# ТЕХНОЛОГИИ БОЛЬШИХ ДАННЫХ ДЛЯ КОРРЕЛЯЦИИ СОБЫТИЙ БЕЗОПАСНОСТИ НА ОСНОВЕ УЧЕТА ТИПОВ СВЯЗЕЙ

***Котенко И.В.[1], Федорченко А.В.[2], Саенко И.Б.[3], Кушнеревич А.Г.[4]***

*Работа посвящена исследованию подхода к параллельной обработке данных для выполнения задач корреляции событий безопасности на основе технологий больших данных. Рассматриваются различные методы и задачи корреляции событий безопасности, а также технологии обработки больших данных, применимые для задач мониторинга безопасности. Основное внимание уделяется задачам выявления связей между типами событий безопасности и оценки зависимости силы связей от распределения событий во времени. Описывается реализация задач корреляции на платформе Spark, и приводятся результаты экспериментальной оценки показателей процессов корреляции событий безопасности.[5]*

***Ключевые слова**: корреляция событий, инциденты безопасности, коэффициент корреляции Пирсона, SIEM–системы, большие данные, параллельная обработка данных*

**DOI**: 10.21681/2311-3456-2017-5-2-16

**Введение**

В настоящее время средства обнаружения, предупреждения и предотвращения компьютерных атак и вредоносной активности, а также мониторинга и управления безопасностью представлены различными классами решений. Одним из таких классов являются системы SIEM (Security Information and Event Management) [1-3]. Основными задачами SIEM-систем являются сбор больших массивов гетерогенных данных о событиях безопасности и обнаружение инцидентов и угроз безопасности в результате их обработки [4,5]. При этом одной из достаточной острых проблем, стоящих перед современными SIEM-системами, является проблема обработки больших данных, которая вызвана необходимостью обработки огромных массивов разнородных данных о событиях безопасности (логов), поступающих в SIEM-систему от различных источников. В качестве источников больших данных выступают операционные системы, системы управления базами данных, антивирусные средства, сетевые элементы, системы обнаружения атак и т.д. Современная компьютерная сеть, безопасность которой управляется SIEM-системой, может содержать несколько сотен / тысяч таких источников данных. В результате в SIEM-систему за день могут поступать на обработку данные о десятках / сотнях миллионов событий безопасности.

Обработка этих данных включает такие операции, как фильтрация, агрегация, приоритезация, корреляция и другие. Наибольшей вычислительной сложностью обладает операция корреляции событий безопасности. Суть этой операции заключается в определении причинно-следственных связей между поступающими на обработку событиями. Это позволяет выявлять вредоносную и аномальную активности, определять источник и цель атаки, обнаруживать многошаговые атаки, делать выводы об инцидентах безопасности и вырабатывать эффективные контрмеры [6,7]. При этом следует отметить, что все эти действия должны выполняться в реальном или близком к реальному времени, чтобы не дать возможности злоумышленнику реализовать свои вредоносные цели.

Однако проблема выполнения операции корреляции событий безопасности в SIEM системах в реальном или близком к реальном времени остается нерешенной проблемой. Ее решение возможно только с использованием современных подходов,

1 Котенко Игорь Витальевич, доктор технических наук, профессор, Санкт-Петербургский институт информатики и автоматизации Российской академии наук, Университет ИТМО, Санкт-Петербург, Россия. E-mail: ivkote@comsec.spb.ru

2 Федорченко Андрей Владимирович, мл. научный сотрудник, Санкт-Петербургский институт информатики и автоматизации Российской академии наук, Университет ИТМО, Санкт-Петербург, Россия. E-mail: fedorchenko@comsec.spb.ru

3 Саенко Игорь Борисович, доктор технических наук, профессор, ведущий научный сотрудник, Санкт-Петербургский институт информатики и автоматизации Российской академии наук, Университет ИТМО, Санкт-Петербург, Россия. E-mail: ibsaen@comsec.spb.ru

4 Кушнеревич Алексей Геннадьевич, младший научный сотрудник, Санкт-Петербургский институт информатики и автоматизации Российской академии наук, Университет ИТМО, Санкт-Петербург, Россия. E-mail: kushnerevich@comsec.spb.ru

5 Работа выполнена при поддержке гранта РНФ №15-11-30029.

средств и методов обработки Больших Данных. Одним из таких подходов является использование современных программных средств реализации параллельных потоковых вычислений.

Известно несколько средств реализации параллельных вычислений. К числу наиболее распространенных относятся Hadoop и Spark. При этом Spark на больших объемах входных данных показывает более высокую производительность. В то же время Spark является более молодым средством. Работы, в которых Spark применяется для обработки событий безопасности, только начинают появляться. Это позволило сформулировать основные цели работы, которыми являются (1) формализация задач выявления связей между типами событий безопасности и оценки зависимости силы связей от распределения событий во времени, решаемых в SIEM-системах при корреляции событий безопасности, (2) реализация алгоритмов решения этих задач в Spark и (3) экспериментальная оценка полученных решений.

Новизна работы заключается в формальной спецификации и реализации в среде параллельных вычислений нового подхода к выявлению косвенных связей между типами событий безопасности и оценке зависимости силы связи по отношению к времени, основанной на использовании коэффициентов корреляции Пирсона. Теоретический вклад работы состоит в развитии методов корреляции, которые позволяют выявлять однотипные и разнотипные связи между событиями безопасности, а также развитии технологии обработки Больших данных на примере корреляции больших массивов разнородных данных о событиях безопасности в SIEM системах.

**Постановка задачи и релевантные работы**

О процессе корреляции данных о событиях безопасности принято говорить в широком и узком смысле слова. В широком смысле слова под корреляцией данных понимается вся предварительная обработка данных, собираемых SIEM системой от источников. Результаты этой обработки помещаются в хранилище данных SIEM системы и используются в дальнейшем для проведения более детального и тщательного анализа информации о безопасности. Обычно в рамках корреляции данных о событиях безопасности, понимаемой в широком смысле слова, выделяются следующие этапы (рис. 1) [6,8,9]:

- нормализация (приведение собираемых данных к единому формату);

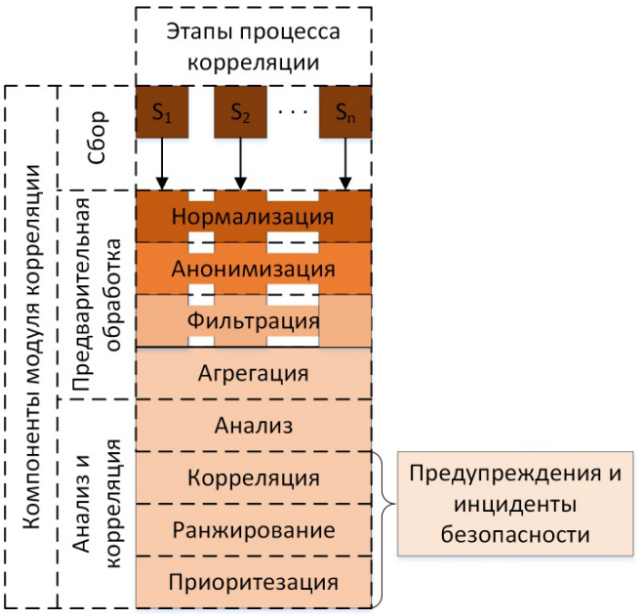


***Рис. 1.*** *Схема процесса корреляции данных в SIEM-системе*

- анонимизация (преобразование данных с целью исключения их нежелательного разглашения);
- фильтрация (отсеивание дублируемых, малозначительных или бесполезных данных);
- агрегация (получение новых данных с использованием различных функций агрегирования, таких как average(), count(), min(), max() and so on);
- анализ (нахождение закономерностей появления и зависимостей, в том числе скрытых, между событиями безопасности);
- собственно корреляция (определение взаимосвязей между экземплярами событий и их групп);
- ранжирование (оценка результатов корреляции по определенным признакам);
- приоритезация (вычисление степени важности результатов процесса корреляции).

В последовательности этапов, представленной на рис. 1, корреляция понимается в узком смысле слова. Ее суть заключается в нахождении причинно-следственных связей между анализируемыми событиями безопасности. Эта операция по своей вычислительной сложности является наиболее сложной из всех этапов, перечисленных на рис.1. Реализация этого этапа с помощью параллельных вычислений представляет несомненный интерес с научной и практической точек зрения. Поэтому в дальнейшем в работе говоря о корреляции событий безопасности, будем рассматривать определение этой операции в узком смысле слова.

Таким образом, постановка задачи, решаемой в работе, заключается в следующем. Входными

(исходными) данными является поток данных о событиях безопасности, которые относятся к различным типам. Необходимо разработать метод обработки входного потока данных, позволяющий выявлять прямые и косвенные связи между экземплярами событий различных типов и допускающий свое распараллеливание в среде Spark.

Различные методы корреляции данных были впервые использованы в Intrusion Detection Systems (IDSs) для обнаружения связей между сетевыми событиями с целью их последующей агрегации и выявления атак, в том числе распределенных и многошаговых [3]. Методы корреляции данных о событиях безопасности, реализованные в IDSs, были в дальнейшем внедрены и адаптированы в SIEM-системах для обработки данных о событиях безопасности. В настоящее время процесс корреляции данных в различных SIEM-системах отличается большим многообразием реализованных в них методов и подходов [10,11]. В общем случае следует считать, что для корреляции возможно использование комбинации различных методов, каждый из которых обладает своими достоинствами и недостатками.

Все методы корреляции данных можно условно разделить на сигнатурные и эвристические. Данные методы могут применять различные подходы, основанные на анализе схожести, статистическом анализе, интеллектуальном анализе данных и прочие. Анализ применяемых методов корреляции в существующих решениях значительно усложняется ввиду отсутствия работ с детальным описанием используемых методов корреляции.

В настоящее время наиболее распространенным и простым в реализации, однако достаточно сложным в настройке и не приспособленным к автоматизированной адаптации, является правило-ориентированный метод [12-14]. Основным принципом настройки SIEM-систем, использующих данный метод, является составление правил корреляции в зависимости от характеристик анализируемой инфраструктуры. Работа модуля корреляции на основе правило-ориентированного метода базируется на фиксированном соотнесении событий друг с другом при выполнении определенных условий. Данные условия могут содержать логические операции над данными, их свойствами и вычисляемыми показателями. Главным недостатком данного метода является сложность процесса составления правил. Другим недостатком является тот факт, что качество выполнения корреляции правило-ориентированным методом напрямую зависит от квалификации администратора безопасности (проектировщика).

Другие методы корреляции, такие как шаблонно-ориентированный (сценарно-ориентированный) [12], граф-ориентированный [15,16], основанный на машине конечных состояний [15,17], основанный на анализе схожести [18,19] и другие, по своей сути имеют различные модели представления событий безопасности и их связей. Однако, в конечном итоге, они также могут быть выражены в виде правил, поскольку все они являются сигнатурными методами.

Достаточно интересным современным направлением развития методов корреляции событий является применение подходов, базирующихся на машинном обучении и интеллектуальном анализе данных, таких как байесовские сети [12,15,20], иммунные сети [15,20], искусственные нейронные сети [15,20-22] и другие. Достоинство данных подходов заключается в возможности самостоятельной (безусловной) корреляции событий с минимизацией ручной настройки. Однако для построения моделей обучения требуется предварительный анализ самих данных, который далеко не всегда можно автоматизировать. Кроме того, применение интеллектуальных подходов накладывает дополнительные требования по оценке адекватности и качества моделей, а исходные данные обучения должны быть репрезентативными.

Применение параметрических и непараметрических показателей корреляции (линейных, различных ранговых и других), в том числе коэффициентов Пирсона, описывается в [23-25] для решения задач оценки алгоритмов, выявления образов распределенных DoS-атак, выделения подмножеств признаков данных, по которым производится обнаружение вторжений.

В [26] рассматриваются детерминированный и стохастический подходы корреляции событий для задачи обнаружения сетевых атак. Предлагается вероятностная модель процесса корреляции событий, которая основана на их пространственно-временном анализе. В рамках представленной модели, пространства событий связываются в цепи последовательностей, а каждому пространству в текущий момент соответствует конкретное состояние из множества состояний. Полученные цепи последовательностей используются для вычисления вероятности выполнения определенного сценария атаки.

В [27] рассматривается процесс корреляции для обнаружения атак в облачной вычислительной среде, как системе распределенных сенсоров.

Для выполнения подхода предлагается использовать технологию CEP (Complex Event Processing). Предлагаемая онтологическая модель обнаружения атак включает сценарии, индикаторы, симптомы и воздействия атак, а также состояние предполагаемой цели.

В [28] предлагается использование модели поведения приложений для выявления неправомерной и аномальной активности. Исходными данными для построения модели являются события, отражающие системные вызовы всевозможных приложений. Выделено 5 уровней представления модели, верхним уровнем является выделенная функциональность. На основе исходной информации формируется профиль нормального поведения за счет обработки мульти-графа, в котором каждой вершиной является событие.

Для определения возможности применять современные средства параллельной обработки в процессе корреляции данных о событиях безопасности были проанализированы работы, посвященные применению технологии обработки Big Data для выполнения различных задач обеспечения безопасности. Пример Stream Processing Engine для обработки событий представлен в [29]. Предлагаемый подход позволяет обеспечить гибкое управление ресурсами и балансировку нагрузки с низкими накладными расходами. В [30] описываются результаты исследования различных средств обработки больших данных из состава фрэймворков Hadoop и Spark, которые применялись для решения задач анализа сетевого трафика с целью выявления аномальной активности. В этой работе для проведения экспериментов был использован вычислительный кластер, а задачи анализа трафика были реализованы за счет самообучающихся алгоритмов. Результаты проведенных исследований показали явное преимущество технологии многопоточной обработки данных Spark над другими сравниваемыми технологиями работы с большими данными.

[31] посвящена сравнению технологий обработки больших данных для анализа сетевого трафика. Результатом этой работы является выявление преимущества в точности классификации трафика с помощью алгоритма случайного леса деревьев решений (Random Forest) над алгоритмом наивного Байеса (Naïve Bayes). Также, как и в [30], указывается вычислительное преимущество технологии Spark.

В [32] алгоритмы интеллектуального анализа больших данных для выполнения задач мониторинга безопасности были реализованы в библиотеке MLib из состава фрэймворка многопоточной обработки информации Spark.

Анализ релевантных работ показывает, что в настоящее время разработано достаточно большое количество методов корреляции данных, которые обладают различными достоинствами и недостатками. Ряд предлагаемых подходов базируется на использовании сценариев атак для формирования последовательностей атакующих действий, что относит их к классу сигнатурных методов и предполагает значительные временные затраты для настройки и адаптации к целевой инфраструктуре. Важным направлением дальнейшего совершенствования методов корреляции является их адаптация к технологиям обработки Big Data и параллельных вычислений. Среди современных средств, реализующих параллельные вычисления, одним из наиболее предпочтительных является фреймворк Spark. В представленной работе акцент делается на выявлении связей между типами событий и оценке зависимости силы связей от распределения событий во времени. Одним из подходов к оценке указанной зависимости является применение двумерного линейного коэффициента Пирсона, поскольку данный показатель требует наименьших затрат вычислительных ресурсов.

## Математические основы процесса корреляции событий безопасности

### *A. Основные задачи процесса корреляции*

Для исследования возможности и оценки результативности параллельной обработки данных в процессе корреляции событий безопасности необходимо в первую очередь выделить основные задачи корреляции. К числу общих задач процесса корреляции относятся:

- преобразование низкоуровневых событий в высокоуровневые события с помощью агрегирования;
- определение связей между разноуровневыми событиями и информацией безопасности;
- приоритезация событий вычисление важности событий и их групп в рамках задачи обеспечения безопасности;
- обнаружение инцидентов и формирование предупреждений безопасности.

Первоначальным этапом корреляции событий безопасности является сбор разнородных (сырых) данных из множества разнородных источников. Заключительным этапом корреляции является определение текущего состоянии защищенности анализируемой сетевой инфраструктуры.

В ходе исследования различных аспектов параллельной обработки событий безопасности следует учитывать, что процесс корреляции является непрерывным. Кроме того, он должен выполняться в реальном масштабе времени. Также стоит отметить, что основной причиной применения параллельных вычислений в процессе корреляции событий безопасности является условно неограниченное количество источников. Следовательно, необходимо полагать, что количество исходных данных для обработки также является неограниченным. Выполнение данных условий возможно осуществлять с помощью применения технологий обработки Big Data и средств, реализующих параллельную обработку данных (например, Hadoop или Spark).

Глобальная цель нашего исследования заключается в разработке метода автоматизированной корреляции разнородной информации безопасности, основанной на параллельных вычислениях. Для достижения данной цели предлагается использовать результаты структурного, функционального, поведенческого и эволюционного анализа защищаемых инфраструктур. Выделение указанных выше задач способствует достижению глобальной цели и обусловлено соответствующими аспектами сложности анализируемых компьютерных инфраструктур как сложных динамических систем [33].

В рамках разработки метода автоматизированной параллельной корреляции больших данных о событиях безопасности в работе исследовалось решение следующих подзадач задач:

- определение косвенных однотипных связей между типами событий;
- вычисление частных и общих коэффициентов корреляции между типами событий;
- определение временного окна (репрезентативной выборки) событий для анализа.

Ниже приводится описание порядка решения каждой из выше упомянутых задач для исследования возможности их эффективной параллелелизации.

***Б. Определение косвенных однотипных связей***

В предыдущем исследовании авторов [34] в качестве основы механизма поиска зависимостей в исходных данных для применения различных статистических методов корреляции был выбран граф связей типов событий. При данном подходе связи между типами событий предлагается разделять на следующие классы:

- прямые связи;
- косвенные однотипные и разнотипные связи.

Выделение прямых связей осуществляется на основе анализа структур типов событий по их равнозначным (одинаковым) свойствам. Например, для ОС Windows при сравнении типов событий «Завершение процесса» (4689) и «Вызвана привилегированная служба» (4673), равнозначным свойством является «ProcessId» (процесс-инициатор).

Косвенные однотипные связи соединяют события, свойства которых эквивалентны по типу содержимого, косвенные разнотипные связи связывают события, свойства которых эквивалентны по значениям, но имеют различные типы содержимого. Например, событие типа «Запуск процесса» (4688) помимо свойства «NewProcessID» содержит свойство «CreatorProcessID». Оба свойства инициализируют идентификатор процесса, только в первом случае – наследника, а во втором – предка. Данная связь является косвенной однотипной по типу содержимого. Событие типа «Запуск процесса» содержит также свойство «ProcessName», обозначающее «Имя исполняемого модуля в файловой системе». «NewProcessID» и «ProcessName» это свойства разного типа, и они образуют между собой косвенную разнотипную связь. Нахождение косвенных связей делается на основе содержательного анализа свойств типов событий.

Анализ структур типов событий является последовательно выполняемой операцией, поскольку прямые связи определяются при поступлении только первого экземпляра события неизвестного типа, при условии, что формат типа является нормализованным.

Определение косвенных связей условно является однократно выполняемой операцией, но продолжительной во времени с точки зрения анализа исходных данных и принятия решения о наличии связи между типами событий. Иными словами, прямые связи определяются незамедлительно при поступлении события нового типа, а выявление косвенных связей требует предварительного анализа достаточно большого количества данных с накоплением полученных результатов.

Определение косвенной однотипной связи $R^{indir.same}$ для любых двух типов $t_1$ и $t_2$ производится путем оценки результата пересечения множеств значений их свойств:

$$R^{indir.same} = Evaluate\,(V\,(p_1) \cap V\,(p_2)), \quad (1)$$

где $p_1$ и $p_2$ являются неравнозначными свойствами, которые характеризуют типы $t_1$ и $t_2$, а $V\,(p_1)$ и $V\,(p_2)$ являются, соответственно, множествами их значений.

При этом предполагается, что оценка пересечения свойств и непосредственное принятие решения о наличии косвенных связей между рассматриваемыми типами может производиться путем вычисления отношения количества идентичных значений к общему количеству значений (уникальных в рамках сравнения) двух множеств. Это условие схематично иллюстрируется на рис. 2.

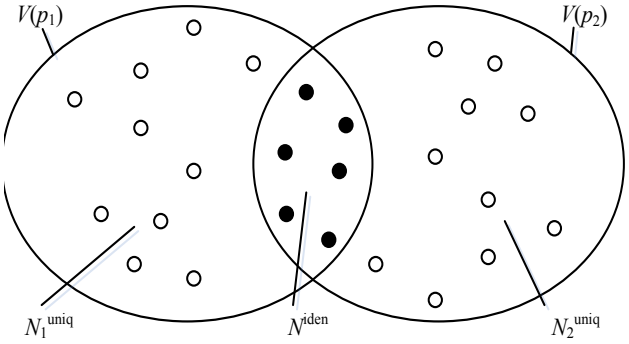


***Рис. 2.*** *Основные параметры для вычисления косвенных связей*

Положим, что $N^{iden}$ обозначает количество идентичных значений во множествах $V(p_1)$ и $V(p_2)$, $N_1^{uniq}$ количество уникальных значений во множестве $V(p_1)$, а $N_2^{uniq}$ – количество уникальных значений во множестве $V(p_2)$. Тогда в качестве функции $Evaluate\ (V(p_1) \cap V(p_2))$ можно использовать следующее выражение:

$$Evaluate\ (V(p_1) \cap V(p_2)) = \frac{N^{iden}}{N_1^{uniq} + N_2^{uniq} - N^{iden}} \quad (2)$$

При этом учитывается, что множества $V(p_1)$ и $V(p_2)$ не нулевые. Также необходимым требованием для использования выражения (2) является наличие по крайне мере одного отличающегося значения во множествах $V(p_1)$ и $V(p_2)$. В противном случае свойства $p_1$ и $p_2$ будут однозначно идентифицированы как однотипные неравнозначные.

При параллельном вычислении оценки наличия косвенной связи между типами событий следует учитывать, что изначально обработке подвергаются все значения всех свойств типов. Только после установления однотипного неравнозначного отношения между двумя рассматриваемыми свойствами производится модификация графа связей для тех типов, которые содержат в своей структуре данные свойства. Кроме того, необходимо отметить, что косвенная однотипная связь может быть установлена внутри одного типа событий ($t_1 = t_2$).

Определение косвенной разнотипной связи, по сути, представляет собой операцию группировки разнотипных неравнозначных свойств в результате их корреляции. В данном случае также может быть применен метод ранговой корреляции, когда для двух и более свойств производится ранжирование их значений, полученных из экземпляров событий безопасности. Однако в настоящей работе данная операция не рассматривается. Этот вопрос планируется рассмотреть в дальнейших исследованиях.

Полученные в результате выполнения выше описанных операций косвенные однотипные связи являются основой для уточнения графа отношений типов событий и дальнейшего вычисления коэффициентов корреляции между типами событий.

***B. Определение коэффициента корреляции между типами событий***

Привязка событий ко времени приводит к тому, что между двумя произошедшими событиями всегда имеется некоторый, как правило, ненулевой, временной интервал. В результате в качестве одного из основных признаков корреляции следует учитывать *временную задержку* между двумя рассматриваемыми событиями.

Обозначим временную задержку между событием $e_n$ и событием $e_i$ $(i < n)$ как $dT_i$. При этом для установления наличия связи экземпляра события $e_n$ с конкретным типом событий $t$ следует учитывать только множество ранее зафиксированных событий $\{e^t_1, e^t_2, \dots e^t_k\}$, $k < n$, поскольку $e$ является зафиксированным во времени свидетельством какого-либо действия.

В качестве второго признака корреляции предлагается учитывать *относительный вес прямых связей* между событиями. Относительный вес $w_i$ связи события $e_n$ с событием $e_i$ $(i < n)$ вычисляется по следующей формуле:

$$w_i = \frac{N_i^{iden}}{N_i^{direct}}, \quad (3)$$

где $N_i^{iden}$ – количество идентичных значений равнозначных свойств событий $e_n$ и $e_i$; $N_i^{direct}$ – общее количество равнозначных свойств между двумя событиями.

Таким образом, в результате обработки данных в ходе процесса корреляции событий необходимо сформировать два количественных ряда, соответствующих выделенным признакам для выполнения корреляции. Первый ряд содержит значения временных задержек, а второй ряд содержит относительные веса прямых связей между экземплярами событий одного типа и экземплярами событий другого или этого же типа. Это позволит

$$r_{12}^{priv} = \frac{n\sum_{i=1}^{n-1} w_i dT_i - \sum_{i=1}^{n-1} w_i \cdot \sum_{i=1}^{n-1} dT_i}{\sqrt{n\sum_{i=1}^{n-1}(w_i)^2 - (\sum_{i=1}^{n-1} w_i)^2} \cdot \sqrt{n\sum_{i=1}^{n-1}(dT_i)^2 - (\sum_{i=1}^{n-1} dT_i)^2}} \quad (4)$$

где $n$ – число ранее произошедших событий вместе с событием $e_n$.

определить частные коэффициенты корреляции между типами (или внутри типа) событий.

Для вычисления частного коэффициента корреляции между типами событий предлагается использовать коэффициент парной корреляции (коэффициент Пирсона). В этом случае частный коэффициент корреляции $r_{12}^{priv}$ между типами событий $t_1$ и $t_2$ вычисляется по формуле 4.

Общий коэффициент корреляции $r_{12}^{gen}$ между типами событий $t_1$ и $t_2$ может быть вычислен как среднее значение для частных коэффициентов по следующей формуле:

$$r_{12}^{gen} = \frac{\sum_{i=1}^{k} r_i^{priv}}{k} \quad (5)$$

где $r_i^{priv}$ – $i$-й частный коэффициент корреляции; $k$ – количество частных коэффициентов корреляции между типами $t_1$ и $t_2$.

Предложенный подход к установлению взаимоотношений между типами событий ориентирован на определение линейных зависимостей показателей схожести (относительных весов) от времени. Однако стоит отметить, что вычисление коэффициентов корреляции является предварительным этапом в оценке связей между типами, позволяющим уточнить направление ребер в графе связей типов событий.

***D. Определение временного окна анализа событий безопасности для их корреляции***

Достаточно важной задачей для выполнения корреляционного анализа, в том числе для определения необходимого и достаточного количества используемых данных, является определение статистической выборки, или *временного окна*. В этом случае необходимым условием успешного решения задачи является полнота (репрезентативность) и минимальная избыточность в анализируемом наборе исходных данных.

Одним из способов решения задачи вычисления временного окна является применение частотно-временного анализа зафиксированных действий. В данном случае экземпляры должны быть упорядочены по типам, поскольку на основе полученной выборки будет вычисляться коэффициент корреляции между типами событий. Данная операция подразумевает деление генеральной совокупности на временные интервалы, задаваемые экспертным путем, и вычисление частот появления экземпляров событий для каждого типа в каждом интервале. Пример частотно-временного анализа фрагмента журнала событий безопасности представлен на рис. 3.

На рис.3 отображено количество зафиксированных событий по типам в пределах каждого из периодов. В анализируемом фрагменте журнала выделено 5 типов (EventID в ОС Windows), а общее число периодов равно 100, по 5 минут каждый. С точки зрения вычисления временно-

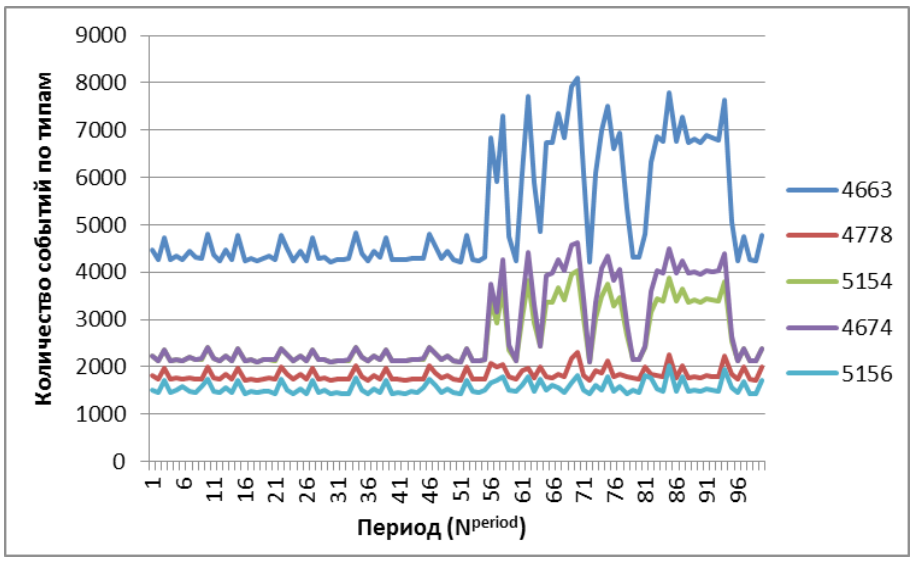


***Рис. 3.*** *График частот появления событий безопасности по типам в периодах*

го окна, представленный график не является достаточно информативным, хотя при визуальной оценке можно сказать, что в целом зафиксированные события имеют связь, а также частично обладают цикличным характером появления. Однако следует отметить наличие явно зависимого возмущения уровня частот событий типов 4663, 5154 и 4674. Также примечателен тот факт, что события последних двух типов имели одинаковые показатели частот до указанного момента времени.

Также необходимо отметить, что другим фактором оценки репрезентативности вместо частот появления могут (а в некоторых случаях и должны) выступать другие показатели, например, свойство схожести, отклонение коэффициента корреляции и другие.

Более эффективным способом вычисления временного окна является отслеживание относительного изменения средней частоты появления событий каждого типа по периодам за прошедшее время. График относительного изменения частот событий безопасности по типам при использовании данных, аналогичных данным, представленным на рис. 3, приведен на рис. 4.

При достижении максимально допустимого значения отклонения частоты событий (например, 0.005) на всем промежутке исходных данных (начиная с периода $T_0$) по отношению к среднему значению частоты за пройденное время (точка $T_k$), следует расширить временное окно, по меньшей мере, на один период, чтобы за это время была возможность расчета коэффициентов корреляции. При положительном исходе (отклонение частоты не превысило максимальный уровень) допускается дальнейшие выполнение процесса корреляции на множестве событий отрезка времени $[T_0; T_{k+1}]$ при $k > 2$, что позволит проверять выполнение требования определения необходимого объема исходных данных для анализа [34].

В представленном примере частотно-временного анализа взято прямое направление хода времени. Однако определение временного окна также может быть выполнено в обратном исчислении времени (так как производится анализ истории) с целью поиска момента $T_k$. Данный подход теоретически обеспечивает оценку репрезентативности выборки для ее извлечения из генеральной совокупности. В дальнейшем мы планируем расширить данную методику за счет использования других факторов оценки, например, динамичности изменения значений свойств событий, отклонения схожести и других.

Также следует отметить, что временное окно должно обязательно вычисляться с разделением по фактору группировки, если данная операция присутствует в анализе. В случае вычисления зависимости показателя схожести типов событий от временной задержки, подобным фактором является тип события. Это объясняется тем, что разные типы свидетельствуют о разных происходящих явлениях. Отсюда следует, что при обнаружении связи рассматриваемого экземпляра события с определенным явлением необходимо учитывать временное окно для извлечения выборки среди событий соответствующего типа.

**Реализация параллельных вычислений для решения задач корреляции событий безопасности**

В случае включения компонента корреляции в целевую инфраструктуру для определения косвенных однотипных связей между типами собы-

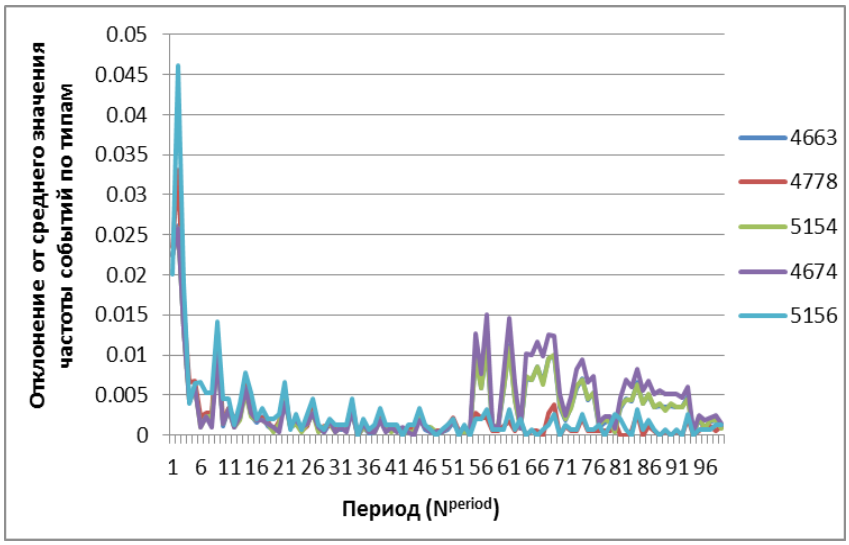


***Рис. 4.*** *График относительного изменения частот событий безопасности по типам*

тий в реальном времени необходимы значительные вычислительные мощности. Это вызвано тем, что количество возможных свойств и их значений может исчисляться тысячами и миллионами экземпляров соответственно. В свою очередь, задача вычисления частных и общих коэффициентов корреляции заключается в последовательном вычислении относительного веса связи $w_i^{rel}$ между экземплярами событий за счет сравнения их свойств, а также в расчете временной задержки $dT_i$ и их суммировании для нахождения частного коэффициента корреляции. Даже с использованием оптимального размера временного окна объем вычислений при решении этой задачи достаточно велик. В случае обычных локальных вычислений это приводит к большому времени обработки входных наборов данных о событиях безопасности. Длительность процесса корреляции данных в этом случае может варьироваться от нескольких десятков минут до нескольких часов. Наконец, решение задачи определения временного окна подразумевает мгновенную обработку миллионов элементов матрицы отношений между типами событий, если учитывать требование работы в режиме, близком к реальному времени [34].

Для того, что повысить эффективность обработки данных о событиях безопасности, в том числе снизить время, затрачиваемое на их обработку при корреляции событий безопасности, необходимо прибегнуть к методам организации параллельных и распределенных вычислений. Одним из способов организации такого рода вычислений является использование технологий обработки больших данных, в том числе на основе программной платформы Spark. Для выполнения параллельных вычислений для процесса корреляции событий безопасности был выбран способ разделения входных данных на части и анализ по частям в разных потоках. Фрагмент исходного кода, реализующего решение задачи определения косвенных однотипных связей, представлен на далее:

```
if __name__ == «__main__»:
        conf = SparkConf()
     conf.setAppName(«Analysis»)
    sc = SparkContext(«local[8]», conf=conf)

     input_path = «...»
     files_list = [join(input_path, f) for f
in listdir(input_path) if isfile(join(input_
path, f))]
     files = sc.parallelize(files_list)
     result = files.map(calcOverallRs)
     output = result.collect()

     output_file = open(«...», ‹w›)
     for lst in output:
         for r in lst:
              output_file.write(«Overall R =
{}.\n».format(r))
            output_file.close()
```

В приведенном на выше коде вначале создается и устанавливается конфигурация приложения Spark. Затем создается точка входа `SparkContext`, которая предоставляет связь с кластером Spark. При создании точки входа указывается, в каком режиме запускается задача. В рассмотренном выше случае, задача запускается локально в восьми параллельных потоках. Однако, кроме локального режима, здесь можно указать и распределенный режим под управлением менеджеров ресурсов YARN из пакета Hadoop или Mesos.

В переменной `input_path` указывается папка, файлы из которой станут входными данными. Пути к входным файлам размещаются в переменной `files_list` и составляют локальную коллекцию. Далее с помощью функции `parallelize` локальная коллекция преобразуется в распределенную. Для каждого элемента распределенной коллекции `files` далее применяется оператор отображения `map`, который вызывает функцию, переданную ему в качестве параметра для всех элементов `files`. При этом запуск этой функции осуществляется в отдельных параллельных потоках для каждого элемента. Функция `calcOverallRs` является функцией вычисления общих коэффициентов корреляции. Преобразование распределенной коллекции в локальную осуществляется с помощью функции `collect`. Завершающие действия кода предназначены для сохранения результатов вычисления коэффициентов корреляции в файл.

Рассмотрим теперь алгоритмы реализации параллельных вычислений для нахождения общих коэффициентов корреляции. Участок кода Spark, представленный на следующем отрывке кода, находит все уникальные типы событий `eventIDs` во входном dataset:

```
tree = ElementTree.parse(file)
root = tree.getroot()
ids  =  root.findall(«./ns:Event/ns:System/
ns:EventID»
eventIDs = set(id.text for id in ids)
```

Поиск уникальных типов событий происходит следующим образом. Вначале строится дерево XML-файла `tree`. В результате получается корень дерева `root`. Далее выполняется поиск всех тегов `EventID`, вложенных в корень. Для найденных тегов берутся их значения (`text`), которые составляют набор всех типов событий `eventIDs` во входном файле.

После того, как найдены уникальные типы событий, события группируются по типам. Для каждого типа событий вначале происходит вычисление частных коэффициентов корреляции. Затем находятся общие коэффициенты корреляции.

Код Spark, реализующий этот алгоритм, представлен ниже. При этом следует отметить, что `eventTypes` является словарем, в котором по значению типа можно получить все события, относящиеся к данному типу.

```
   for typeA, typeB in product(eventTypes,
eventTypes):
   dataNames1 = findTypeAEventDataNames()
   dataNames2 = findTypeBEventDataNames()

   # Find direct connections by intersecting
   directConns = list(dataNames1 & dataNames2)
   partialRs = []
   k = 0
   for e1 in eventTypes[typeA]:
      relativeWeights = []
      dTs = []
      t1 = e1.getTimestamp()
      nV1 = e2.findDataNameValues()
      for e2 in eventTypes[typeB]:
          t2 = e2.getTimestamp()
          dt = (t1 - t2).total_seconds()

          # Considering only past events to t1
          if dt <= 0:
              continue
          nV2 = e2.findDataNameValues()

          # k is the number of matching values
          k = sum([1 if nV1[i] == nV2[i] else
0 for i in range(len(directConns))])

          # if no direct conns is found then w
= 0
              w = k / len(directConns) if
len(directConns) else 0
          relativeWeights.append(w)
          dTs.append(dt)

      n = len(dTs)
      numerator =
   n*sum([relativeWeights[i]*dTs[i] for i in
range(n)]) - sum(relativeWeights)*sum(dTs)
      denominator = sqrt(n*sum([pow(rel
ativeWeights[i], 2) for i in range(n)]) -
pow(sum(relativeWeights), 2)) * \
   sqrt(n * sum([pow(dTs[i], 2) for i in
range(n)]) - pow(sum(dTs), 2))
      partialR = numerator / denominator
      partialRs.append(partialR)
      relativeWeights.clear()
      dTs.clear()

   overallR = sum(partialRs) / len(partialRs)
   print(«Overall R =», overallR)
   partialRs.clear()
```

Сначала находятся имена свойств типов `dataNames`, затем находятся все прямые связи типов путем пересечения множеств `dataNames`. Далее попарно для всех событий двух типов вычисляются временные задержки $dT_i$ и относительные веса прямых связей $w_i$. Если временная задержка $dT_i$ имеет отрицательное значение, то это событие не берется в рассмотрение, поскольку оно произошло после текущего.

После вычисления всех $w_i$ и $dT_i$ вычисляются частные коэффициенты корреляции в соответствии с (4). На основе частных коэффициентов корреляции далее вычисляется общий коэффициент корреляции в соответствии с (5).

**Результаты экспериментов**

Для проведения экспериментов использовалась платформа Supermicro X9DRL-3F с двумя процессорами Intel Xeon E5-2620 v2 @ 2.1 ГГц. На базе гипервизора ESXi 6.0 была создана виртуальная машина с операционной системой Ubuntu Server 16.04. Машине было выделено 8 thread, зарезервировано (reserved) 2 ГГц процессора, также было зарезервировано 32 ГБ ОЗУ. Поверх операционной системы была установлена программная платформа Spark 1.6.1.

В работе для проведения экспериментальной оценки в целях повышения достоверности полученных результатов используется входной поток данных, собранный с офисного компьютера. В качестве входных данных был взят файл размером 7 ГБ, в котором содержалось приблизительно 7 миллионов событий, относящихся к 81 типу. Исходные данные были предварительно собраны из журнала событий безопасности ОС Windows 8 на офисном компьютере за период, примерно равный 2 суткам.

В ходе анализа исходных данных было установлено, что все типы событий можно разделить на две группы: (1) часто встречающиеся и (2) редко

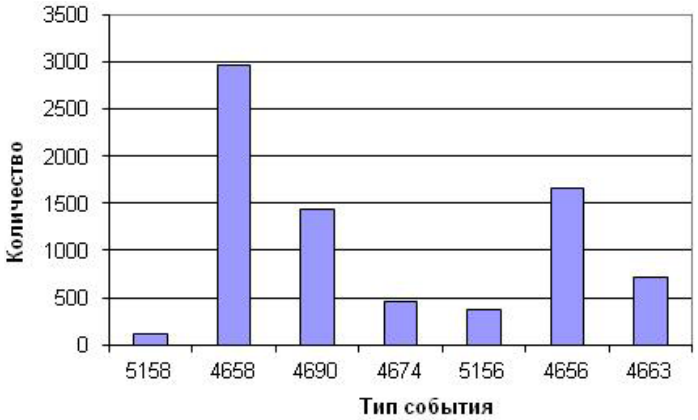


***Рис. 5**. Статистика часто встречающихся типов событий*

встречающиеся. К первой группе относятся типы событий со значением *Count* менее 100, ко второй – остальные типы.

Показатели количества произошедших событий первой и второй группы представлены на рис. 5 и рис. 6 соответственно.

Из рис. 5 и рис. 6 видно, что наиболее сложными являются вычисления относительных весов прямых связей между такими типами, как 4658 и 4656. Это приводит к необходимости обрабатывать 2959 * 1664 пар событий в режиме, близком к реальному, что является чрезвычайно сложной задачей, если не использовать параллельные вычисления и не ограничивать анализируемую выборку.

При выполнении задачи выявления косвенных связей в ходе анализа исходных данных было определено 134 уникальных свойства типов событий. Пересечение множеств значений данных свойств позволило выявить 15 свойств, относящихся друг к другу как неравнозначные однотипные. Уточнение графа связей типов событий за счет дополнения узлов соответствующими косвенными связями увеличило количество ребер (общее количество связей) примерно на 20%.

Статистические подходы при выполнении процесса корреляции событий безопасности применялись ранее. Сложность примеснения вероятностной оценки заключается в необходимости обеспечения соответствия исходных данных определенным требованиям, например, их однородности, нормальному закону распределения и другим. Выбор конкретного статистического показателя (коэффициента корреляции, коэффициента детерминации и пр.) и возможность его применения в реализации процесса корреляции можно определить в ходе анализа исходных данных и результатов вычисления такого показателя. В данной работе предложено использовать операцию вычисления показателя линейной корреляции Пирсона с учетом факторов относительной схожести событий ($w$) и временной задержки их возникновения ($dT$). Для конкретизации и сравнения результатов коэффициенты корреляции вычисляются между парами типов событий, поскольку показатель относительной схожести зависит от типовой структуры событий. В данном случае важность фактора схожести и временной задержки для общего процесса корреляции не учитывается.

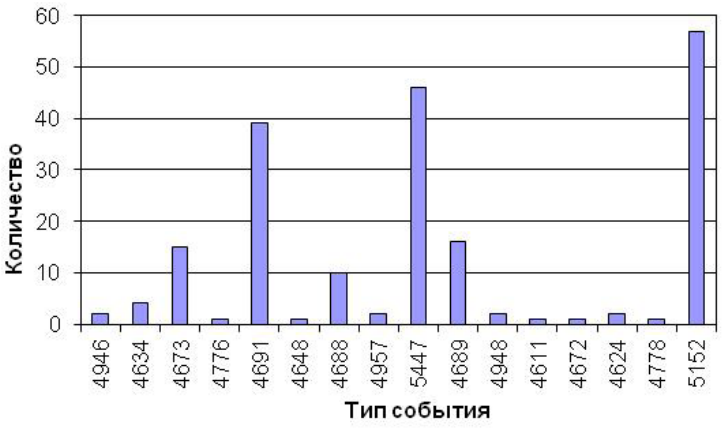


***Рис. 6.** Статистика редко встречающихся типов событий*

*Таблица 1.*
*Значения для времени обработки входных наборов данных*

| Режим обработки | Объем, МБ | Время, сек |
|---|---|---|
| локальный | 9,1 | 3876 |
| параллельный | 9,1 | 531 |
| параллельный | 18,2 | 1039 |
| параллельный | 27,3 | 1573 |
| параллельный | 36,4 | 2074 |
| параллельный | 45,5 | 2615 |
| параллельный | 54,6 | 3136 |
| параллельный | 63,7 | 3637 |
| параллельный | 72,8 | 4198 |

Вычисленные коэффициенты корреляции могут быть использованы для предварительной оценки происходящих процессов в анализируемой инфраструктуре. Стоит отметить, что результаты данных вычислений свидетельствуют о наличии и характере линейной связи между указанными факторами, однако отсутствие линейной связи не может свидетельствовать об отсутствии связи как таковой (возможно наличие сложной нелинейной связи).

Экспериментальная оценка оперативности реализации процесса корреляции на рассмотренном выше наборе данных была проведена с учетом двух сценариев на основании решения задачи поиска общих коэффициентов корреляции. В первом случае использовался однопоточный режим при частоте процессора 2 ГГц. При этом объем входного набора данных равнялся 9,1 МБ. В этом сценарии полагалось, что параллелизация вычислений для выполнения процесса корреляции событий безопасности отсутствовала. Результаты, полученные в этом сценарии, рассматриваются как контрольные, с которыми сравниваются результаты, полученные при реализации параллельных вычислений.

Во втором сценарии применялся метод параллелизации вычислений на основе разделения набора данных на части. Вычисления производились на процессоре 8x2ГГц ЦПУ с реализацией восьми потоков. В качестве входного набора использовался набор данных, увеличенный по сравнению с предыдущим в 8 раз. Таким образом, максимальный объем входного набора данных достигал 72,8 МБ.

Результаты экспериментальной оценки времени обработки входных наборов данных представлены в табл. 1.

Зависимости времени обработки данных от объема входного набора данных представлены на рис. 7.

Анализируя данные на рис. 7 можно сделать следующие выводы. Во-первых, время обработки данных при решении задачи корреляции событий безопасности уменьшается с увеличением количества параллельных потоков. При этом эта зависимость близка к прямо пропорциональной, т.е. применение восьми потоков позволило сократить время обработки приблизительно в восемь раз. С другой стороны, зависимость времени обработки данных от объема входного набора близка к линейному виду. Так, если объем входного набора увеличить в восемь раз, то время обработки данных также увеличится примерно в восемь раз. Это является вполне закономерным результатом.

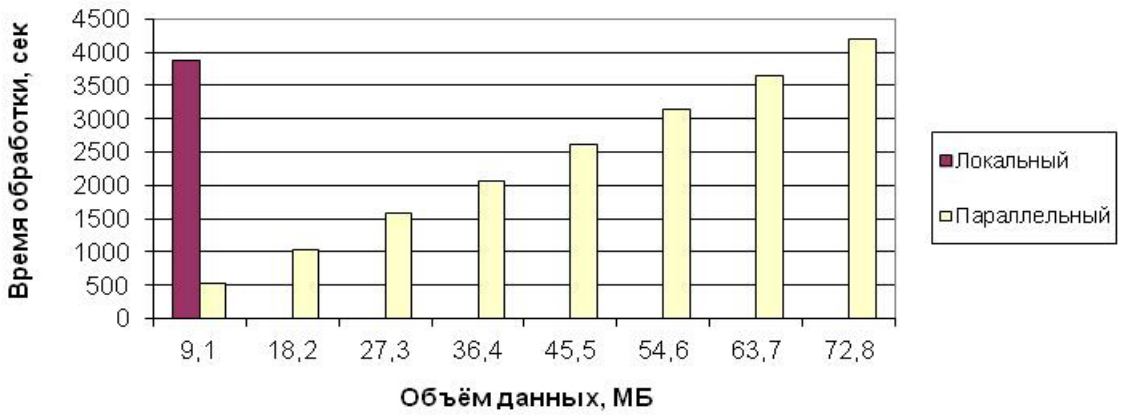


*Рис. 7. Зависимости времени обработки от объема входного набора данных*

Сравнение с методами машинного обучения для анализа сетевого трафика, например [32], показало, что время вычисления коэффициентов корреляции сравнимо с временем извлечения признаков на платформе Spark и значительно меньше времени выполнения этапа обучения при сопоставлении объемов данных и вычислительных ресурсов. В целом экспериментальная оценка времени обработки данных в процессе корреляции событий безопасности показывает, что применение параллельных вычислений позволяет реализовать требования по выполнению этого процесса в реальном или близком к реальному времени.

**Выводы**

В настоящей работе предложен новый подход к реализации процесса корреляции событий безопасности. Данный подход базируется на выявлении связей между событиями безопасности и оценке зависимости их относительной силы от распределения событий во времени, основанной на вычислении и использовании коэффициентов корреляции, а также распараллеливании вычислений с использованием технологий обработки больших данных. Рассмотрено содержание процесса корреляции событий безопасности и выделены основные задачи, требующие для своего решения значительных вычислительных ресурсов. Показана возможность и оценена результативность применения параллельных вычислений на примере решения задачи вычисления частных и общих коэффициентов корреляции на множествах связей между типами событий. В ходе выполнения задачи поиска косвенных однотипных связей были получены результаты, позволяющие уточнять граф связей между типами событий журнала безопасности ОС Windows, построенным по прямым связям, примерно на 20 процентов. Экспериментальные результаты показали, что время обработки данных в процессе корреляции событий безопасности находится в обратно пропорциональной зависимости от количества реализованных в Spark параллельных потоков. При этом зависимость времени обработки от объема входного набора данных близка к линейному виду. Это позволяет утверждать, что применение параллельных вычислений позволяет реализовать требования по выполнению этого процесса с учетом предварительной оценки исходных данных в реальном или близком к реальному времени.

В дальнейшем предполагается формирование списка значимых факторов из множества возможных факторов и выбор наиболее подходящих статистических показателей, характеризующих связь между ними, для оценки связанности конкретных экземпляров событий. Предполагается исследование подходов с использованием различных факторов оценки и группировки событий. Полученные результаты планируется использовать при разработке алгоритмов решения задач корреляции с учетом комплексирования выполняемых операций. Дальнейшими направлениями исследований также является интеграция функциональных возможностей Spark в среду существующих SIEM-систем.

***Рецензент**: Молдовян Александр Андреевич, доктор технических наук, профессор, заведующий научно-исследовательским отделом проблем информационной безопасности Федерального государственного бюджетного учреждения науки «Санкт-Петербургский институт информатики и автоматизации Российской академии наук», Санкт-Петербург, Россия. E-mail: maa1305@yandex.ru*

**Литература**

1. И.В. Котенко, И.Б. Саенко, Р.М. Юсупов. Новое поколение систем мониторинга и управления инцидентами безопасности // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2014. № 3(198). С. 7–18.
2. И.В. Котенко, И.Б. Саенко. К новому поколению систем мониторинга и управления безопасностью // Вестник Российской академии наук, Том 84, № 11, 2014, С.993–1001.
3. И.В. Котенко, И.Б. Саенко. SIEM-системы для управления информацией и событиями безопасности // Защита информации. Инсайд, 2012, № 5, С.54-65.
4. I. Kotenko, O. Polubelova, and I. Saenko. «The ontological approach for SIEM data repository implementation», in Proceedings of the 2012 IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing, 2012, pp. 761–766.
5. И.В. Котенко, И.Б. Саенко, О.В. Полубелова, А.А. Чечулин. Применение технологии управления информацией и событиями безопасности для защиты информации в критически важных инфраструктурах // Труды СПИИРАН. Вып.1 (20), 2012. С.27-56.
6. C. Kruegel, F. Valeur, and G. Vigna, Intrusion Detection and Correlation: Challenges and Solutions, Springer, 2004, 118 p.
7. I.V. Kotenko, A.A. Chechulin. A Cyber Attack Modeling and Impact Assessment Framework. Proceedings of 5th International Conference on Cyber Conflict 2013 (CyCon 2013). 2013, pp. 119–142.
8. И.В. Котенко, И.Б. Саенко. Построение системы интеллектуальных сервисов для защиты информации в условиях кибернетического противоборства // Труды СПИИРАН. 2012. № 3(22). С. 84–100.
9. И.В. Котенко, И.Б. Саенко. Архитектура системы интеллектуальных сервисов защиты информации в критически важных инфраструктурах // Труды СПИИРАН. 2013. № 1(24). С. 21–40.

10. А.В. Федорченко, Д.С. Левшун, А.А. Чечулин, И.В. Котенко. Анализ методов корреляции событий безопасности в SIEM-системах. Часть 2 // Труды СПИИРАН. 2016. Вып. 6(49). С.5-27.
11. И.В. Котенко. Интеллектуальные механизмы управления кибербезопасностью // Управление рисками и безопасностью // Труды Института системного анализа Российской академии наук. Т.41, 2009. С.74-103.
12. R. Sadoddin and A. Ghorbani, «Alert correlation survey: framework and techniques», in Proceedings of 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services (PST`06), 2006, Article no. 37.
13. A. Hanemann and P. Marcu, «Algorithm design and application of service-oriented event correlation», in Proceedings of Conference BDIM 2008, 3rd IEEE/IFIP International Workshop on Business-Driven IT Management, 2008, pp. 61–70.
14. T. Limmer and F. Dressler, «Survey of event correlation techniques for attack detection in early warning systems», Tech. report, University of Erlangen, Dept. of Computer Science 7, 2008, 37 p.
15. A. Muller. «Event correlation engine», Master`s Thesis, Swiss Federal Institute of Technology Zurich, 2009, 165 p.
16. P. Ning and D. Xu. «Correlation analysis of intrusion alerts», in Intrusion Detection Systems: series Advances in Information Security, Springer, 2008, vol. 38, pp. 65–92.
17. A.A. Ghorbani, W. Lu, and M. Tavallaee, Network Intrusion Detection and Prevention, Springer, 2010, 224 p.
18. M. Hasan. «A conceptual framework for network management event correlation and filtering systems», in Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management, 1999, pp. 233–246.
19. U. Zurutuza and R. Uribeetxeberria, «Intrusion detection alarm correlation: a survey», in Proceedings of IADAT International Conference on Telecommunications and computer Networks, 2004, pp. 1–3.
20. D.W. Guerer, I. Khan, R. Ogler, and R. Keffer, An Artificial Intelligence Approach to Network Fault Management, SRI International, 1996, 10 p.
21. M. Tiffany. «A survey of event correlation techniques and related topics», http://www.tiffman.com/netman/netman.html.
22. H.T. Elshoush and I. M. Osman, «Alert correlation in collaborative intelligent intrusion detection systems – A survey», in Applied Soft Computing, 2011, pp. 4349–4365.
23. G. Kou, Y. Lu, Y. Peng, Y. Shi. Evaluation of classification algorithms using MCDM and rank correlation. International Journal of Information Technology & Decision Making, No. 1, Vol. 11, 2012, pp.197-225.
24. W. Wei, F. Chen, Y. Xia, G. Jin. A Rank Correlation Based Detection against Distributed Reflection DoS Attacks. IEEE Communications Letters, vol. 17, is. 1, 2013, pp.173-175.
25. G. Beliakov, J. Yearwood, A. Kelarev. Application of Rank Correlation, Clustering and Classification in Information Security. Journal of Networks, No 7, 2012, pp 935-945.
26. G. Jiang, G. Cybenko. Temporal and spatial distributed event correlation for network security. American Control Conference 2004, IEEE Xplore, 2004, pp.996-1001.
27. M. Ficco. Security event correlation approach for cloud computing. International Journal of High Performance Computing and Networking, Vol. 7, No. 3, 2011, pp. 173–185.
28. M. Davis, E. Korkmaz, A. Dolgikh, V. Skormin. Resident Security System for Government/Industry Owned Computers // Computer Network Security. Lecture Notes in Computer Science, Springer-Verlag, Vol.10446, pp.185-194.
29. V. Gulisano, R. Jimenez-Peris, M. Patiño-Martinez, C. Soriente, P. Valduriez. StreamCloud: An Elastic and Scalable Data Streaming System. IEEE Transactions on Parallel and Distributed Systems, Volume: 23, Issue: 12, Dec. 2012, pp. 2351-2365.
30. S. Marchal, X. Jiang, R. State, and T. Engel, «A big data architecture for large scale security monitoring», in Proceedings of the IEEE International Congress on Big Data, 2014, pp. 56 63.
31. D Priya, «Network anomaly detection on fast streaming data using spark», in International Journal of Advanced Research in Computer and Communication Engineering, 2016, vol. 5, issue 7, pp. 487-489.
32. G. Koutsoumpakis, Spark-Based Application for Abnormal Log Detection, Uppsala Universitet, 2014, 40 p.
33. Y. Bar-Yam, Dynamics of Complex Systems, Perseus Books, Cambridge, MA, USA, 1997.
34. A. Fedorchenko, I. Kotenko, and D. El Baz, «Correlation of security events based on the analysis of structures of event types», in Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2017), 2017. pp 270-276.

# BIG DATA TECHNOLOGIES FOR SECURITY EVENT CORRELATION BASED ON EVENT TYPE ACCOUNTING

***Kotenko I.[6], Fedorchenko A.[7], Saenko I.[8], Kushnerevich A.[9]***

*The work is devoted to the research of the approach to parallel data processing for performing the tasks of security events correlation based on big data technologies. Various methods and tasks of security events correlation, as well as technologies for processing big data, applicable for security monitoring tasks are considered. The main attention is paid to the tasks of identifying the links between security events types and the*

6 Igor Kotenko, Dr.Sc., Professor, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, ITMO University, Saint-Petersburg, ivkote@comsec.spb.ru

7 Andrey Fedorchenko , St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, ITMO University, Saint-Petersburg, fedorchenko@comsec.spb.ru

8 Igor Saenko, Dr.Sc., Professor, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, ITMO University, Saint-Petersburg, ibsaen@comsec.spb.ru

9 Alexey Kushnerevich, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, ITMO University, Saint-Petersburg, kushnerevich@comsec.spb.ru

*evaluation of the strength dependence of the events distribution in time. The implementation of the correlation tasks on the Spark platform is described, and the results of the experimental evaluation of the security events correlation processes are given.*

***Keywords:*** *events correlation, security events, Pearson's correlation coefficient, SIEM-systems, big data, parallel data processing*

**References**

1. I.V. Kotenko, I.B. Saenko, and R.M. Yusupov, «New Generation of Security information and Event Management Systems», in St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunication and Control Systems. 2014. vol. 3(198). pp. 7–18.
2. I.V. Kotenko and I.B. Saenko. «Creating New Generation Cybersecurity Monitoring and Management Systems», in Herald of the Russian Academy of Sciences, 2014, vol.84, no.6, pp.993-1001.
3. I.V. Kotenko and I.B. Saenko. «SIEM-systems for security information and events management», in Protection of the information. Inside, 2012, no.5, pp.54–65.
4. I. Kotenko, O. Polubelova, and I. Saenko, «The ontological approach for SIEM data repository implementation», in Proceedings of the 2012 IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing, 2012, pp. 761–766.
5. I.V. Kotenko, I.B. Saenko, O.V. Polubelova, and A.A. Chechulin, «Application of security information and event management technology for information security in critical infrastructures», SPIIRAS Proceedings, 2012, Issue 1(20), pp.27-56.
6. C. Kruegel, F. Valeur, and G. Vigna, «Intrusion Detection and Correlation: Challenges and Solutions», Springer, 2004, 118 p.
7. I.V. Kotenko, and A.A. Chechulin, «A Cyber Attack Modeling and Impact Assessment Framework» in Proceedings of 5th International Conference on Cyber Conflict 2013 (CyCon 2013). 2013, pp. 119–142.
8. I.V. Kotenko and I.B. Saenko, «Developing the system of intelligent services to protect information in cyber warfare», in SPIIRAS Proceedings, 2012, Issue 3(22), pp.84-100.
9. I.V. Kotenko and I.B. Saenko, «Architecture of the system of intelligent services to protect information in cyber warfar», SPIIRAS Proceedings. 2012. vol. 1(24). pp. 21–40.
10. A.V. Fedorchenko, D.S. Levshun, A.A. Chechulin, and I.V. Kotenko, «Analysis of security event sorrelation methods in SIEM-systems. Part 2», in SPIIRAN Proceedings. 2016. Issue. 6(49). pp.5-27.
11. I.V. Kotenko, «Intelligent mechanisms of cybersecurity management», in Risk and security management. Proceedings of the Institute of System Analysis of the Russian Academy of Sciences, 2009, Vol.41, pp.74-103.
12. R. Sadoddin and A. Ghorbani, «Alert correlation survey: framework and techniques», in Proceedings of 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services (PST`06), 2006, Article no. 37.
13. A. Hanemann and P. Marcu, «Algorithm design and application of service-oriented event correlation», in Proceedings of Conference BDIM 2008, 3rd IEEE/IFIP International Workshop on Business-Driven IT Management, 2008, pp. 61–70.
14. T. Limmer and F. Dressler, «Survey of event correlation techniques for attack detection in early warning systems», Tech. report, University of Erlangen, Dept. of Computer Science 7, 2008, 37 p.
15. A. Muller. «Event correlation engine», Master`s Thesis, Swiss Federal Institute of Technology Zurich, 2009, 165 p.
16. P. Ning and D. Xu. «Correlation analysis of intrusion alerts», in Intrusion Detection Systems: series Advances in Information Security, Springer, 2008, vol. 38, pp. 65–92.
17. A.A. Ghorbani, W. Lu, and M. Tavallaee, Network Intrusion Detection and Prevention, Springer, 2010, 224 p.
18. M. Hasan. «A conceptual framework for network management event correlation and filtering systems», in Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management, 1999, pp. 233–246.
19. U. Zurutuza and R. Uribeetxeberria, «Intrusion detection alarm correlation: a survey», in Proceedings of IADAT International Conference on Telecommunications and computer Networks, 2004, pp. 1–3.
20. D.W. Guerer, I. Khan, R. Ogler, and R. Keffer, An Artificial Intelligence Approach to Network Fault Management, SRI International, 1996, 10 p.
21. M. Tiffany. «A survey of event correlation techniques and related topics», http://www.tiffman.com/netman/netman.html.
22. H.T. Elshoush and I. M. Osman, «Alert correlation in collaborative intelligent intrusion detection systems – A survey», in Applied Soft Computing, 2011, pp. 4349–4365.
23. G. Kou, Y. Lu, Y. Peng, Y. Shi. Evaluation of classification algorithms using MCDM and rank correlation. International Journal of Information Technology & Decision Making, No. 1, Vol. 11, 2012, pp.197-225.
24. W. Wei, F. Chen, Y. Xia, G. Jin. A Rank Correlation Based Detection against Distributed Reflection DoS Attacks. IEEE Communications Letters, vol. 17, is. 1, 2013, pp.173-175.
25. G. Beliakov, J. Yearwood, A. Kelarev. Application of Rank Correlation, Clustering and Classification in Information Security. Journal of Networks, No 7, 2012, pp 935-945.
26. G. Jiang, G. Cybenko. Temporal and spatial distributed event correlation for network security. American Control Conference 2004, IEEE Xplore, 2004, pp.996-1001.
27. M. Ficco. Security event correlation approach for cloud computing. International Journal of High Performance Computing and Networking, Vol. 7, No. 3, 2011, pp. 173–185.
28. M. Davis, E. Korkmaz, A. Dolgikh, V. Skormin. Resident Security System for Government/Industry Owned Computers // Computer Network Security. Lecture Notes in Computer Science, Springer-Verlag, Vol.10446, pp.185-194.
29. V. Gulisano, R. Jimenez-Peris, M. Patiño-Martinez, C. Soriente, P. Valduriez. StreamCloud: An Elastic and Scalable Data Streaming System. IEEE Transactions on Parallel and Distributed Systems, Volume: 23, Issue: 12, Dec. 2012, pp. 2351-2365.
30. S. Marchal, X. Jiang, R. State, and T. Engel, «A big data architecture for large scale security monitoring», in Proceedings of the IEEE International Congress on Big Data, 2014, pp. 56 63.
31. D Priya, «Network anomaly detection on fast streaming data using spark», in International Journal of Advanced Research in Computer and Communication Engineering, 2016, vol. 5, issue 7, pp. 487-489.
32. G. Koutsoumpakis, Spark-Based Application for Abnormal Log Detection, Uppsala Universitet, 2014, 40 p.
33. Y. Bar-Yam, Dynamics of Complex Systems, Perseus Books, Cambridge, MA, USA, 1997.
34. A. Fedorchenko, I. Kotenko, and D. El Baz, «Correlation of security events based on the analysis of structures of event types», in Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2017), 2017. pp 270-276.