В автоматизированной системе реализована процедура формирования тезауруса, а также дерева связей. Подобные функции вызываются посредством графического интерфейса и выводятся в отдельных модальных диалогах для просмотра результата построения. Для вызова построителя тезауруса необходимо выполнить команду Построить -> Тезаурус.> (рис. 6).

Кроме построения тезауруса можно вызвать одну из команд построения дерева связей для конкретного типа связей, например для связи типа «ЕСТЬ_НЕКОТОРЫЙ».

Вывод тезауруса показан на экране.

СПИСОК ЛИТЕРАТУРЫ

1. Шеховцов О. И. Информационная технология предпроектного обследования производственных систем // Изв. ГЭТУ. «Информационные технологии в технических и организационных системах». 1997. Вып. 514. С. 95–103.

2. Давыдов С., Ефимов А. IntelliJ IDEA. Профессиональное программирование на Java. СПб., 2005.

V. P. Permyakov, O. I. Shekhovtsov
Saint Petersburg Electrotechnical University «LETI»

THE VISUALIZATION SYSTEM IS THE ONTOLOGICAL MODEL OF THE SUBJECT AREA

Describes a system for the visualization of ontological models of knowledge representation: the System is implemented using object oriented programming in Java using the graphical library Swing. First provides basic information about the structure of the ontological formalism, and then the screenshots reflecting the view of the components of the model. The development of PI and PZ the levels of the model is made by new improved interface.

Examples Windows to create domain objects, their attributes, and their properties and characteristics, and define relationships between objects. Relationships are defined by straight lines of different colors. Through a graphical interface, you can create an unlimited number of system objects and edit their attributes and relationships according to the data model.

Visualization system, ontological model, domain-independent model, problem-oriented model, concepts, connection

УДК 004.056

Е. С. Новикова, Я. А. Бекенева, А. В. Шоров
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Исследование методов корреляции событий безопасности для обеспечения безопасности облачных вычислительных сред

Представлены результаты анализа процесса корреляции событий для решения задач информационной безопасности, представлено описание основных его этапов и операций, выполняемых над событиями. Рассмотрены возможные способы построения компонента корреляции событий, описаны их достоинства и недостатки.

Безопасность облачных технологий, корреляция событий безопасности, операции над событиями, архитектура модуля корреляции событий

Системы облачных вычислений привлекательны для использования в бизнес-процессах благодаря ряду серьезных преимуществ, таких, как легкое конфигурирование вычислительных устройств под необходимые нужды организации, гибкость и эластичность предоставляемых облач-

ных сервисов и экономическая выгода. Новая парадигма построения вычислительных систем, связанная с разделением общих вычислительных ресурсов, аутсорсингом, возможностью совместного использования услуг между арендаторами (multi-tenacy), ставит новые проблемы-вызовы, связанные с обеспечением безопасного использования информационных ресурсов системы облачных вычислений. Для обеспечения их бесперебойного функционирования и защищенности данных пользователей необходимо развивать системы их защиты от различных угроз, включая сложные целевые атаки, выполняемые в несколько этапов, часто разнесенных во времени. Обнаружение таких атак требует тщательного анализа событий безопасности, получаемых от различных датчиков безопасности и объектов облачной инфраструктуры за длительный период времени. Многие современные системы обнаружения вторжений не могут установить взаимосвязи между событиями безопасности в виде последовательности этапов выполнения атаки, поскольку не способны анализировать текущие угрозы в историческом контексте [1]. Кроме того, в большинстве случаев они не осуществляют оценку достоверности генерируемых событий безопасности, а критичность события безопасности часто не зависит от уровня критичности контролируемых ресурсов. Корреляция этих двух показателей позволила бы администратору безопасности более точно расставлять приоритеты событиям безопасности для своевременного реагирования на них. Для устранения этих недостатков в системах управления информационной безопасностью предлагается использовать в качестве составного компонента модуль корреляции событий безопасности.

В настоящей статье рассматриваются основные этапы корреляции событий и основные операции над событиями, а также основные подходы к построению современных систем корреляции событий безопасности.

Основные этапы корреляции событий. К основным задачам модуля корреляции событий безопасности следует отнести:

– снижение уровня избыточных событий безопасности;

– фильтрацию событий безопасности с низким уровнем достоверности;

– определение сценариев атак;

– фильтрацию и расстановку приоритетов событиям безопасности;

– предсказание следующего действия злоумышленника;

– предсказание других возможных атак.

Под *событием* в информационных системах понимается создание некоторой регистрационной записи о происходящей в системе активности в ответ на изменение состояния информационной системы, ее ресурсов, процессов и т. д. Обычно такая запись отражает 3 основные характеристики активности – тип (или форма) события, значимость события и связь с другими событиями [2]. Тип события обозначает атрибуты или компоненты информационной системы, вовлеченные в некоторый процесс. Связь с другими событиями (процессами) в системе выражается в форме установления взаимоотношений между другими событиями (или процессами) – она может быть причинно-следственной (каузальной) или объединяющей (если событие является абстракцией множества низкоуровневых событий). Примером понятия «событие» в области информационной безопасности облачных технологий могут служить события безопасности/уведомления от межсетевых экранов, систем обнаружения вторжений, событий гипервизора виртуальных машин и т. д.

Процесс корреляции событий в общем случае независимо от их природы связан с двумя основными понятиями [3], [4]:

– обработкой событий;

– корреляцией событий.

Таблица 1

Этап	Выполняемое действие
Фильтрация событий	Удаление событий, не релевантных решаемой задаче и наблюдаемым процессам в системе
Агрегация событий	Слияние дублирующих данных для одного события
Анализ основных причин (root cause analysis)	Анализ зависимостей между событиями, в большинстве случаев на основе построенной модели окружающей среды и графа зависимостей, для того чтобы понять, могут ли одни события объяснить другие. Это ключевой этап всего процесса корреляции событий, и он может быть разбит на подэтапы, характер которых зависит от непосредственно решаемой задачи. Например, при обнаружении сетевых вторжений данный этап может состоять из подэтапов – восстановление хода атаки, восстановление сессии атаки, определение и ход атаки и т. д. [5], [6]
Маскировка событий	Игнорирование событий, поступающих от систем, зависимых от отказавшей системы

Под обработкой событий понимаются вычисления и операции, определенные на множестве событий, целью которых чаще всего является их чтение, создание, преобразование и удаление. Корреляция событий – это способ извлечения высокоуровневых знаний из информации, представленной в виде множества событий. Примером знаний, которые могут быть получены в результате корреляции событий, являются определение аномальной ситуации, предсказания возможных будущих ситуаций или выявление первопричины аномалии. Очевидно, что эти знания непосредственно могут быть использованы при решении задачи обеспечения информационной безопасности облачных технологий. Для получения таких знаний корреляция событий выполняется в 4 основных этапа [4] (табл. 1).

Эти этапы представлены во многих других научно-исследовательских работах [6]–[8] и обычно дополняются этапами, связанными с предварительной обработкой данных (сбором, стандартизацией формата событий, получаемых от различных источников данных и их нормализацией) и реагированием на результаты корреляции. Например, в [9] процесс корреляции события дополнен этапом, на котором осуществляется выбор контрмер. В [5] этапы корреляции объединены в логические группы – сбор событий безопасности; агрегация и верификация событий безопасности; формирование событий высокого уровня. На рис. 1 представлена общая схема процесса корреляции событий [5], [10].

Основные операции над событиями. Непосредственно процесс выявления паттернов в событиях (или анализ сложных событий) может быть описан с помощью множества операций [11]–[13]. Выбор операций зависит от решаемой задачи и, соответственно, этапа процесса корреляции [7]. По своему смыслу операции могут быть разделены на 3 основные группы: операции фильтрации, которые осуществляют отбор событий, участвующих в процессе корреляции; операции сопоставления, которые выполняют поиск паттернов среди входящих событий и создают новые события, которые удовлетворяют некоторому шаблону; операции вывода, которые используют выходные данные операции сопоставления для создания новых событий или настройки параметров событий.

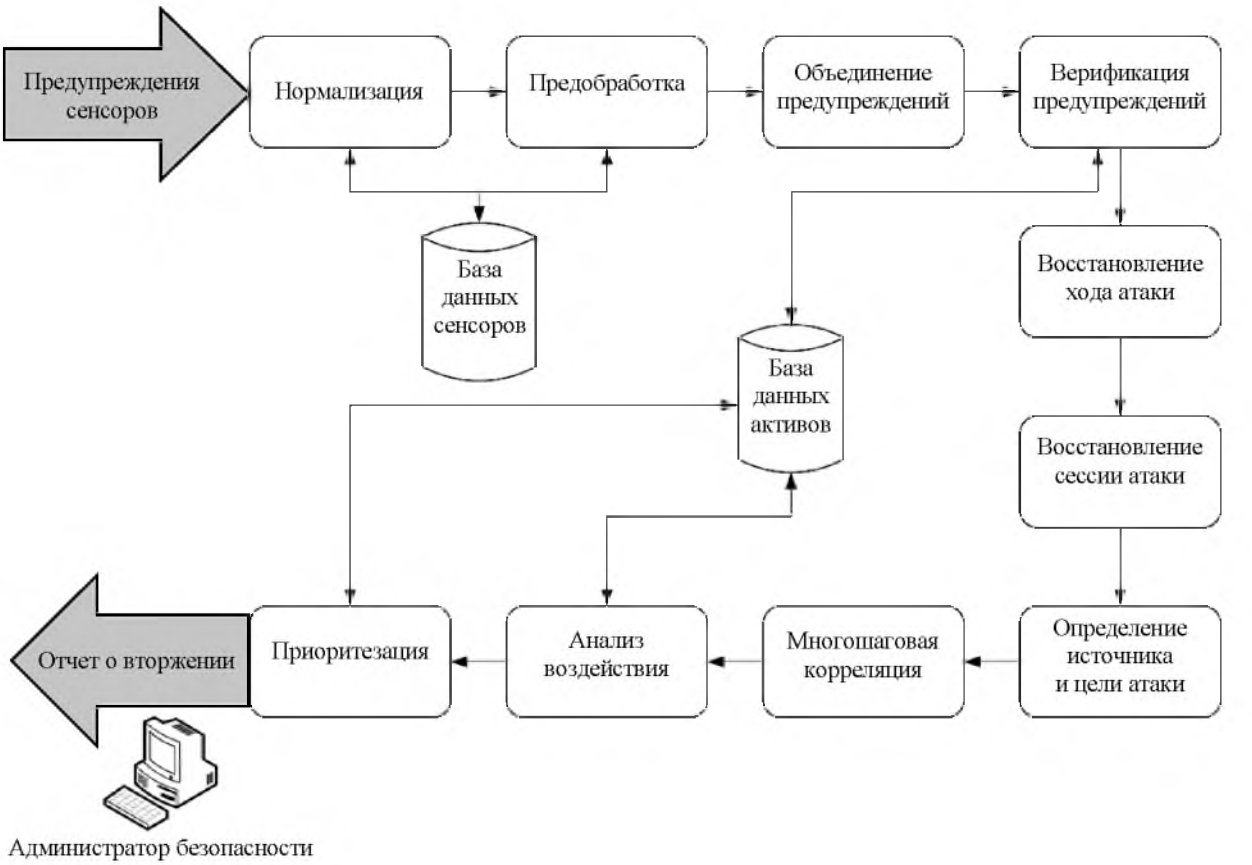

Рис. 1

К группе *фильтрации* можно отнести следующие операции:

• Сжатие (или удаление дубликатов, агрегация событий, compression).

Операция замены множества идентичных событий, которые отличаются только временем генерации, на одно-единственное событие. Результирующее событие содержит в идеальном случае число исходных событий и времена первого и последнего событий.

• Агрегация событий (aggregation).

Под агрегацией событий понимается объединение множества событий в одно новое, при этом не требуется выполнение условия идентичности исходных событий, как в случае операции сжатия.

• Фильтрация событий (filtering).

Обычно выделяют 2 режима фильтрации событий – без запоминания состояния (stateless filtering) и с запоминанием состояния (stateful filtering). В первом случае речь идет об удалении событий, обладающих определенными свойствами, из входящего потока. При этом не принимается во внимание внутреннее состояние модуля корреляции, учитываются только атрибуты непосредственно самого события. Во втором случае операция фильтрации осуществляет подавление определенных событий в зависимости от контекста (состояния) модуля корреляции событий. В этом случае оператор фильтрации часто называют оператором подавления событий. Частным случаем операции подавления является операция маскирования событий, которая также называется топологическим маскированием событий. Целью этой операции является сокрытие событий от узлов, которые уже сообщили о возможных проблемах.

• Задание порога срабатывания (thresholding).

Операция, в результате которой создается событие, если частота определенного события превышает заданное пороговое значение или, наоборот, падает ниже его.

• Композиция (composition).

Операция обрабатывает группу событий, поступающих от разных источников данных, выбирая подмножество с помощью некоторой функции сопоставления, и создает новые события на основе отобранных событий.

Группу *сопоставления* составляют следующие операции:

• Логические операции.

Под логическими операциями понимается объединение событий с помощью операторов булевой логики.

• Установление временных связей (temporal relationship).

Данная операция устанавливает взаимосвязи во времени между событиями или порядком их появления. Например, интервал времени между событиями A и B не превышает 10 мин.

• Преобразование (modification или enrich).

Операция преобразует атрибуты исходного события согласно некоторой заданной функции, например по результатам его сравнения с другим уже существующим событием.

• Выявление шаблона (или кластеризация (clustering)).

Под кластеризацией понимается операция, в результате которой создается новое событие на основе некоторого сложноструктурированного паттерна событий, построенного с использованием ранее перечисленных операций. Например, создать событие C, *если* частота событий A превышает порог 5 событий в минуту, *и* ни одного события B не было зарегистрировано в последний час.

К операциям *вывода* можно отнести следующие:

• Изменение атрибута события (escalation, modification).

В результате этой операции изменяется значение определенного параметра события (чаще всего – его приоритет).

• Обобщение (Generalization).

Под обобщением понимается соотнесение события в некоторый класс более общих событий. Несмотря на то, что новой информации в данном случае не создается, процесс выявления «причины-следствие» значительно упрощается формированием некоторых обобщающих конструкций.

• Уточнение (Specialization).

Операция по своему смыслу является противоположной операции обобщения. В результате ее выполнения некоторое более общее событие заменяется на событие, относящееся к подклассу класса исходного события. Например, если приходит сообщение о том, что некоторый заданный хост недоступен, операция «уточнение» может автоматически создать события, сообщающие о том, что службы, запущенные на заданном хосте, недоступны. Возможно, новые сообщения не содержат дополнительной информации о причине недоступности хоста, однако они могут послужить триггером для срабатывания некоторых правил, обрабатывающих недоступные сервисы.

Для обеспечения масштабируемости и эластичности непосредственно модуля корреляции в [14] определены две операции – семантический роутер (SemanticRouter) и слияние событий (EventMerger). Операция *семантический роутер* обеспечивает параллельность выполнения про-

цесса корреляции, в частности он выполняет маршрутизацию, т. е. определяет модуль-получатель входных событий, полученных от разных потоков-обработчиков очередей событий, выполняемых параллельно. Операция *слияние событий* функционирует наоборот: все входные потоки событий от разных потоков объединяются в один поток данных, упорядоченный во времени. На рис. 2 представлен принцип совместного функционирования этих двух операций.

Архитектура модуля корреляции событий. В большинстве случаев для построения компонента корреляции событий используется архитектура, управляемая событиями (event-driven architecture, EDA), которая является шаблоном архитектуры программного обеспечения. В научной литературе представлено 2 основных подхода к построению компонентов корреляции событий – *централизованная* архитектура компонента [15], [16] и *распределенная* архитектура [14], [17]–[19].

А. Централизованная архитектура компонента корреляции событий. Ключевым элементом данного типа архитектуры является центральный узел корреляции событий, который анализирует события от различных датчиков безопасности [15],

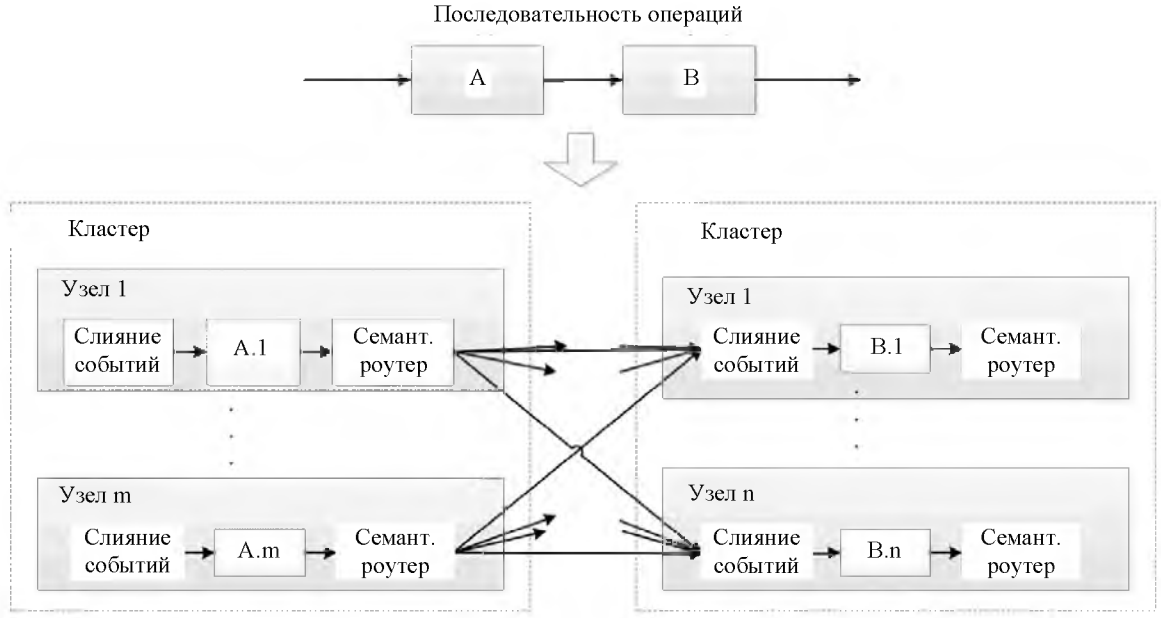

Рис. 2

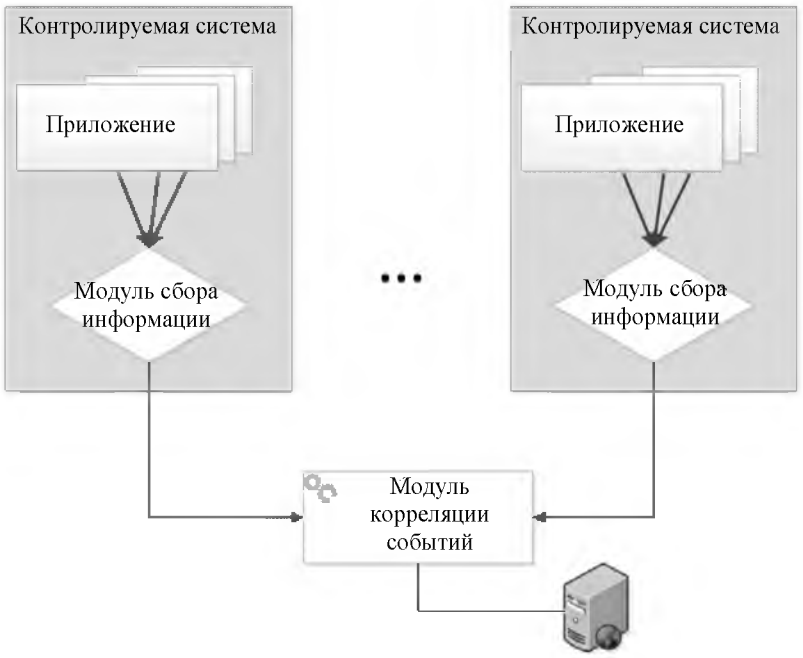

Рис. 3

[16]. На рис. 3 представлена централизованная архитектура компонента безопасности.

В. Распределенная архитектура модуля корреляции событий. Определенные недостатки централизованной архитектуры компонента корреляции событий (нагрузка на сеть, вычислительные ограничения центрального компонента корреляции событий) могут быть устранены при использовании распределенной архитектуры корреляции событий [17], [18]. Кроме того, такая архитектура способна повысить общую эффективность функционирования системы корреляции событий [18]. В общем случае распределенная архитектура модуля корреляции событий имеет 2 типа:

– полностью распределенная архитектура модуля корреляции событий;

– иерархическая архитектура модуля корреляции событий.

В первом случае все компоненты корреляции не зависят друг от друга, иерархические связи между ними отсутствуют. Главный модуль корреляции может быть выбран случайным образом, так как в некоторых случаях требуется координация взаимодействия, например для обмена информацией о новых типах атак между модулями корреляции. Однако, если по какой-то причине главный модуль выходит из строя, его легко может заменить любой другой. Схема такой архитектуры представлена на рис. 4.

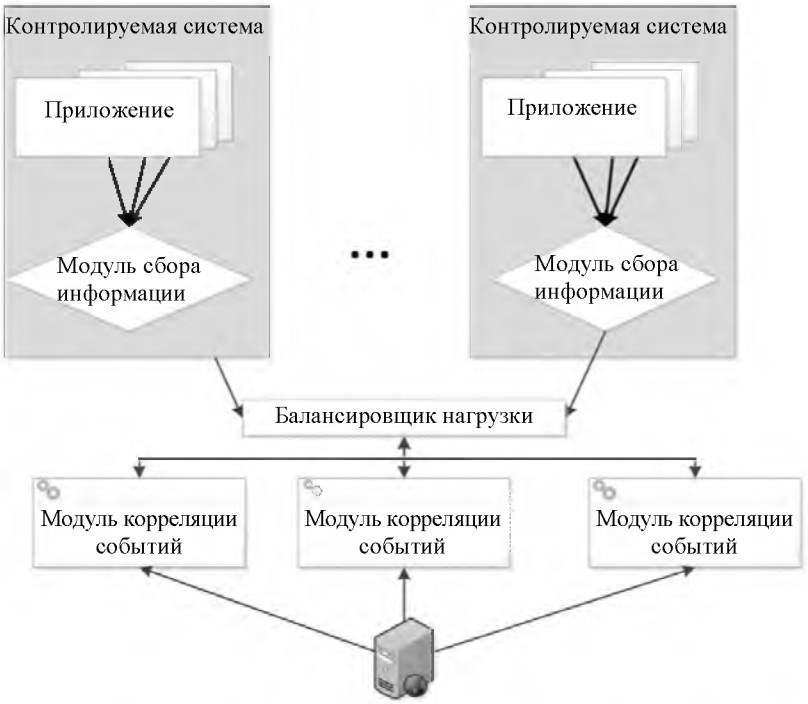

Рис. 4

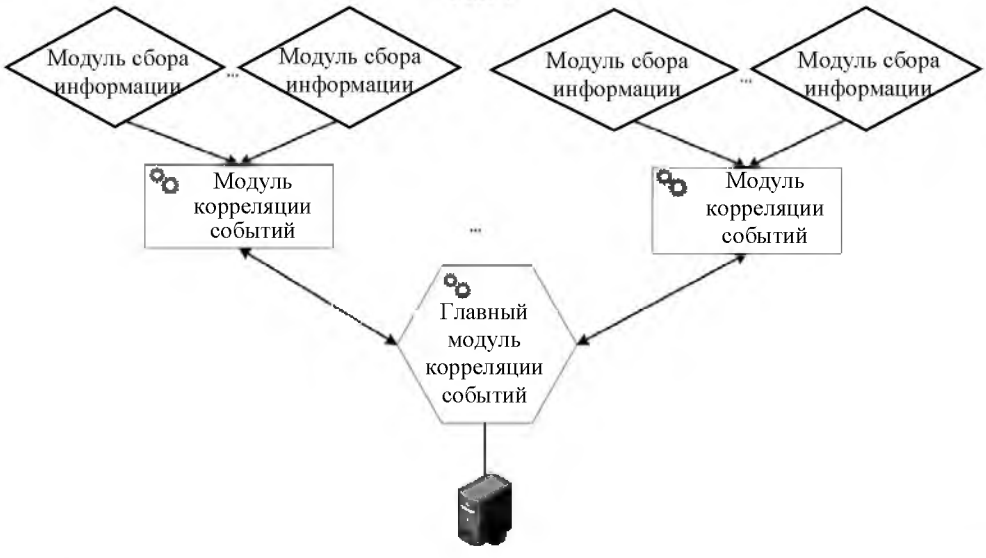

Рис. 5

Таблица 2

Тип архитектуры	Основной элемент	Достоинства	Недостатки
Централизованная	Центральный узел корреляции событий	Относительная простота алгоритма корреляции событий	• Все события передаются на центральный модуль корреляции, в результате создается дополнительная нагрузка на пропускную способность сети. • Вычислительные способности компонента корреляции ограничены, поэтому при большом потоке событий безопасности возникают временные задержки в обработке событий [18]. • Обычно не существует каких-либо механизмов защиты, установленных между центральным модулем корреляции событий и датчиками безопасности. В случае недоступности модуля корреляции событий вся система корреляции событий выходит из строя
Полностью распределенная	Может быть выбран случайным образом	Взаимозаменяемость модулей корреляции событий	Необходимость разработки дополнительного модуля корреляции, который бы занимался расстановкой приоритетов проанализированных событий и выводом этой информации пользователю
Иерархическая	Главный модуль корреляции	Повышение вычислительной эффективности модуля корреляции	Сложности при настройке двухэтапного алгоритма корреляции

Иерархическая архитектура компонента корреляции событий представлена на рис. 5 [18], [14]. Определенные сложности с использованием данного подхода к построению модуля корреляции связаны с настройкой алгоритма корреляции, поскольку он выполняется в 2 этапа – локально и централизованно (или глобально). Локальные модули корреляции выполняют корреляцию локальных событий безопасности от ограниченного числа датчиков безопасности, результаты локальной корреляции событий отсылаются модулям, принадлежащим более высоким уровням корреляции. Эти модули осуществляют дополнительный анализ данных, выполняют расчет статистик для пользователя. Центральный модуль корреляции (корневой узел в иерархии) строит граф корреляции на основе сформированных ранее результатов, он предоставляет администратору безопасности отчет об обнаруженной атаке и ее характеристиках.

Сравнительный анализ архитектур модуля корреляции событий представлен в табл. 2.

В настоящей статье рассмотрены этапы корреляции событий безопасности, определены основные операции над ними. Представлены возможные архитектурные решения построения компонента корреляции событий, рассмотрены достоинства и недостатки предложенных решений. Дальнейшие исследования связаны с изучением математических моделей и методов корреляции данных для построения эффективной системы обнаружения атак в облачной системе. Работа выполнена при поддержке Министерства образования и науки Российской Федерации в рамках работы «Организация проведения научных исследований» (Задание № 2.6113.2017/6.7), а также при финансовой поддержке гранта РФФИ № 16-07-00625.

СПИСОК ЛИТЕРАТУРЫ

1. Mirheidari S. A., Arshad S., Jalili R. Alert Correlation Algorithms: A Survey and Taxonomy Cyberspace Safety and Security // Proc. of 5th Intern. Symp., CSS 2013, Zhangjiajie, China, Nov. 13–15, 2013. Vol. 8300. P. 183–197.

2. Luckham D. The power of events: An introduction to complex event processing in distributed enterprise systems // Intern. Workshop on Rules and Rule Markup Languages for the Semantic Web. Springer Berlin Heidelberg, 2008.

3. Etzion O., Niblett P. Event Processing in Action. № ISBN: 9781935182214. Manning Publications Co., 2010.

4. Müller A. Event Correlation Engine // Computer engineering and networks laboratory / TIK Institut für Technische Informatik und Kommunikationsnetze, 2009.

5. Vasilomanolakis E. Taxonomy and survey of collaborative intrusion detection // ACM Computing Surveys (CSUR). 2015. Vol. 47, № 4. P. 55.

6. Elshoushand H. T., Osman I. M. An improved framework for intrusion alert correlation // Proc. of World Congress on Engineering 2012 (WCE 2012). Vol. 1. P. 518–524.

7. Mirheidari S. A., Arshad S., Jalili R. Alert correlation algorithms: A survey and taxonomy // Cyberspace Safety and Security. Springer Intern. Publishing, 2013. P. 183–197.

8. Dadkhah S., Shoja M. R. K., Taheri H. Alert Correlation through a Multi Components Architecture // Intern. J. of Electrical and Computer Engineering (IJECE). 2013. Vol. 3, № 4. P. 461–466.

9. López S. Z. Monitoring Control for Remote Software Maintenance // Project Report. 2010.

10. Анализ методов корреляции событий безопасности в SIEM-системах. Ч. 1 / А. В. Федорченко, Д. С. Левшун, А. А. Чечулин, И. В. Котенко // Тр. СПИИРАН. 2016. Вып. 47. С. 5–27.

11. Jakobson G., Weissman M. Alarm correlation // Network, IEEE. 1993. Vol. 7(6). P. 52–59.

12. Meira D. M. A Model For Alarm Correlation in Telecommunications Networks // PhD thesis, Federal University of Minas Gerais, 1997.

13. Ahmed T. A taxonomy on intrusion alert aggregation techniques // Biometrics and Security Technologies (ISBAST), 2014 Intern. Symp. on. IEEE, 2014. P. 244–249.

14. Vianello V. A Scalable SIEM Correlation Engine and Its Application to the Olympic Games IT Infrastructure // Intern. Conf. on Availability, Reliability and Security, Regensburg, 2013. P. 625–629.

15. Butun I., Morgera S. D., Sankar R. A survey of intrusion detection systems in wireless sensor networks // IEEE Communications Surveys & Tutorials. 2014. Vol. 16, № 1. P. 266–282.

16. Runtime correlation engine for system monitoring and testing / V. Holub, T. Parsons, P. O'Sullivan, J. Murphy // In ICAC'09: Proc. of the 6th Intern. Conf. on Autonomic computing, New York, USA, 2009. P. 43–44.

17. Donghai T., Changzhen H., Qi, Y., Jianqiao W. Hierarchical Distributed Alert Correlation Model // In Proc. of the 2009 Fifth Intern. Conf. on Information Assurance and Security. Vol. 02 (IAS'09). Vol. 2. IEEE Computer Society, Washington, DC, USA. P. 765–768.

18. Scalable Run-Time Correlation Engine for Monitoring in a Cloud Computing Environment / M. Wang, V. Holub, T. Parsons, J. Murphy, P. O'Sullivan // In Proc. of the 2010 17th IEEE Intern. Conf. and Workshops on the Engineering of Computer-Based Systems (ECBS '10). IEEE Computer Society, Washington, DC, USA. P. 29–38.

19. Streamcloud: an elastic and scalable data streaming system / V. Gulisano, R. Jimenez-Peris, M. Patino-Martinez, C. Soriente, P. Valduriez // ERCIM NEWS. 2012. Vol. 89. P. 32–33.

E. S. Novikova, Ya. A. Bekeneva, A. V. Shorov
Saint Petersburg Electrotechnical University «LETI»

SURVEY ON SECURITY EVENT CORELLATION TECHNIQUES FOR SAFETY OF CLOUD COMPUTING ENVIRONMENT

Alert correlation techniques processing events from different heterogeneous security sensors sources allows significantly improve efficiency of the modern intrusion systems used in cloud environment. Inthepaperauthorspresentresultsoftheanalysisoftheeventcorrelationprocess, consider its main stages and basic operations defined over set of the events. Possible solutions for design of event correlation component are presented, their advantages and disadvantages are discussed.

Cloud technology security, security event correlation, event operation, architecture of the event correlation component