant). Квантификатор сопоставляет с наименьшим количеством символов, удовлетворяющих шаблону, и расширяет свою область только при необходимости удовлетворения остальной части

шаблона. Таким образом, он предпочитает минимальное количество символов, чтобы шаблон все еще был совместимым.

Например, в выражении .*?foo, .*? начинает с «поглощения» ничего. Поскольку «foo» не стоит в начале строки, он вынужден «поглотить» первую букву (например, «x»), что вызывает первое совпадение.

«Промежуточные» (Possessive). Квантификатор сопоставляет с максимальным количеством символов, удовлетворяющих шаблону, и отказывается от дополнительных попыток, чтобы дать другим частям шаблона возможность сопоставить. Это полезно в случаях, когда дополнительные попытки не приведут к положительному результату, так как шаблон уже находится в пределах своего допустимого диапазона.

Например, в выражении .*+foo, .*+ «поглощает» всю входную строку, не оставляя ничего для “foo” в конце выражения. Это выражение не найдет совпадение.

Таблица 8 – Синтаксис PCRE, иные символы и выражения

Символ	Описание
\f	Подача страницы (шестнадцатеричный 0C)
\n	Новая строка
\r	Возврат каретки
\t	Табуляция
\0dd	Символ с восьмеричным кодом 0dd
\o{ddd..}	Символ с восьмеричным кодом ddd.
\xhh	Символ с шестнадцатеричным кодом hh
\x{hh..}	Символ с шестнадцатеричным кодом hh.

3.2 Работа с процессом парсинга

3.2.1 Синтаксис для процесса парсинга

Для осуществления процесса парсинга необходимы специальные парсеры, в которых описаны алгоритмы и шаблоны для данного процесса (табл.9).

Таблица 9 – Атрибуты парсингов

Атрибут	Возможные значения	Описание
decoder	Любое уникальное имя правила нормализации	Атрибут, определяющий имя декодера. Например: <**decoder name**="test_decoder"> … </**decoder**>
parent	Любое имя существующего правила нормализации	Ссылаемся на родительский парсер, а создаваемый парсер станет дочерним. Обратите внимание, что родительский парсер может иметь много дочерних, но дочерний не может стать родительским (пункт 3.1.1.3). <**decoder** name="decoder_junior"> <**parent**>decoder_father</**parent**> … </**decoder**> Примечание:*Decoder_junior* войдет только в том случае, если *decoder_parent* ранее создали пару.
accumulate	-	Позволяет отслеживать события по нескольким сообщениям журнала.
ftscomment	Любое значение типа String	Добавляет комментарий к атрибуту «**fts**».
Атрибут	Возможные значения	Описание

program_name	Regex, sregex или pcre2 выражения (см. пункт. 3.1)	Устанавливает имя программы как условие применения парсера. Заголовок лога должен содержать имя программы, соответствующее регулярному выражению. Например: <**decoder** name="test_decoder"> <**program_name** type="pcre2">(?i)test</program_name > … </**decoder**>
prematch	Regex или pcre2 выражения (см. пункт. 3.1)	Устанавливает регулярное выражение в качестве условия применения парсера. Лог должен соответствовать регулярному выражению без учета заголовков, подобных системному.
order	См. табл. 12	Значения, которые извлекает регулярное выражение, будут храниться в этих группах. Определяет, что содержат группы в скобках, и порядок их получения.
use_own_name	По умолчанию значение: n/a Допустимые значения:True	Только для дочерних парсеров. Позволяет установить имя дочернего парсера из атрибута имени вместо использования имени родительского.

Атрибут	Возможные значения	Описание

regex	Regex или pcre2 выражения (см. пункт. 3.1)	Используется для поиска интересующих полей и их извлечения по шаблону. Например: <**regex**> [+-]?(\d+(\.\d+)?|\.\d+)([eE][+-]?\d+)? </**regex**> При использовании данного поля обязательно также необходимо определить «**order**». Кроме того, для «**regex**» требуется «**prematch**» или «**program_name**», определенные в том же или в родительском парсере. Например: <**decoder** name="sudo-fields"> <**parent**>sudo</**parent**> <**prematch**>\s</**prematch**> <**regex**>^\s*(\S+)\s*:</**regex**> <**order**>srcuser</**order**> <**fts**>name,srcuser,location</**fts**> <**ftscomment**>First time user executed the sudo command</**ftscomment**> </**decoder**>
fts	См.табл. 12	Используется для обозначения парсеров, при срабатывании которых, администратор хотел бы получить уведомление. Например, извлечение пользователя, который сгенерировал предупреждение, и место, откуда оно пришло: <**decoder** name="fts-decoder"> <**fts**>srcuser, location</**fts**> … </**decoder**>
Атрибут	Возможные значения	Описание

type	См. табл. 10	Определяет тип логов, которым парсеры будут соответствовать. Например: <**decoder**> <**type**>syslog</**type**> … </**decoder**>
plugin_decoder	По умолчанию значение: n/a Допустимые значения: PF_Decoder SymantecWS_Decoder SonicWall_Decoder	Указывает плагин, который будет выполнять декодирование. Используется в случаях невозможности извлечение с помощью регулярного выражения. PF_Decoder – предназначен для процесса парсинга и анализа событий, связанных с Packet Filter (PF) – это инструмент фильтрации пакетов в операционной системе FreeBSD. SymantecWS_Decoder – предназначен для процесса парсинга и анализа событий, связанных с продуктами безопасности от Symantec, такими как Symantec Endpoint Protection (SEP). SonicWall_Decoder – предназначен для процесса парсинга и анализа событий, связанных с устройствами SonicWall, такими как брандмауэры и межсетевые экраны.

Атрибут	Возможные значения	Описание
var	Имя переменной	Определяет переменные, которые можно повторно использовать

		внутри одного и того же файла. Например: <var name="header">myprog</var> <var name="offset">after_parent</var> <var name="type">syscall</var> <decoder name="syscall"> <prematch>^$header</prematch> </decoder> <decoder name="syscall-child"> <parent>syscall</parent> <prematch offset="$offset">^:$type </prematch> <regex offset="after_prematch">(\S+)</regex> <order>syscall</order> </decoder>

Таблица 10 – Типы логов

Значение по умолчанию	syslog
Допустимые значения	firewall
	ids
	web-log
	syslog
	squid
	windows
	host-information

Таблица 11 – Данные, которые может хранить атрибут «order»

Значение по умолчанию	n/a

Статические поля	srcuser	Извлекает исходное имя пользователя
	dstuser	Извлекает целевое имя пользователя
	user	То же, что и dstuser (однако использовать необходимо только одно из полей)
	srcip	Исходный IP-адрес
	dstip	Целевой IP-адрес
	srcport	Исходный порт
	dstport	Целевой порт
	protocol	Протокол
	system_name	Имя системы
	id	ID события
	url	Ссылка на событие
	action	Действие над событием (deny, drop, accept, etc.)
	status	Статус события (success, failure, etc.)
	data	Любые данные
	extra_data	Любые дополнительные данные
Динамические поля	Любая String, не включенная в предыдущий список	

Таблица 12 – Данные для извлечения для отправки уведомления

Значение по умолчанию	n/a

Допустимые значения	location	Откуда поступает лог
	srcuser	Извлекает исходное имя пользователя
	dstuser	Извлекает целевое имя пользователя
	user	То же, что и dstuser (однако использовать необходимо только одно из полей)
	srcip	Исходный IP адрес
	dstip	Целевой IP адрес
	srcport	Исходный порт
	dstport	Целевой порт
	protocol	Протокол
	system_name	Имя системы
	id	ID события
	url	Ссылка на событие
	action	Действие над событием (deny, drop, accept, etc.)
	status	Статус события (success, failure, etc.)
	data	Данные
	extra_data	Любые дополнительные данные

3.2.2 Пример написания парсеров

Необработанные события поступают в систему в разном формате, и для разных форматов используются специализированные алгоритмы обработки текста события (парсеры). Схема полей событий представлена в Приложении.

При создании пользовательского парсера желательно иметь в качестве примеров несколько записей о событиях одного и того же типа. Так как одно и то же событие в различных случаях может содержать или не содержать те или

иные данные. Следует обратить внимание, что для пользовательских парсеров следует использовать идентификационные номера от 100000 до 120000.

3.2.3 Динамические и статические поля

Статистические поля – предопределенные поля для хранения извлеченной информации из необработанного события. Всего существует 13 статистических полей (табл. 3). Однако одновременное извлечение возможно только из 8 полей. Пример:

```
<decoder name="web-accesslog">
  <type>web-log</type>
  <prematch>^\d+.\d+.\d+.\d+ - </prematch>
  <regex>^(\d+.\d+.\d+.\d+) - \S+ [\S+ -\d+] </regex>
  <regex>"\w+ (\S+) HTTP\S+ (\d+) </regex>
  <order>srcip,url,id</order>
</decoder>
```

Если списка из 13 статистических полей или одновременного извлечения 8 полей недостаточно, следует воспользоваться динамическими полями и родственными парсерами (см. пункт 3.1.1.2). Пример:

```
<decoder name="auditd-config_change">
  <parent>auditd</parent>
  <regex offset="after_regex">^auid=(\S+) ses=(\S+) op="(\.+)"</regex>
  <order>audit.auid,audit.session,audit.op</order>
</decoder>
```

Система преобразует любое поле в «<**order**>» в поле JSON. По умолчанию количество полей, которые можно извлечь в «<**order**>», равно 64.

В следующем примере показано, как парсер «Auditd» извлекает информацию из необработанного события:

```
** Alert 1486483073.60589: - audit,audit_configuration,
2017 Feb 07 15:57:53 siem-example->/var/log/audit/audit.log
Rule: 80705 (level 3) -> 'Auditd: Configuration changed'
type=CONFIG_CHANGE msg=audit(1486483072.194:20): auid=0 ses=6 op="add rule" key="audit-wazuh-a" list=4 res=1
audit.type: CONFIG_CHANGE
audit.id: 20
```

audit.auid: 0
audit.session: 6
audit.op: add rule
audit.key: audit
audit.list: 4
audit.res: 1

Далее показан результат преобразования в формат JSON:


```
{
 "rule": {
  "level": 3,
  "description": "Auditd: Configuration changed",
  "id": 80705,
  "firedtimes": 2,
  "groups": [
   "audit",
   "audit_configuration"
  ]
 },
 "agent": {
  "id": "000",
  "name": "wazuh-example"
 },
 "manager": {
  "name": "wazuh-example"
 },
 "full_log": "type=CONFIG_CHANGE msg=audit(1486483072.194:20): auid=0 ses=6 op=\"add rule\" key=\"audit-wazuh-a\" list=4 res=1",
 "audit": {
  "type": "CONFIG_CHANGE",
  "id": "20",
  "auid": "0",
  "session": "6",
  "op": "add rule",
  "key": "audit",
  "list": "4",
  "res": "1"
```

```
  },
  "decoder": {
    "parent": "auditd",
    "name": "auditd"
  },
  "timestamp": "2017 Feb 07 15:57:53",
  "location": "/var/log/audit/audit.log"
}
```


3.2.4 Родственные парсеры

Родственные парсеры – иерархические парсеры, основная цель которых упрощение процесса парсинга для больших объемов информации из различных событий (рис.2).

Поступившее в систему событие будет последовательно сопоставляться с каждым парсером без «родителя». Когда событие прошло соответствие условию любого из них, далее оно проходит анализ по его дочерним элементам.

Важно понимать, что как только событие сопоставляется с определенным родительским парсером, оно перестанет просматриваться остальными из набора правил. По этой причине при построении парсеров следует воздержаться от слишком общих условий, во избежание ложных срабатываний или пропусков нужных парсеров.

В процессе сопоставления на уровне парсера используются регулярные выражения (см.пункт 3.1), которые требуют, чтобы совпадающая строка имела определенную структуру. Однако события часто предоставляют информацию, пропуская части или изменяя порядок, следовательно, создание множества дочерних парсеров, для соответствия каждой из возможных комбинаций, является неэффективным и непроизводительным.

Для решения данной проблемы можно создать набор дочерних парсеров, которые вместе являются «родительскими» сами по себе. В результате, когда один из этих парсеров сопоставлен, он также проверяет «родственные» парсеры, извлекая по одному фрагменту информации за раз (рис.3).

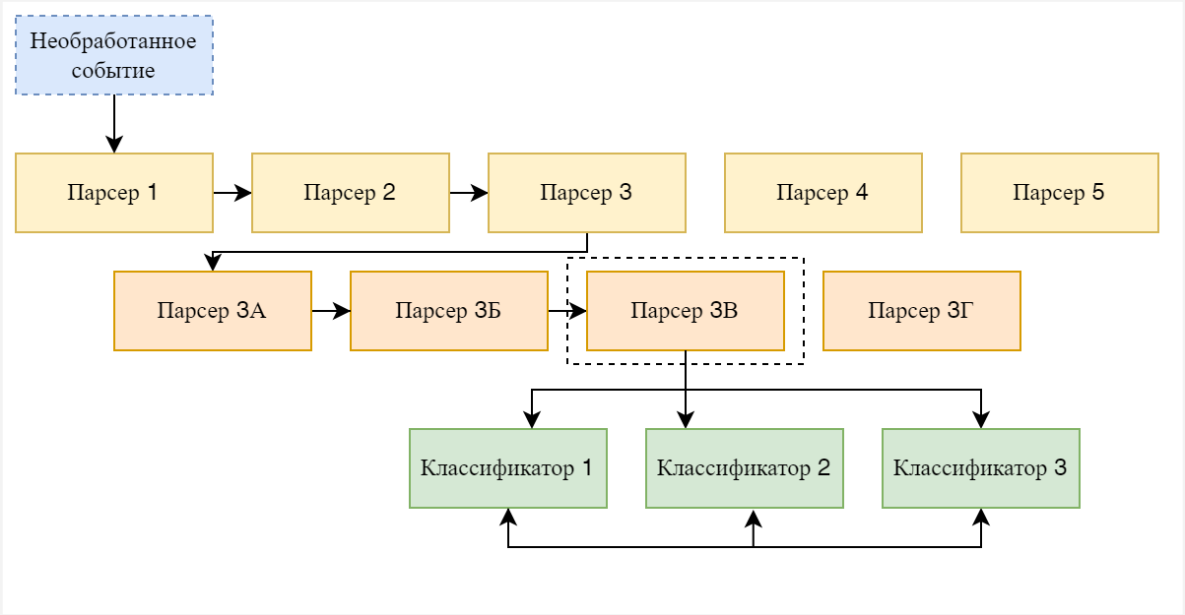

Рисунок 3 – Схема процесса анализа события с параллельным поиском по дочерним

Как следствие, если порядок был изменен или появилась дополнительная информациия, то анализируемый модуль сможет проанализировать и извлечь из события максимум данных.

Далее рассмотрим на практическом примере. События поступили в следующем виде:

1. Vnimanie: computer1 – dostup k vredonosnomy website - https://plohoywebsite.com
2. Vnimanie: computer2 – dostup k vredonosnomy website - https://ochenplohoywebsite.com
3. Vnimanie: computer3 – dostup k vredonosnomy website - https://neochenplohoywebsite.com

Как видно из примера, текст «Vnimanie:» всегда присутствует в начале каждого пришедшего события, поэтому его можно использовать в предварительном сопоставлении родительского правила парсинга. Также возможно использование регулярного выражения (см. пункт 3.1) для создания родительского правила парсинга.

Родительский парсер представляет собой:

```
<decoder name="parent_test">
  <prematch>^Vnimanie:</prematch>
</decoder>
```

В приведенном родительском правиле установлено имя носителя «"parent_test"» (это может быть любое имя), а для предварительного сопоставления установлено регулярное выражение «^Vnimanie». Следующее правило попытается найти в начале каждого события совпадение с выражением «Vnimanie:» и решить, следует ли триггерить дочерние правила для дальнейшего процесса парсинга.

Результат тестирования:

**Messages:
WARNING: (7003): '238ff9d8' token expires
INFO: (7202): Session initialized with token '7917436f'
**Phase 1: Completed pre-decoding.
full event: 'Vnimanie: computer1 - dostup k vredonosnomy website - https://plohoywebsite.com'
**Phase 2: Completed decoding.
name: 'parent_test'
**Phase 1: Completed pre-decoding.
full event: 'Vnimanie: computer2 - dostup k vredonosnomy website - https://ochenplohoywebsite.com'
**Phase 2: Completed decoding.
name: 'parent_test'
**Phase 1: Completed pre-decoding.
full event: 'Vnimanie: computer3 - dostup k vredonosnomy website - https://neochenplohoywebsite.com'
**Phase 2: Completed decoding.
name: 'parent_test'

Теперь нам нужно создать дочерний парсер, который проанализирует поля, в том числе содержащие динамические значения, и сохранит их. Дочерний парсер представляет собой:

```
<decoder name="child_test">
  <parent>parent_test</parent>
  <regex offset="after_parent">^\s(\.+) - dostup k vredonosnomy website - (https://\.+)</regex>
  <order>computer_name,website</order>
</decoder>
```

Имя парсера указывается в поле «<**decoder** name="child_test">». Имя должно быть уникальным. Далее следует указать «родителя» для дочернего элемента «<**parent**>parent_test</**parent**>».

Затем с помощью регулярных выражений извлекаем из сырого события данные. Для того чтобы дочерним парсером начал выполнять определенные действия после завершения работы родительского, необходимо использовать опцию «after_parent».

Далее следует создать остальную часть регулярного выражения:

1. Чтобы начать поиск сначала текста после родительского совпадения, опции «after_parent», следует использовать метасимвола «^»;
2. Использование выражение «\s» соответствует пробелу;
3. Использование «(\.+)» подразумевает совпадение с любым символом или фразой, например, Computer1;
4. Использование «- dostup k vredonosnomy website -» соответствует статическому тексту, который будет присутствовать в каждом событии, Данный текст получается от родительского парсера;
5. Использование «(https://\.+)» подразумевает совпадение с любым символом или фразой, что будет после «https:// ».

Атрибут «<**order**>» определяет, что будет содержаться в группах в круглых скобках, и порядок их получения. В данном примере, будут значения для заголовков «**computer_name**» и «**website**». Атрибут «<**regex**>» найдет поля и извлечет их с помощью оператора (). В случае примера 1, будут извлечены значения «**computer1**» и «**https://plohoywebsite.com**».

Результат тестирования:

```
**Messages:
   WARNING: (7003): '238ff9d8' token expires
   INFO: (7202): Session initialized with token '954e2d0c'
 **Phase 1: Completed pre-decoding.
   full event: 'Vnimanie: computer1 - dostup k vredonosnomy website -
https://plohoywebsite.com'
 **Phase 2: Completed decoding.
   name: 'parent_test'
   computer_name: 'computer1'
   website: 'https://plohoywebsite.com'
 **Phase 1: Completed pre-decoding.
```

full event: 'Vnimanie: computer2 - dostup k vredonosnomy website - https://ochenplohoywebsite.com'
**Phase 2: Completed decoding.
name: 'parent_test'
**Phase 1: Completed pre-decoding.
full event: 'Vnimanie: computer3 - dostup k vredonosnomy website - https://neochenplohoywebsite.com'
**Phase 2: Completed decoding.
name: 'parent_test'
computer_name: 'computer3'
website: 'https://neochenplohoywebsite.com'

3.3 Работа с процессом классификации

3.3.1 Синтаксис для процесса классификации

Как только событие прошло процесс парсинга, оно переходит на этап классификации (рис.1). В данном разделе будет описываться синтаксис, используемый в процессе классификации.

Таблица 14 – Атрибуты правил

Атрибут	Возможные значения	Описание
group		Позволяет классифицировать правила по определенным категориям. Каждое правило должно принадлежать хотя бы к одной группе. Чтобы определить правила, которые срабатывают только в том случае, если сработало другое правило в определенной группе, используйте аттрибуты if_group и if_matched_group. Например, указание принадлежности правила к нескольким группам: <group name="GROUP1_NAME,GROUP2_NAME,"> <rule id="100234" level="3"> <if_sid>230</if_sid> <field name="alert_type">normal</field>

		<description>The file limit set for this agent is $(file_limit). Now, $(file_count) files are being monitored.</description> </rule> </group>
Атрибут	Возможные значения	Описание
rule	См.табл. 16	Используется для описания правил. Пример: <rule id="3151" level="10" frequency="8" timeframe="120"> <if_matched_sid>3102</if_matched_sid> <description>sendmail: Sender domain has bogus MX record. </description> <description>It should not be sending e-mail.</description> <mitre> <id>T1114</id> <id>T1499</id> </mitre> <group>multiple_spam,pci_dss_11.4,gdpr_IV_35.7.d,nist_800_53_SI.4,tsc_CC6.1,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group> </rule>
match	По умолчанию значение: n/a По умолчанию использует sregex, но можно и любое регулярное выражения (см. пункт. 3.1)	Данный атрибут отвечает за поиск соответствия указанных данных в правиле и в событии и определяет, нужно ли выполнить правило. Пример, если найдено соответствие с id=100200 и событие содержит фразу «Queue flood!», то срабатывает правило и присваивается уровень критичности 3: <rule id="100001" maxsize="300" level="3"> <if_sid>100200</if_sid> <match>Queue flood!</match> <description>Flooded events queue.</description> </rule>

Атрибут	Возможные значения	Описание
regex	По умолчанию значение: n/a По умолчанию использует regex, но можно и любое регулярное выражения (см. пункт. 3.1)	Аналогично атрибуту «<match>». Пример, если найдено соответствие с id=100200 и событие содержит действующий IP-адрес, срабатывает правило и присваивается уровень критичности 3: <rule id="100001" level="3"> <if_sid>100500</if_sid> <regex>\d+.\d+.\d+.\d+</regex> description>Matches any valid IP</description> </rule>
decoded_as	По умолчанию значение: n/a Имя парсера	Необходимое условие для запуска правила. Если событие прошло процесс парсинга по определенному парсеру, запускается правило. Пример: <rule id="53500" level="0"> <decoded_as>smtpd</decoded_as> <description>OpenSMTPd grouping. </description> </rule>
category	По умолчанию значение: n/a Любой тип	Необходимое условие для запуска правила. Правило запускается, если парсеру присвоена определенная категория. Пример: <rule id="01" level="0" noalert="1"> <category>syslog</category> <description>Generic template for all syslog rules. </description> </rule>

Атрибут	Возможные значения	Описание
field	По умолчанию значение: n/a Любое имя или sregex, pcre2 выражение	Необходимое условие для запуска правила. Проверяет на соответствие значение в определенном поле. Имеет свои атрибуты: 1. name – указывает имя поля, извлеченного парсером; 2. negate – используется для отмены регулярного выражения. Возможные значения: no, yes. По умолчанию, используется no; 3. type – устанавливает тип регулярного выражения Возможные значения: osregex, osmatch, pcre2. По умолчанию, используется osregex.
srcip	По умолчанию значение: n/a Любой IP-адрес	Необходимое условие для запуска правила. Проверяет на соответствие введенный IP-адрес с любыми IP-адрес или CIDR, которые есть в поле «srcip» в распарсенных событиях (см. табл.10). Используется символ «!» для отрицания. Пример, если приходит событие с IP-адреса «10.25.23.12», то запуститься правило и присвоится уровень 8: <rule id="100105" level="8"> <if_sid>100100</if_sid> <srcip>10.25.23.12</srcip> <description>Forbidden srcip has been detected.</description> </rule>

Атрибут	Возможные значения	Описание
dstip	По умолчанию значение: n/a Любой IP-адрес	Необходимое условие для запуска правила. Проверяет на соответствие введенный IP-адрес с любыми IP-адрес или CIDR, которые есть в поле «dstip» в распарсенных событиях (см. табл.10). Используется символ «!» для отрицания. Пример, если целевой IP-адрес не «198.168.41.30», то то запуститься правило и присвоится уровень 5: <rule id="100110" level="5"> <if_sid>100100</if_sid> <dstip>!198.168.41.30</dstip> <description>A different dstip has been detected.</description> </rule>
srcport	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1)	Необходимое условие для запуска правила. Проверяет на соответствие введенный порт с любым портом, которые есть в поле «srcport» в распарсенных событиях (см. табл.10). Пример, если порт источника находится в диапазоне от 50 000 до 50 007, то сработает правило и присваивается уровень 5: <rule id="100110" level="5"> <if_sid>100100</if_sid> <srcport type="pcre2">^5000[0-7]$</srcport> <description>Source port $(srcport) is detected.</description> </rule>
dstport	По умолчанию значение: n/a Любое регулярное	Необходимое условие для запуска правила. Проверяет на соответствие введенный порт с любым портом, которые есть в поле «dstport» в распарсенных событиях (см. табл.10).

	выражение (см. пункт 3.1)	

Атрибут	Возможные значения	Описание
data	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1.)	Необходимое условие для запуска правила. Проверят на соответствие данным, которые есть в поле «data» в распарсенных событиях (см. табл.10).
extra_data	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1)	Необходимое условие для запуска правила. Проверят на соответствие данным, которые есть в поле «extra_data» в распарсенных событиях (см. табл.10). Пример, если событие принадлежит категории «windows», а распарсенное поле «extra_data» содержит «Symantec AntiVirus», то сработает правило и присвоит уровень 0: <rule id="7301" level="0"> <category>windows</category> <extra_data>^Symantec AntiVirus </extra_data> <description>Grouping of Symantec AV rules from eventlog.</description> </rule>
user	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1)	Необходимое условие для запуска правила. Проверяет имя пользователя. Пример, если пользователь не «admin» или «root» успешно войдет в систему, сработает правило и присвоит уровень 12. <rule id="140101" level="12"> <if_group>authentication_success</if_group> <user negate="yes">admin|root</user> <description>Unexpected user successfully logged to the system.</description>

		</rule>

Атрибут	Возможные значения	Описание
system_name	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1)	Необходимое условие для запуска правила. Проверяет имя системы.
program_name	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1)	Необходимое условие для запуска правила. Проверяет имя программы. Пример, при перезапуске программы «syslogd» сработает правило и присвоит уровень 5. <rule id="1005" level="5"> <program_name>syslogd</program_name> <match>^restart</match> <description>Syslogd restarted.</description> <group>pci_dss_10.6.1,gpg13_10.1,gpg13_4.14,gdpr_IV_35.7.d,hipaa_164.312.b,nist_800_53_AU.6, </group> </rule>
protocol	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1)	Необходимое условие для запуска правила. Проверяет протокол.

Атрибут	Возможные значения	Описание
hostname	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1)	Необходимое условие для запуска правила. Проверяет имя хоста или лог-файла. Пример, при действии установки, обновления или удаления на хосте «yum.log$», сработает правило и присвоит уровень 0: <rule id="2931" level="0"> <hostname>yum.log$</hostname> <match>^Installed|^Updated|^Erased</match> <description>Yum logs.</description> </rule>
time	По умолчанию значение: n/a Любой временной диапазон	Необходимое условие для запуска правила. Проверяет время создания события. Пример, при успешном входе в систему с 18:00 по 08:00 сработает правило и присвоит уровень 9: <rule id="17101" level="9"> <if_group>authentication_success</if_group> <time>6 pm - 8:30 am</time> <description>Successful login during non-business hours.</description> <group>login_time,pci_dss_10.2.5,pci_dss_10.6.1,gpg13_7.1,gpg13_7.2,gdpr_IV_35.7.d,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,nist_800_53_AU.6,</group> </rule>
id	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1)	Необходимое условие для запуска правила. Проверяет идентификатор. Пример, <rule id="81100" level="0"> <decoded_as>kernel</decoded_as> <id>usb</id> <description>USB messages grouped.</description>

		</rule>

Атрибут	Возможные значения	Описание
weekday	По умолчанию значение: n/a monday - sunday, weekdays, weekends	Необходимое условие для запуска правила. Проверяет дни недели, когда было создано правило. Пример, при успешном входе в систему в выходные дни сработает правило и присвоит уровень 9: <rule id="17102" level="9"> <if_group>authentication_success</if_group> <weekday>weekends</weekday> <description>Successful login during weekend.</description> <group>login_day,pci_dss_10.2.5,pci_dss_10.6.1,gpg13_7.1,gpg13_7.2,gdpr_IV_35.7.d,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,nist_800_53_AU.6,</group> </rule>
url	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1)	Необходимое условие для запуска правила. Проверяет URL-адрес. Пример, срабатывает дочернее правило и присваивается уровень 0: <rule id="31102" level="0"> <if_sid>31101</if_sid> <url>.jpg$\|.gif$|favicon.ico$\|.png$|robots.txt$\|.css$|.js$\|.jpeg$</url> <compiled_rule>is_simple_http_request</compiled_rule> <description>Ignored extensions on 400 error codes.</description> </rule>

Атрибут	Возможные значения	Описание
location	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1)	Необходимое условие для запуска правила. Проверяет локацию (см. табл.17). Пример, если события поступают из «osquery$», то присваивается уровень 3: <rule id="24000" level="3"> <location>osquery$</location> <description>osquery message</description> </rule>
action	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1) или тип данных String	Необходимое условие для запуска правила. Проверяет действие. Пример, дочернее правило, присвоит уровень 4, при условии, что действия с «Netscreen» были «warning» или «WARN»: <rule id="4502" level="4"> <if_sid>4500</if_sid> <action type="osregex"> warning|WARN </action> <description> Netscreen warning message. </description> </rule>
status	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1)	Проверяет статус события. Пример: <rule id="213" level="7"> <if_sid>210</if_sid> <status>aborted</status> <description> Remote upgrade could not be launched. Error: $(error).</description> <group>upgrade,upgrade_failure,</group> </rule>
srcgeoip	По умолчанию значение: n/a Любое регулярное выражение (см.	Необходимое условие для запуска правила. Проверяет источник GeoIP.

	пункт 3.1)	

Атрибут	Возможные значения	Описание
dstgeoip	По умолчанию значение: n/a Любое регулярное выражение (см. пункт 3.1)	Необходимое условие для запуска правила. Проверяет пункт назначения GeoIP.
if_sid	По умолчанию значение: n/a ID правил, отделенные запятых или пробелмаи.	Необходимое условие для запуска правила. Работает по принципу родительских парсеров, однако дочернее правило, при необходимости, может быть родительским. Пример: <rule id="100110" level="5"> <if_sid>100100, 100101</if_sid> <match>Error</match> <description>There is an error in the log.</description> </rule>
if_group	По умолчанию значение: n/a Любое имя группы	Необходимое условие для запуска правила. Соответствует, если группа уже совпадала ранее. Пример, если группа «sysmon_event1» уже совпадала и значение поля «"sysmon.image"» соответствует «lsm.exe», сработает правило и присвоит уровень 12: <rule id="184676" level="12"> <if_group>sysmon_event1</if_group> <field name="sysmon.image">lsm.exe</field> <description>Sysmon - Suspicious Process - lsm.exe</description> <group>pci_dss_10.6.1,pci_dss_11.4,gdpr_IV_35.7.d,hipaa_164.312.b,nist_800_53_AU.6,nist_800_53_SI.4,</group> </rule>

Атрибут	Возможные значения	Описание
if_level	По умолчанию значение: n/a Любой уровень от 1 до 16.	Если уровень уже был активирован другим правилом, то будет соответствовать.
if_matched_sid	По умолчанию значение: n/a Любой идентификатор правила. Не используется для уровня критичности 0.	Аналогично атрибуту «if_sid», однако заданное количество раз в течении определенного времени. Используется вместе с «frequency» и «timeframe». Пример, если правило «30315» срабатывает 10 раз за 120 секунд и запросы были сделаны одним и тем же пользователем, сработает правило «30316» и присвоит уровень 10: <rule id="30316" level="10" frequency="10" timeframe="120"> <if_matched_sid>30315</if_matched_sid> <description>Apache: Multiple Invalid URI requests from source.</description> <mitre> <id>T1499</id> </mitre> <group>gdpr_IV_35.7.d,hipaa_164.312.b,invalid_request,nist_800_53_AU.14,nist_800_53_AC.7,nist_800_53_SI.4,pci_dss_10.2.4,pci_dss_11.4,tsc_CC6.1,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group> </rule>

Атрибут	Возможные значения	Описание
if_matched_group	По умолчанию значение: n/a Любое имя группы	Аналогично атрибуту «if_group», однако заданное количество раз в течении определенного времени. Пример, если группа «virus» была сопоставлена 8 раз за последние 360 секунд, сработает правило и присвоит уровень 12. <rule id="40113" level="12" frequency="8" timeframe="360"> <if_matched_group>virus</if_matched_group> <description>Multiple viruses detected - Possible outbreak.</description> <group>virus,pci_dss_5.1,pci_dss_5.2,pci_dss_11.4,gpg13_4.2,gdpr_IV_35.7.d,nist_800_53_SI.3,nist_800_53_SI.4,</group> </rule>
info	По умолчанию значение: n/a Любое значение типа String	Дополнительная информация (текст, ссылка, CVE код). Пример: <rule id="5714" level="14" timeframe="120" frequency="3"> <if_matched_sid>5713</if_matched_sid> <match>Local: crc32 compensation attack</match> <description>sshd: SSH CRC-32 Compensation attack</description> <info type="cve">2001-0144</info> <info type="link">http://www.securityfocus.com/bid/2347/info/</info> <group>exploit_attempt,pci_dss_11.4,pci_dss_6.2,gpg13_4.12,gdpr_IV_35.7.d,nist_800_53_SI.4,nist_800_53_SI.2,</group> </rule>

Атрибут	Возможные значения	Описание
list	По умолчанию значение: n/a Путь для табличных списков (см. пункт 3.4).	Поиск в табличных списках. Пример, если в табличном списке найдено соответствие «audit.key» равное «Write» сработает правило и присвоит уровень 3: <rule id="80780" level="3"> <if_sid>80700</if_sid> <list field="audit.key" lookup = "match_key_value" check_value = "write">etc/lists/audit-keys</list> <description>Audit: Watch - Write access</description> <group>audit_watch_write,gdpr_IV_30.1.g,</group> </rule
options	Дополнительные параметры правила	1. alert_by_email – Всегда предупреждайте по электронной почте. 2. no_email_alert – Никогда не предупреждайте по электронной почте. 3. no_log – Не регистрируйте это событие. 4. no_full_log – Не включайте это full_log в событие. Используйте только один параметр. Пример: <rule id="9800" level="8"> <match>illegal user|invalid user</match> <description>sshd: Attempt to login using a non-existent user</description> <options>no_log</options> </rule>

Атрибут	Возможные значения	Описание

description	По умолчанию значение: n/a Любое значение типа String	Текстовое описание правила для уточнения, что правило проверяет и какие действия выполняет. Рекомендуется использовать при составлении пользовательских правил. Можно добавлять любое динамическое или статическое поле (см пункт 3.1.1). Пример: <rule id="100005" level="8"> <match>illegal user|invalid user</match> <description>sshd: Attempt to login using a non-existent user from IP $(attempt_ip)</description> <options>no_log</options> </rule>
mitre	Идентификатор тактики MITRE ATT&CK (см. Документация MITRE ATT&CK)	Указывает идентификатор или идентификаторы метода MITRE ATT&CK, которые соответствуют правилу. Пример: <rule id="100002" level="10"> <description>Attack technique sample.</description> <mitre> <id>T1110</id> <id>T1037</id> </mitre> </rule>
var	Название переменных. Часто используемая: «BAD_WORDS»	Определяет переменную, которую можно использовать в любом месте одного файла. Он должен быть определен на базовом уровне набора правил. Пример: <var name="joe_folder">/home/joe/</var> <group name="local,"> <rule id="100001" level="5"> <if_sid>550</if_sid> <field name="file">^$joe_folder</field> <description>A Joe's file was modified.</description> <group>ossec,pci_dss_10.6.1,gpg13_10.1,gdpr_IV_35.7.d,</group> </rule> </group>

		Пример: <var name= "BAD_WORDS">error|warning|failure</var>

Таблица 15 – Возможные опции атрибута rule

Опция	Допустимые значения	Описание
level	от 0 до 16	Уровень критичности
id	Любое число от 1 до 999999	Идентификатор правила
maxsize	Любое число от 1 до 9999	Максимальный размер события
frequency	Любое число от 2 до 9999	Частота
timeframe	Любое число от 1 до 99999	Время в секундах
ignore	Любое число от 1 до 999999	Время в секундах, для игнорирования правила, после его срабатывания
overwrite	yes, no	Указывает, нужно ли перезаписывать существующие правила или добавлять новые. Если «yes», то новое правило будет перезаписывать любые существующие правила с таким же именем. Если «no», то новое правило будет добавлено к уже существующим правилам, не затрагивая их.
noalert	0, 1	По умолчанию используется 0. Используется для того, чтобы не родительское правило генерировало предупреждение, а дочерние

Таблица 16 – Возможные параметры атрибута location

Компонент	Локация

Windows Eventchannel	EventChannel
Windows Eventlog	WinEvtLog
FIM (Syscheck)	syscheck
Rootcheck	rootcheck
Syscollector	syscollector
Vuln Detector	vulnerability-detector
Azure Logs	azure-logs
AWS S3 integration	aws-s3
Osquery integration	osquery
OpenSCAP integration	open-scap
CIS-CAT integration	wodle_cis-cat
SCA module	sca

3.3.2 Пример написания правил корреляции

Например, события поступили в следующем виде:

1. Vnimanie: computer1 - dostup k vredonosnomy website - https://plohoywebsite.com
2. Vnimanie: computer2 - dostup k vredonosnomy website - https://ochenplohoywebsite.com
3. Vnimanie: computer3 - dostup k vredonosnomy website - https://neochenplohoywebsite.com

Напишем для примера такие правила:

```
<group name="test_rules">
<rule id="100022" level="5">
<decoded_as>parent_test</decoded_as>
<field name="website">https://neochenplohoywebsite.com</field>
<description>Test successful. $(computer_name) obratilsya k
$(website)!</description>
```

```
</rule>
</group>

<group name="test_rules">
<rule id="100023" level="10">
<decoded_as>parent_test</decoded_as>
<field name="website">https://plohoywebsite.com</field>
<description>Test successful. $(computer_name) obratilsya k opasnomy
$(website)!</description>
</rule>
</group>

<group name="test_rules">
<rule id="100024" level="12">
<decoded_as>parent_test</decoded_as>
<field name="website">https://ochenplohoywebsite.com</field>
<description>Test successful. $(computer_name) obratilsya k ochen opasnomy
$(website)!</description>
</rule>
</group>
```

Атрибуты, используемые в составленных правилах:

1. Атрибут «**group**» используется для классификации;
2. Обязательный атрибут «**rule** id» и его опции;
3. Атрибут «**decoded_as**» указывает анализировать события, которые были обработаны определенным парсером;
4. Атрибут «**field**» используется как необходимое условие для запуска правила. Он проверит совпадение содержимого поля, извлеченного парсером;
5. Атрибут «**description**» содержит описание, поясняющее цель создания правила.

В данных парсерах для примера различается уровень угрозы и описание в зависимости от того, какое значение атрибута «"website"» было передано после процесса парсинга. В зависимости от переданных атрибутов выполнялись разные правила корреляции и присваивались различные уровни.

Результат:

**Messages:

WARNING: (7003): '75d568d7' token expires
INFO: (7202): Session initialized with token '51ee9aa0'

**Phase 1: Completed pre-decoding.
full event: 'Vnimanie: computer1 - dostup k vredonosnomy website - https://plohoywebsite.com'

**Phase 2: Completed decoding.
name: 'parent_test'
computer_name: 'computer1'
website: 'https://plohoywebsite.com'

**Phase 3: Completed filtering (rules).
id: '100023'
level: '10'
description: 'Test successful. computer1 obratilsya k opasnomy https://plohoywebsite.com !'
groups: '["test_rules"]'
firedtimes: '1'
mail: 'false'
**Alert to be generated.

**Phase 1: Completed pre-decoding.
full event: 'Vnimanie: computer2 - dostup k vredonosnomy website - https://ochenplohoywebsite.com'

**Phase 2: Completed decoding.
name: 'parent_test'
computer_name: 'computer2'
website: 'https://ochenplohoywebsite.com'

**Phase 3: Completed filtering (rules).
id: '100024'
level: '12'
description: 'Test successful. computer2 obratilsya k ochen opasnomy https://ochenplohoywebsite.com !'
groups: '["test_rules"]'

firedtimes: '1'
mail: 'true'
**Alert to be generated.

**Phase 1: Completed pre-decoding.
full event: 'Vnimanie: computer3 - dostup k vredonosnomy website - https://neochenplohoywebsite.com'

**Phase 2: Completed decoding.
name: 'parent_test'
computer_name: 'computer3'
website: 'https://neochenplohoywebsite.com'

**Phase 3: Completed filtering (rules).
id: '100022'
level: '5'
description: 'Test successful. computer3 obratilsya k https://neochenplohoywebsite.com !'
groups: '["test_rules"]'
firedtimes: '1'
mail: 'false'
**Alert to be generated.

3.3.3 Пример написания правил агрегации

Для созданий правил агрегации необходимо использовать синтаксис процесса классификации (см. пункт 3.3.1).

То правило, которое нужно агрегировать, необходимо переопределить в файле local_rules.xml. Для этого достаточно скопировать правило агрегации из файла системных правил. Например, правило с «id="5760"»:

```
<rule id="5760" level="5">
  <if_sid>5700,5716</if_sid>
  <match>Failed password|Failed keyboard|authentication error</match>
  <description>sshd: authentication failed.</description>
  <mitre>
    <id>T1110.001</id>
    <id>T1021.004</id>
```

```
  </mitre>
  <group>authentication_failed,</group>
</rule>
```

Далее необходимо изменить текст правила и добавить в файл local_rules.xml, новое правило. Для этого следует заменить «level = 1 и добавить опцию «overwrite="yes"» в атрибуте «**rule**» (табл. 16) для переопределения. Все остальные поля можно удалить, кроме «**description**», которое является обязательным:

```
<rule id="5760" level="1" overwrite="yes">
  <description>sshd: authentication failed.</description>
</rule>
```

Затем в local_rules нужно добавить правило агрегации:

```
<rule id="100002" level="5" frequency="6" timeframe="600">
   <if_matched_sid>5760</if_matched_sid>
   <description>Authentification failed 6 times from srcip</description>
 </rule>
```

Где:

1. Атрибут «frequency» показывает количество срабатываний агрегируемого правила;
2. Атрибут «timeframe» показывает временной промежуток (в секундах), за который агрегируемое правило должно сработать «frequency» раз, для того чтобы запустилось правило агрегации;
3. Для указания id агрегируемого правила используем «**if_matched_sid**»;
4. Для описания используется поле «**description**».

3.3.4 Пример связанных правил корреляции и агрегации

Предлагается рассмотреть пример связанных правил классификации, которые выявляют успешную попытку аутентификации после множества неуспешных попыток аутентификации (Bruteforce). Правило будет рассматриваться на основе событий sshd.

Будет использоваться 3 группы правил:

1. Выявление множественных неуспешных попыток аутентификации «**group** *name*="sshd_test,"»;
2. Успешная попытка аутентификация «**group** *name*="pam_test,"»;
3. Успешный подбор пароля после множественных неудачных попыток входа «**group** *name*="logincrack_test,"» (при условии, что сработают правила из группы 1 и 2).

Текст правил:

```
<group name="sshd_test,">
  <rule id="111100" level="0" noalert="1">
    <decoded_as>sshd_test</decoded_as>
    <description>SSHD test.</description>
  </rule>

  <rule id="111116" level="1">
    <description>sshd: неудачная попытка входа0.</description>
    <if_sid>111100</if_sid>
    <match>^Failed|^error: PAM: Authentication</match>
  </rule>

  <rule id="111160" level="5">
    <description>sshd: неудачная попытка входа.</description>
    <group>authentication_failed_test,</group>
    <if_sid>111100,111116</if_sid>
    <mitre>
      <id>T1110.001</id>
      <id>T1021.004</id>
    </mitre>
  </rule>

  <rule id="111163" frequency="3" ignore="30" level="10" timeframe="30">
    <description>sshd: Множественные неудачные попытки входа.</description>
    <group>authentication_failures_test,</group>
    <if_matched_sid>111160</if_matched_sid>
    <mitre>
      <id>T1110</id>
    </mitre>
  </rule>
```

</group>

Правило с «*id*="111100"» должно срабатывать, если события от sshd, следовательно, в поле «**decoded_as**» указывается название парсера «sshd_test».

Правило с «*id*="111116"» ссылается на правило «*id*="111100"», следовательно, анализируются только события от sshd. Затем указывается регулярное выражение в параметре «**match**» для фильтрации событий на вхождение строк «*Failed*» или «*error*» для «*PAM: Authentication*».

Правило с «*id*="111160"» ссылается на правило «*id*="111100"» и «*id*="111116"», следовательно, когда проходят эти два правила, событию присваивается уровень 5 и другие параметры («**mitre**» и «**group**»).

Правило с «*id*="111163"» имеет атрибут «frequency», в котором указывается количество срабатываний правила для того, чтобы правило агрегации сработало. Идентификатор указывается в поле «**if_matched_sid**». В атрибуте «timeframe» указывается временной промежуток (в секундах), за который правило, указанное в поле «**if_matched_sid**», должно сработать «frequency» раз, чтобы запустилось правило агрегации. В атрибуте «*ignore*» указывается временной промежуток (в секундах), который должны пройти, прежде чем правило снова может сработать.

Событию присваивается уровень 10, а также создается инцидент со средним уровнем (рис.5).

```
<group name="pam_test,">
  <rule id="112200" level="0" noalert="1">
    <description>Grouping of the pam_unix rules_test.</description>
    <if_sid>5500</if_sid>
  </rule>

  <rule id="112201" level="3">
    <description>PAM: успешный вход.</description>
    <group>authentication_success_test,</group>
    <if_sid>112200</if_sid>
    <match>session opened for user</match>
    <mitre>
      <id>T1078</id>
    </mitre>
  </rule>
</group>
```

Правило с «*id*="112200"» ссылается на системное правило «*id*="5500"», которое указывается в поле «**if_sid**», и срабатывает, если события PAM.

Правило с «*id*="112201"» ссылается на правило «*id*="112200"», следовательно, анализируются только события PAM. Затем указывается в параметре «**match**» регулярное выражение фильтрации событий «*session opened for user*». Если событие прошло фильтрацию, присваивает уровень 3 и другие параметры («**mitre**» и «**group**»).

```
<group name="logincrack_test,">
  <rule id="123123" ignore="60" level="14" timeframe="60">
    <description>Успешный подбор пароля после множественных неудачных попыток входа.</description>
    <if_matched_sid>111163</if_matched_sid>
    <if_sid>112201</if_sid>
    <mitre>
      <id>T1078</id>
      <id>T1110</id>
    </mitre>
  </rule>
</group>
```

Правило с «*id*="123123"» ссылается на правило, указанное в поле «**if_sid**» и сработает, если за 60 секунд правило указанное в поле «**if_matched_sid**» также сработало. Присваивается уровень 14 и другие параметры («**mitre**», «**description**»), а также создается инцидент с высоким уровнем (рис.5).

Результат выполнения:

Время	Источник	Локализация	Сырое событие
15.08.2024, 17:19:09	10.72.144.19	Успешный подбор пароля после множественных неудачных попыток входа.	Aug 15 17:19:09 netbox sshd[3995035]: pam_unix(sshd:sess
15.08.2024, 17:19:09	10.72.144.19	sshd: Множественные неудачные попытки входа.	Aug 15 17:19:09 netbox sshd[3995035]: Accepted password
15.08.2024, 17:18:58	10.72.144.19	sshd: неудачная попытка входа.	Aug 15 17:18:58 netbox sshd[3995033]: Failed password for
15.08.2024, 17:18:57	10.72.144.19	sshd: неудачная попытка входа.	Aug 15 17:18:57 netbox sshd[3995033]: Failed password for

Рисунок 4 –События, прошедшие этап анализа

ID	Критичность	Дата	Название	Статус
392	⌃⌃	16.08.2024, 09:03:10	Успешный подбор пароля после множественных неудачных попыток входа.	Новые
391	⌃	16.08.2024, 09:02:58	sshd: Множественные неудачные попытки входа.	Новые

Рисунок 5 – Инциденты, созданные системой после этапа анализа

3.4 Создание табличных списков

Табличные списки – это специальные базы данных, которые используются для хранения списков значений, например, IP-адресов и доменных имен. Табличные списки могут быть созданы для хранения списка разрешенных или запрещенных значений, а также в правилах процесса классификации.

Файл списка представляет собой обычный текстовый файл. Каждая строка имеет уникальный ключ и символ-разделитель – двоеточие. После разделителя можно добавить значение. Значения могут повторяться, но ключи должны быть уникальными. Пример:

key1:value1
key2:value2
key3:value2

Для того, чтобы включить символ двоеточие как часть ключа, например, в MAC-адресах, необходимо экранировать полный ключ, используя кавычки. Например:

"a0:a0:a0:a0:a0:a0":
"b1:b1:b1:b1:b1:b1":

С помощью ключа можно определить наличие или отсутствие поля в заданном списке. Добавляя значение, можно использовать его в качестве критерия в правилах. Например, есть имена учетных записей (ключи), связанные с именем отдела (значением). Можно создать правило, которое срабатывает, когда пользователь не из финансового отдела входит в систему на финансовом сервере (табл.17).

Таблица 17 – Параметры для создания табличного списка

Ключ	CIDR	Возможные совпадения
192.168.:	192.168.0.0/16	192.168.0.0 - 192.168.255.255
172.16.19.:	172.16.19.0/24	172.16.19.0 - 172.16.19.255
10.1.1.1:	10.1.1.1/32	10.1.1.1

Пример файла списка IP-адресов:

192.168.: Matches 192.168.0.0 - 192.168.255.255

172.16.19.: Matches 172.16.19.0 - 172.16.19.255
10.1.1.1: Matches 10.1.1.1

Пример использования табличных списков в правилах:

```
<rule id="110700" level="10">
  <if_group>123</if_group>
  <list field="srcip" lookup="address_match_key">etc/lists/List-one</list>
  <description>IP blacklisted in LIST ONE</description>
  <group>list1,</group>
</rule>

<rule id="110701" level="10">
  <if_group>123</if_group>
  <list field="srcip" lookup="address_match_key">etc/lists/List-two</list>
  <description>IP blacklisted in LIST TWO</description>
  <group>list2,</group>
</rule>

<rule id="110710" level="10">
  <if_sid>110700</if_sid>
  <list field="srcip" lookup="address_match_key">etc/lists/List-two</list>
  <description>IP blacklisted in LIST ONE and LIST TWO</description>
  <group>list1,list2,</group>
</rule>
```

В этих правилах просматривают, есть ли IP-адрес в списке 1 или 2, или в обоих.

Атрибут «lookup» может иметь несколько значений.

Приведенный ниже пример представляет собой поиск ключа в табличные списки на полное соответствие:

```
<list field="user" lookup="match_key">etc/lists/list-user</list>
```

Параметр «"match_key"» является параметром по умолчанию и может быть опущен:

```
<list field="user">etc/lists/list-user</list>
```

Если поле представляет собой IP-адрес, то использовать «"address_match_key"»:

<list field="srcip" lookup="address_match_key">etc/lists/list-IP</list>

Этот пример представляет собой поиск ключа в табличные списки на полное отсутствие его в списке:

<list field="user" lookup="not_match_key">etc/lists/list-user</list>

Если поле представляет собой IP-адрес, то использовать «"not_address_match_key"»

<list field="srcip" lookup="not_address_match_key">etc/lists/list-IP</list>

Приведенный ниже пример представляет собой поиск ключа в табличные списки на полное соответствие и при положительном совпадении возвращаемое значение ключа будет обработано с использованием регулярного выражения в атрибуте «check_value»:

<list field="user" lookup="match_key_value" check_value= "^block"> etc/lists/list-user</list>

Если поле представляет собой IP-адрес, то использовать «"address_match_key_value"»:

<list field="srcip" lookup="address_match_key_value" check_value= "^reject"> etc/lists/list-IP</list>

3.5 Создание правил обогащения

Обогащение событий – заполнение полей нормализованных, агрегированных и корреляционных событий согласно правилам обогащения. Поля заполняются данными, полученными из табличных списков.

Правила обогащения позволяют добавлять поля в события с дополнительной информацией. Например, в событии есть имя учётной записи пользователя. С помощью правила обогащения вы можете добавить сведения об отделе, должности и руководителе этого пользователя в поля события.

Таблица 18 – Параметры для создания правила обогащения

Атрибут	Описание
Поле события	Наименования поля события, по которому правило будет анализировать событие
Путь к табличному списку	Связь с существующим списком
Новое поле события	Наименования поля события, которое правило добавит в событие, т.е. его обогатит. Если наименование состоит из нескольких слов, их можно писать через символ точки или слитно. Пробелы и другие символы недопустимы.

Пример. Необходимо обогатить события, поступающие с агентов с операционной системой Linux, полем «agent.Enrich.Name». Правило обогащения выглядит так:

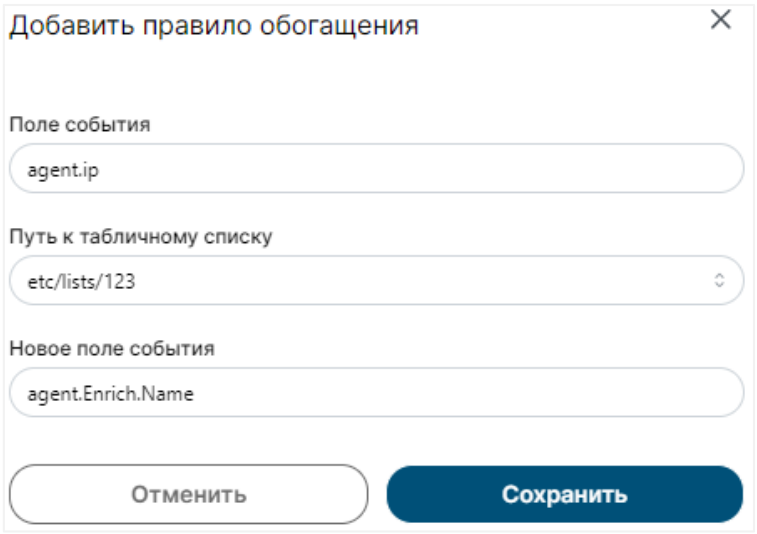

Рисунок 6 – Создание правила обогащения

В «Поле события» необходимо ввести поле события, по которому будет производится сравнение. В «Путь к табличному списку» выбираем из раскрывающегося перечня табличный список. Наименование нового поля следует прописать в «Новое поле события». В режиме редактирования правила обогащения «Поле события» является неизменным текстовым блоком.

Табличный список содержит следующую информацию, где «Ключ» содержит все IP-адреса агентов, установленных на Linux, а «Значение» – данные, которые будут содержать в поле «agentEnrichName»:

Таблица 19 – Табличный списокок «test123»

Ключ	Значение
10.72.112.01	linux-agent enrich
10.72.112.02	linux-agent enrich
10.72.112.03	linux-agent enrich

Далее, все события, поступающие с IP-адресов указанный в табличном списке test123, будут обогащены полем «agent.Enrich.Name», в котором будут данные, указанные в поле «Значение» в табличном списке test123.

ПРИЛОЖЕНИЕ. СХЕМА ПОЛЕЙ СОБЫТИЙ

Название поля	Тип данных	Описание
1. Параметры корреляции		
correlation.name	Текстовое поле	Название правила корреляции, с помощью которого выявлено событие.
correlation.count	Текстовое поле	Количество событий, прошедших процедуру агрегации.
rule.id	Текстовое поле	Идентификатор правил корреляции и агрегации. Значение может быть не уникальным.
rule.level	Текстовое поле	Уровень важности события
group	Текстовое поле	Категория правила корреляции
rule.mitre.id	Текстовое поле	Уникальный идентификатор техники Mittre Att&ck.
rule.mitre.tactic	Текстовое поле	Название тактики Mittre Att&ck.
rule.mitre.technique	Текстовое поле	Название техники Mittre Att&ck.
rule.groups	Текстовое поле	Группа или список групп, которым принадлежит правило.
2. Информационные поля		
rule.description	Текстовое поле	Текстовое описание правила.
3. Адресаты		
3.1. Отправитель		
src.fqdn	Текстовое поле	Полное доменное имя (FQDN) узла источника.
src.host	Текстовое поле	IP-адрес или название узла источника.
src.hostname	Текстовое поле	Название узла источника.
src.ip	Текстовое поле	IP-адрес (IPv4 или IPv6) узла источника.
src.mac	Текстовое поле	MAC-адрес узла источника.
src.port	Текстовое поле	Порт узла источника.
src.interface	Текстовое поле	Название интерфейса источника.
assigned_src_host	Текстовое поле	FQDN внешнего узла – источника.
assigned_src_ip	Текстовое поле	IP-адрес внешнего узла – источника.

Название поля	Тип данных	Описание
assigned_src_port	Текстовое поле	Порт внешнего узла – источника.
3.2. Получатель		
dst.fqdn	Текстовое поле	Полное доменное имя (FQDN) узла назначения.
dst.host	Текстовое поле	IP-адрес или название узла назначения.
dst.hostname	Текстовое поле	Имя узла назначения.
dst.ip	Текстовое поле	IP-адрес (IPv4 или IPv6) назначения.
dst.mac	Текстовое поле	MAC-адрес назначения.
dst.port	Текстовое поле	Порт назначения.
dst.interface	Текстовое поле	Название интерфейса назначения.
assigned_dst_host	Текстовое поле	FQDN внешнего узла – назначения.
assigned_dst_ip	Текстовое поле	IP-адрес внешнего узла – назначения.
assigned_dst_por	Текстовое поле	Порт внешнего узла – назначения.
4. Роли во взаимодействии		
4.1. Субъект		
subject	Текстовое поле	Субъект, производящий действие над объектом.
subject.account.dn	Текстовое поле	Уникальное в рамках Active Directory имя учетной записи (Distinguished Name) – субъекта, производящего действие над объектом.
subject.account.domain	Текстовое поле	Домен учетной записи – субъекта, производящего действие над объектом.
subject.account.fullname	Текстовое поле	Имя пользователя, указанное для учетной записи – субъекта, производящего действие над объектом.
subject.account.group	Текстовое поле	Перечень групп, в которых состоит учетная запись – субъект, производящий действие над объектом.

Название поля	Тип данных	Описание
subject.account.id	Текстовое поле	Идентификатор учетной записи – субъекта, производящего действие над объектом.
subject.account.name	Текстовое поле	Логин учетной записи – субъекта, производящего действие над объектом.
subject.account.privileges	Текстовое поле	Привилегии учетной записи – субъекта, производящего действие над объектом.
subject.account.provider	Текстовое поле	Хранилище учетной записи – субъекта, производящего действие над объектом.
subject.account.session_id	Текстовое поле	Идентификатор сессии учетной записи – субъекта, производящего действие над объектом.
subject.application.name	Текстовое поле	Название приложения – субъекта, производящего действие над объектом.
subject.application.account.domain	Текстовое поле	Домен учетной записи в приложении – субъекте, производящем действие над объектом.
subject.application.account.id	Текстовое поле	Идентификатор учетной записи в приложении – субъекте, производящем действие над объектом.
subject.application.account.name	Текстовое поле	Название приложения – субъекта, производящего действие над объектом.
subject.application.account.privileges	Текстовое поле	Права и привилегии доступа учетной записи в приложении – субъекте, производящем действие над объектом.
subject.application.account.session_id	Текстовое поле	Идентификатор сессии учетной записи в приложении – субъекте,

Название поля	Тип данных	Описание
		производящем действие над объектом.
subject.domain	Текстовое поле	Название домена субъекта, производящего действие над объектом.
subject.group	Текстовое поле	Группа, в которую входит субъект, производящий действие над объектом.
subject.id	Текстовое поле	Идентификатор субъекта, производящего действие над объектом.
subject.name	Текстовое поле	Название субъекта, производящего действие над объектом.
subject.privileges	Текстовое поле	Привилегии субъекта, производящего действие над объектом.
subject.process.chain	Текстовое поле	Цепочка порождения процессов для процесса, производящего действие над объектом.
subject.process.cmdline	Текстовое поле	Командная строка из исполняемого файла процесса, производящего действие над объектом.
subject.process.fullpath	Текстовое поле	Полный путь к исполняемому файлу процесса, производящему действие над объектом.
subject.process.guid	Текстовое поле	GUID процесса, производящего действие над объектом.
subject.process.hash	Текстовое поле	Хеш-суммы исполняемого файла процесса, производящего действие над объектом.
subject.process.id	Текстовое поле	Идентификатор процесса (PID), производящего действие над объектом.
subject.process.name	Текстовое поле	Имя исполняемого файла процесса, производящего действие над объектом.

Название поля	Тип данных	Описание
subject.process.original_name	Текстовое поле	Заданное разработчиком и взятое из метаданных имя исполняемого файла процесса, производящего действие над объектом.
subject.process.parent.cmdline	Текстовое поле	Командная строка из родительского исполняемого файла процесса, производящего действие над объектом.
subject.process.parent.fullpath	Текстовое поле	Полный путь к родительскому исполняемому файлу процесса, производящему действие над объектом.
subject.process.parent.guid	Текстовое поле	GUID родительского процесса, производящего действие над объектом.
subject.process.parent.hash	Текстовое поле	Хеш-суммы родительского исполняемого файла процесса, производящего действие над объектом.
subject.process.parent.id	Текстовое поле	Идентификатор родительского процесса (PID), производящего действие над объектом.
subject.process.parent.name	Текстовое поле	Имя родительского исполняемого файла процесса, производящего действие над объектом.
subject.process.parent.path	Текстовое поле	Путь к папке, в которой находится родительский исполняемый файл процесса, производящий действие над объектом.
subject.process.path	Текстовое поле	Путь к папке, в которой находится исполняемый файл процесса, производящий действие над объектом.
subject.process.version	Текстовое поле	Версия исполняемого файла процесса, производящего действие над объектом.

Название поля	Тип данных	Описание
subject.state	Текстовое поле	Состояние субъекта, производящего действие над объектом.
subject.type	Текстовое поле	Тип субъекта, производящего действие над объектом.
subject.version	Текстовое поле	Версия субъекта, производящего действие над объектом.
4.2. Объект		
object	Текстовое поле	Объект, над которым субъект производит действие.
object.account.dn	Текстовое поле	Уникальное в рамках Active Directory имя учетной записи (Distinguished Name) – объекта, над которым субъект производит действие.
object.account.domain	Текстовое поле	Домен учетной записи – объекта, над которым субъект производит действие.
object.account.fullname	Текстовое поле	Имя пользователя, указанное для учетной записи – объекта, над которым субъект производит действие.
object.account.group	Текстовое поле	Перечень групп, в которых состоит учетная запись – объект, над которым субъект производит действие.
object.account.id	Текстовое поле	Идентификатор учетной записи – объекта, над которым субъект производит действие.
object.account.name	Текстовое поле	Логин учетной записи – объекта, над которым субъект производит действие.
object.account.privileges	Текстовое поле	Привилегии учетной записи – объекта, над которым субъект производит действие.

Название поля	Тип данных	Описание
object.account.session_id	Текстовое поле	Идентификатор сессии учетной записи – объекта, над которым субъект производит действие.
object.application.name	Текстовое поле	Название приложения – объекта, над которым субъект производит действие.
object.application.account.domain	Текстовое поле	Домен учетной записи в приложении – объекте, над которым субъект производит действие.
object.application.account.id	Текстовое поле	Идентификатор учетной записи в приложении – объекте, над которым субъект производит действие.
object.application.account.name	Текстовое поле	Логин учетной записи в приложении – объекте, над которым субъект производит действие.
object.application.account.privileges	Текстовое поле	Права и привилегии доступа учетной записи в приложении – объекте, над которым субъект производит действие.
object.application.account.session_id	Текстовое поле	Идентификатор сессии учетной записи в приложении – объекте, над которым субъект производит действие.
object.domain	Текстовое поле	Название домена объекта, над которым субъект производит действие.
object.fullpath	Текстовое поле	Полный путь к объекту, над которым субъект производит действие.
object.group	Текстовое поле	Название группы, в которую входит объект, над которым субъект производит действие.

Название поля	Тип данных	Описание
object.hash	Текстовое поле	Значение хеш-суммы объекта, над которым субъект производит действие.
object.id	Текстовое поле	Идентификатор объекта, над которым субъект производит действие.
object.name	Текстовое поле	Название объекта, над которым субъект производит действие.
object.new_value	Текстовое поле	Конечное значение измененного свойства объекта.
object.num_value	Текстовое поле	Исходное значение измененного свойства объекта (численное значение).
object.path	Текстовое поле	Путь к объекту, над которым субъект производит действие.
object.process.chain	Текстовое поле	Цепочка порождения процессов для процесса, над которым субъект производит действие.
object.process.cmdline	Текстовое поле	Командная строка из исполняемого файла процесса, над которым субъект производит действие.
object.process.fullpath	Текстовое поле	Полный путь к исполняемому файлу процесса, над которым субъект производит действие.
object.process.guid	Текстовое поле	GUID процесса, над которым субъект производит действие.
object.process.hash	Текстовое поле	Хеш-суммы исполняемого файла процесса, над которым субъект производит действие. Суммы должны быть разделены пробелами.
object.process.id	Текстовое поле	Идентификатор процесса (PID), над которым субъект производит действие.

Название поля	Тип данных	Описание
object.process.name	Текстовое поле	Имя исполняемого файла процесса, над которым субъект производит действие.
object.process.original_name	Текстовое поле	Заданное разработчиком и взятое из метаданных имя исполняемого файла процесса, над которым субъект производит действие.
object.process.parent.cmdline	Текстовое поле	Командная строка из родительского исполняемого файла процесса, над которым субъект производит действие.
object.process.parent.fullpath	Текстовое поле	Полный путь к родительскому исполняемому файлу процесса, над которым субъект производит действие.
object.process.parent.guid	Текстовое поле	GUID родительского процесса, над которым субъект производит действие.
object.process.parent.hash	Текстовое поле	Хеш-суммы родительского исполняемого файла процесса, над которым субъект производит действие.
object.process.parent.id	Текстовое поле	Идентификатор родительского процесса (PID), над которым субъект производит действие.
object.process.parent.name	Текстовое поле	Имя родительского исполняемого файла процесса, над которым субъект производит действие.
object.process.parent.path	Текстовое поле	Путь к папке, где находится родительский исполняемый файл процесса, над которым субъект производит действие.
object.process.path	Текстовое поле	Путь к папке, где находится исполняемый файл процесса, над которым субъект производит действие.

Название поля	Тип данных	Описание
object.process.version	Текстовое поле	Версия исполняемого файла процесса, над которым субъект производит действие.
object.property	Текстовое поле	Изменяемое свойство объекта, над которым субъект производит действие.
object.query	Текстовое поле	Текст запроса объекта типа request, над которым субъект производит действие.
object.state	Текстовое поле	Состояние объекта, над которым субъект производит действие.
object.type	Текстовое поле	Тип объекта, над которым субъект производит действие.
object.value	Текстовое поле	Значение изменяемого свойства объекта, над которым субъект производит действие.
object.vendor	Текстовое поле	Производитель объекта, над которым субъект производит действие.
object.version	Текстовое поле	Версия объекта, над которым субъект производит действие.
5. Параметры взаимодействия		
action	Текстовое поле	Действие, производимое субъектом над объектом.
direction	Текстовое поле	Направление взаимодействия между сторонами.
duration	Текстовое поле	Продолжительность события в секундах.
importance (Критичность)	Текстовое поле	Степень важности события с точки зрения ИБ.
logon_auth_method	Текстовое поле	Способ входа в систему.
logon_id	Текстовое поле	Уникальный идентификатор сессии входа пользователя (LogonID) Используется для связывания всех событий, относящихся к одной сессии входа пользователя в

Название поля	Тип данных	Описание
		систему. Особенно полезен в связке с subject.account.name и srcuser.
logon_service	Текстовое поле	Название сервиса, через который выполнена аутентификация в системе, или название приложения или терминала, через который выполняется работа с операционной системой или приложением.
logon_type	Текстовое поле	Тип входа в систему.
protocol	Текстовое поле	Протокол передачи данных (ниже уровня приложения).
protocol.layer7	Текстовое поле	Протокол уровня приложения (по модели OSI).
reason	Текстовое поле	Причина, по которой произошло событие.
status	Текстовое поле	Результат выполнения действия.
6. Дополнительная информация		
datafield1	Текстовое поле	Значения, неподходящие под стандартные поля
datafield2	Текстовое поле	
datafield3	Текстовое поле	
datafield4	Текстовое поле	
datafield5	Текстовое поле	
datafield6	Текстовое поле	
datafield7	Текстовое поле	
datafield8	Текстовое поле	
datafield9	Текстовое поле	
datafield10	Текстовое поле	
datafield11	Текстовое поле	
datafield12	Текстовое поле	
datafield13	Текстовое поле	
datafield14	Текстовое поле	
datafield15	Текстовое поле	
datafield16	Текстовое поле	
datafield17	Текстовое поле	

Название поля	Тип данных	Описание
datafield18	Текстовое поле	
datafield19	Текстовое поле	
datafield20	Текстовое поле	
full_log	Текстовое поле	Сырое событие.
keywords	Текстовое поле	Служебные теги или флаги события
		Используются Windows или другими источниками логов для классификации события. Например, “Audit Success”, “Audit Failure”, “Critical”, “Info” и т.п. Часто имеют числовой вид, представляющий собой битовую маску.
msgid	Текстовое поле	Идентификатор типа события.
previous_log	Текстовое поле	Результат предыдущих операций.
previous_output	Текстовое поле	Логи предыдущих событий.
provider.id	Текстовое поле	Уникальный идентификатор источника события (GUID)
		Используется для точной идентификации службы или приложения, которое сгенерировало событие.
provider.name	Текстовое поле	Имя источника события
		Обычно это читаемое имя компонента Windows, например: Microsoft-Windows-Security-Auditing, Sysmon, Application Error
system_time	Текстовое поле	Время генерации события в формате UTC (из системного лога)
		Отличается от event.received или timestamp, потому что показывает точный момент возникновения события на источнике (машине), а не на стороне Wazuh.
task_id	Текстовое поле	Идентификатор задачи (Task ID)

Название поля	Тип данных	Описание
		Зависит от типа события. Например, в Sysmon или Security логе может использоваться для уточнения подкатегории события. Помогает различать похожие события внутри одного Event ID.
thread_id	Текстовое поле	Идентификатор потока (Thread ID)
		Указывает на поток операционной системы, в котором было сгенерировано событие. Может быть полезен для корреляции действий внутри одного процесса.
7. Источник события		
event_src.category	Текстовое поле	Категория источника события.
event_src.fqdn	Текстовое поле	Полное доменное имя (FQDN) узла – источника события.
event_src.host	Текстовое поле	Название или IP-адрес узла – источника события.
event_src.hostname	Текстовое поле	Название узла – источника события.
event_src.ip	Текстовое поле	IP-адрес (IPv4 или IPv6) узла – источника события.
event_src.rule	Текстовое поле	Название правила, службы или приложения, благодаря которым на источнике зарегистрировано событие.
event_src.subsys	Текстовое поле	Подсистема источника события, в которой генерируются события этого типа.
event_src.title	Текстовое поле	Название продукта – источника события.
event_src.vendor	Текстовое поле	Производитель источника события.
8. Точка сбора		
recv_host	Текстовое поле	Название узла, с которого получено событие.

Название поля	Тип данных	Описание
recv_ipv4	Текстовое поле	IPv4-адрес узла, от которого получено событие.
recv_ipv6	Текстовое поле	IPv6-адрес узла, от которого получено событие.
recv_time	Текстовое поле	Время получения события коллектором
9. Служебные данные		
collector.id	Текстовое поле	Идентификатор коллектора
decoder.ftscomment	Текстовое поле	Комментарий для событий First Time Seen (FTS), поясняющий контекст срабатывания.
decoder.name	Текстовое поле	Имя декодера для идентификации в системе.
decoder.parent	Текстовое поле	Идентификатор родительского декодера.
event.id	Текстовое поле	Идентификатор события.
host	Текстовое поле	Имя хоста.
predecoder.hostname	Текстовое поле	Имя хоста, извлечённое непосредственно из сырого лога на этапе предварительной обработки (pre-decoding).
predecoder.program_name	Текстовое поле	Название сервиса, сгенерировавшего лог на этапе предварительной обработки (pre-decoding).
time	Текстовое поле	Время события (UTC+0).
timestamp	Текстовое поле	Временная метка события на сервере.