УДК 004.056 **DOI** 10.15622/sp.2019.18.5.1182-1211

А.В. Федорченко, Е.В. Дойникова, И.В. Котенко

АВТОМАТИЗИРОВАННОЕ ОПРЕДЕЛЕНИЕ АКТИВОВ И ОЦЕНКА ИХ КРИТИЧНОСТИ ДЛЯ АНАЛИЗА ЗАЩИЩЕННОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ

Федорченко А.В., Дойникова Е.В., Котенко И.В. **Автоматизированное определение активов и оценка их критичности для анализа защищенности информационных систем.**

Аннотация. Цель исследования — разработка методики автоматизированного выделения активов информационной системы и сравнительной оценки уровня их критичности для последующей оценки защищенности анализируемой целевой инфраструктуры. Под активами в данном случае понимаются все информационно-технологические объекты целевой инфраструктуры. Размеры, разнородность, сложность взаимосвязей, распределенность и динамичность современных информационных систем затрудняют определение целевой инфраструктуры и критичности информационно-технологических активов для ее корректного функционирования. Автоматизированное и адаптивное определение состава информационно-технологических активов и связей между ними на основе выделения статичных и динамичных объектов изначально неопределенной инфраструктуры является достаточно сложной задачей. Ее предлагается решить за счет построения актуальной динамической модели отношений объектов целевой инфраструктуры с использованием разработанной методики, которая реализует подход на основе корреляции событий, происходящих в системе. Разработанная методика основана на статистическом анализе эмпирических данных о событиях в системе. Методика позволяет выделить основные типы объектов инфраструктуры, их характеристики и иерархию, основанную на частоте использования объектов, и, как следствие, отражающую их относительную критичность для функционирования системы. Для этого в работе вводятся показатели, характеризующие принадлежность свойств одному типу, совместное использование свойств, а также показатели динамичности, характеризующие вариативность свойств относительно друг друга. Результирующая модель используется для сравнительной оценки уровня критичности типов объектов системы. В работе описываются используемые входные данные и модели, а также методика определения типов и сравнения критичности активов системы. Приведены эксперименты, показывающие работоспособность методики на примере анализа журналов безопасности операционной системы Windows.

Ключевые слова: информационно-технологические активы, типы активов, критичность активов, статистический анализ данных, корреляция событий безопасности, ущерб, оценка защищенности.

1. Введение. Одним из важнейших аспектов при анализе защищенности информационной системы является выделение и оценка критичности используемых информационно-технологических активов. Практически все современные организации так или иначе используют информационные технологии для бизнеса, их объекты связаны различными каналами передачи данных и множеством семантических взаимоотношений, а системы жизнеобеспечения их активов могут быть сильно распределены. Под активами в данном случае понимаются все информационно-технологические объекты целевой инфраструк-

туры (в том числе файлы, пользователи, базы данных, приложения, сервисы, серверы, рабочие станции и др.). Инфраструктура может значительно меняться во времени, в ней могут появляться новые незарегистрированные хосты, программно-аппаратное обеспечение, уязвимости и ошибки конфигурации. В подобных условиях при достаточно масштабной архитектуре целевой информационной инфраструктуры экспертное определение активов и степени их критичности требует значительных временных и человеческих затрат.

Автоматизация выявления активов и иерархии связей между ними, а также оценка их критичности позволят более точно и детально строить модель рисков для анализа защищенности. Под критичностью в данном случае понимается важность объекта для функционирования информационной системы. Определение критичности всех информационно-технологических объектов также важно для задачи выбора контрмер, поскольку реализация мер защиты в масштабных системах со сложными взаимосвязями может привести к непредвиденному побочному ущербу для критичных активов за счет отключения или удаления вспомогательных активов. Однако данная задача является достаточно сложной. Сетевые сканеры позволяют определять различные объекты сети, такие как сервисы, используемые ими порты, хосты, на которых они развернуты, и узлы связи. Тем не менее подобные средства не дают возможности самостоятельно выделять различные типы статичных и динамичных объектов (например, таких объектов, как процессы, сессии, пользователи, привилегии, операционные системы и др.) и их иерархию, что препятствует получению актуальной динамической модели изначально неопределенной информационной системы. Вычисление и анализ показателей объектов инфраструктуры, с точки зрения авторов, является основой для автоматизированного выявления ее наиболее важных типов объектов, а также их конкретных экземпляров, то есть наиболее критичных активов информационной системы. Дальнейшее объединение динамической модели объектов инфраструктуры с условно-статическими типами данных безопасности (уязвимости, эксплойты, программно-аппаратное обеспечение, шаблоны атак, слабости и др.) позволит выявлять вредоносные, нелегитимные и аномальные объекты, проводить проактивный мониторинг состояния защищенности организации для предупреждающей выработки контрмер.

К настоящему моменту исследователями были разработаны различные методики автоматизированного определения распространения ущерба от атак в целевой инфраструктуре и критичности активов информационных систем. В частности, хорошо себя показали подходы, основанные на графах зависимостей сервисов. В том числе в [1] авто-

ры предлагают использовать зависимости между ресурсами системы для определения критичности неосновных активов и последующего выбора мер противодействия кибератакам [2, 3]. Однако они ограничиваются одним типом объектов (сервисами) и зачастую не рассматривают вопрос автоматизации выявления самих сервисов и зависимостей между ними, а также ориентированы на ручное определение наиболее важных активов и их связь с объектами инфраструктуры.

Таким образом, разработка методики автоматизированного определения активов информационной системы, иерархических связей между ними, то есть динамической модели инфраструктуры, и критичности этих активов является актуальной задачей для оценки защищенности в условиях неопределенности. В основе предлагаемой методики лежат методы корреляции событий, накапливаемых в различных журналах безопасности информационной системы. В [4] авторы описали общую идею применения методов корреляции для выявления типов объектов неопределенной инфраструктуры на основе анализа событий безопасности. В данной работе предлагается конкретная методика выявления типов объектов, а также определения их критичности. Различные виды анализа (статистический, структурный, динамический и др.) таких событий позволяют оценить критичность разных активов (объектов) и их типов с точки зрения функционирования целевой инфраструктуры и ее отдельных элементов с максимальной детализацией (от корневых до конечных объектов), которая ограничена лишь исходными данными. В этом случае подразумевается выделение иерархии типов активов и их наиболее значимых объектов.

Используемый подход корреляции событий позволяет выявить основные типы объектов инфраструктуры, их характеристики и иерархию путем анализа эмпирических данных, а также сравнивать критичность объектов на основе статических и динамических (статодинамических) показателей использования и времени жизни. Ввиду того, что подход основан на анализе эмпирических данных, он позволяет выявить только те объекты (активы) информационной системы, информация о которых наблюдалась в анализируемых событиях. Таким образом, точность работы подхода зависит от уровня детализации при журналировании событий.

Статья организована следующим образом. Во втором разделе рассмотрены релевантные исследования в области оценки критичности информационно-технологических активов и распространения ущерба от атак, автоматизированного определения конфигурации и корреляции событий безопасности. В третьем разделе описана разработанная методика выявления активов и оценки их критичности на основе предлагае-

мого подхода к корреляции событий. В четвертом разделе представлены проведенные эксперименты. В завершающем разделе приводится заключение, и описываются направления будущих исследований.

2. Релевантные работы. При анализе защищенности оценка критичности необходима для определения ущерба от кибератак, а также для определения побочного ущерба при реализации мер безопасности. В различных исследованиях критичность определяется поразному, в том числе как важность актива для функционирования системы, как стоимость актива, как стоимость замены актива. Критичность может вычисляться как с использованием качественных, так и количественных показателей. Для определения ущерба, нанесенного критичным активам, довольно широко применяются графы зависимостей сервисов. Они позволяют отследить распространение ущерба от атак в информационной системе. Их целью является определение того, как та или иная уязвимость информационной инфраструктуры, заданной в виде сервис-ориентированной архитектуры, повлияет на деятельность организации. Граф зависимостей сервисов представляет собой множество сервисов (активов) компьютерной сети, связанных между собой в соответствии с тем, как свойства безопасности одного сервиса зависят от свойств безопасности другого. Учет распространения ущерба через зависимости сервисов позволяет отрегулировать затраты на безопасность, чтобы они не превысили возможный ущерб, обосновать их, а также не упустить важные уязвимости, которые могут привести к серьезным последствиям.

В [5] показатель ущерба от атаки рассчитывается на основе карты системы (граф зависимостей, объединяющий приоритетные ресурсы), иерархии ресурсов (группировка ресурсов по типам с выделением контрмер для каждого типа) и стоимостной модели (в которой ресурсам назначаются стоимости) как сумма стоимостей узлов системной карты, на которые атакующее действие повлияло негативно. В данном случае критичность ресурса определяется его стоимостью. В [6] для определения побочного ущерба при реагировании используется дерево зависимостей между ресурсами и предлагается показатель, отражающий стоимость снижения производительности сервиса в результате потери его доступности. То есть критичностью сервиса определяется его производительность. В [7] предлагается подход к определению уровня распространяемого ущерба с использованием графа зависимостей сервисов. В данном случае критичность актива (сервиса) определяется в зависимости от его стоимости на ранговой шкале. В [8] граф зависимостей сервисов используется для определения побочного ущерба при реагировании на атаки.

Указанные исследования позволяют определить, как распространяется ущерб от кибератак в системе, или определить критичность неосновных активов организации по тому, какой ущерб в итоге приносит нарушение их безопасности. Однако они нацелены именно на сервис-ориентированные архитектуры, и их целью не является автоматизированное определение сервисов системы (или других объектов информационной системы) и связей между ними. Исследование, представленное в настоящей статье, напротив, в большей степени направлено на выделение динамических объектов неопределенной инфраструктуры и последующее сравнение их критичности на основе показателя использования. То есть в данном исследовании под критичностью актива (информационно-технологического объекта) понимается сравнительная частота его использования в системе по сравнению с другими объектами (частота обращений других объектов к данному объекту).

Рассмотрим работы, связанные с автоматическим выделением объектов информационной системы. Основным средством автоматического определения информационной инфраструктуры сети являются сетевые сканеры. Средства сканирования сети делятся на активные, такие как Nmap [9] и Nessus [10], и пассивные, такие как Wireshark [11]. Они позволяют определить топологию сети и частично выявить сервисы сети путем сканирования портов, определения использующих их сервисов и обслуживающего программно-аппаратного обеспечения. Однако эти средства не обладают достаточной степенью точности и полноты информации, и, кроме того, не ставят своей задачей определение зависимостей между ресурсами сети. Впоследствии администраторы могут вручную дополнить собранную информацию. Кроме того, информация о сервисах системы может быть уже представлена в информации о конфигурации системы [12].

Сетевые сканеры позволяют определить объекты с заранее заданной структурой и не предназначены для определения всех типов динамических объектов, таких как пользовательские и системные процессы, сессии, их иерархии, или их роли. В данном исследовании предполагается разработать методику автоматизированного определения динамических объектов неопределенной инфраструктуры на основе анализа событий, что позволит иметь актуальную динамическую структуру информационной системы, а также автоматизированно сравнить критичность объектов информационной системы.

Развернутый обзор методик автоматизированного выявления зависимостей между сервисами дан в [13]. Автор выделяет запрос доступных знаний о зависимостях [14], например в системных файлах или в специально созданном хранилище конфигурации, использование

программного кода для определения зависимостей [15], пассивные методы идентификации [16] на основе анализа взаимодействия объектов, методы на основе нейросетей [17] и методы интеллектуального анализа данных [18, 19]. В [13] также отмечается, что задача все еще не решена полностью.

Описанные работы в основном учитывают объекты одного типа (сервисы). В то время как методика, разрабатываемая в данном исследовании, направлена на автоматизированное выделение различных типов объектов. Данное исследование ближе всего к работам, использующим интеллектуальный анализ данных. Однако в [18] для выделения объектов используются только частотные показатели типов событий, в то время как в данном исследовании применяются и другие характеристики исходной информации, такие как использование и вариативность свойств и их значений, что позволяет более точно выделить типы объектов. Исследование [19] очень близко к текущему, однако авторы исследуют цепочки событий, тогда как в настоящей статье вначале определяются иерархии объектов и их типы, а потом на следующем шаге осуществляется переход на анализ цепочек событий отдельных объектов. Это позволяет точнее сравнивать цепочки событий в будущем — делить события по принадлежности к объектам и строить связи между объектами. В [20] рассматривается корреляция событий и генерация правил (шаблонов) корреляции событий по эмпирическим данным, в то время как цель настоящего исследования состоит в выделении типов объектов сети. Методы, связанные с корреляцией событий, подробнее рассмотрены в следующем разделе для выделения наиболее подходящих с точки зрения поставленной в исследовании задачи.

Поскольку для решения двух описанных выше задач предлагается использовать аппарат корреляции, рассмотрим исследования, посвященные корреляции событий [21]. Изначально корреляция данных применялась в рамках систем обнаружения вторжений для выявления связей между сетевыми событиями с целью их агрегации и последующего обнаружения атак [22-24]. Однако корреляция событий, происходящих в информационной системе, кроме выявления инцидентов и предупреждений безопасности, может применяться для решения различных задач безопасности, в том числе для определения взаимосвязей между разнородной информацией безопасности, для группировки низкоуровневых событий в высокоуровневые мета-события, для выявления типов объектов информационной системы и связей между ними [25].

Наиболее популярным и простым в реализации методом корреляции является метод, основанный на правилах [26, 27]. Главный недостаток этого метода заключается в больших временных затратах на

задание правил администратором. Эффективность данного метода напрямую связана с квалификацией администратора. Многие методы, в том числе основанные на шаблонах (сценариях) [26, 28], графах [28-30], конечных автоматах [31], и другие, используют различные модели для отображения событий и связей между ними, но могут быть реализованы с использованием правил [32].

В настоящее время более перспективными представляются подходы к корреляции, основанные на самообучении [33, 34], такие как байесовские сети [26, 35], вероятностное исчисление событий [36-38], иммунные сети [35], искусственные нейронные сети [35, 39] и другие. Преимуществом данных подходов является возможность независимой (безусловной) корреляции событий с минимальными ручными настройками. Тем не менее для построения моделей обучения необходим предварительный анализ данных, который сложно автоматизировать. Кроме того, предъявляются требования к оценке адекватности и качества моделей, и исходная обучающая выборка должна быть достаточно полной.

Хотя рассмотренные методы применялись для обнаружения вторжений, а не для решения задачи выявления типов активов информационной системы и связей между ними, предполагается, что подобный анализ данных также применим и для решения поставленной в исследовании задачи. В настоящей статье показывается, что динамический анализ событий, происходящих в системе, и вычисление стато-динамических показателей, в том числе частотных характеристик типов событий, вариативности значений свойств, парного использования и вариативности свойств, помогает выявить основные источники событий (объекты) и иерархические взаимосвязи между типами объектов. Разработанная методика корреляции на основе динамического анализа данных для автоматизированного выделения активов подробно описана в следующем разделе.

3. Методика выделения типов объектов и определение критичности активов. Разработанная методика предназначена для автоматизированного определения активов информационной системы, иерархических связей между ними, то есть динамической модели инфраструктуры и критичности этих активов на основе их использования в информационной системе. Методика объединяет несколько последовательных этапов:

1. Сбор и предварительная обработка входных данных. На данном этапе производится сбор данных о событиях из журналов информационной системы и их нормализация для корректного проведения дальнейшего анализа.

2. Выделение типов объектов. На этом этапе применяется статистический анализ собранных данных для выявления типов характери-

стик, типов объектов целевой инфраструктуры, иерархических связей между ними на основе статодинамических показателей, в том числе частотных характеристик типов событий, парного использования и вариативности свойств.

3. Определение критичности объектов и их выделенных типов на основе показателя использования свойств и их значений.

Обобщенная схема первых трех этапов методики, которым поставлены в соответствие уровни, приведена на рисунке 1.

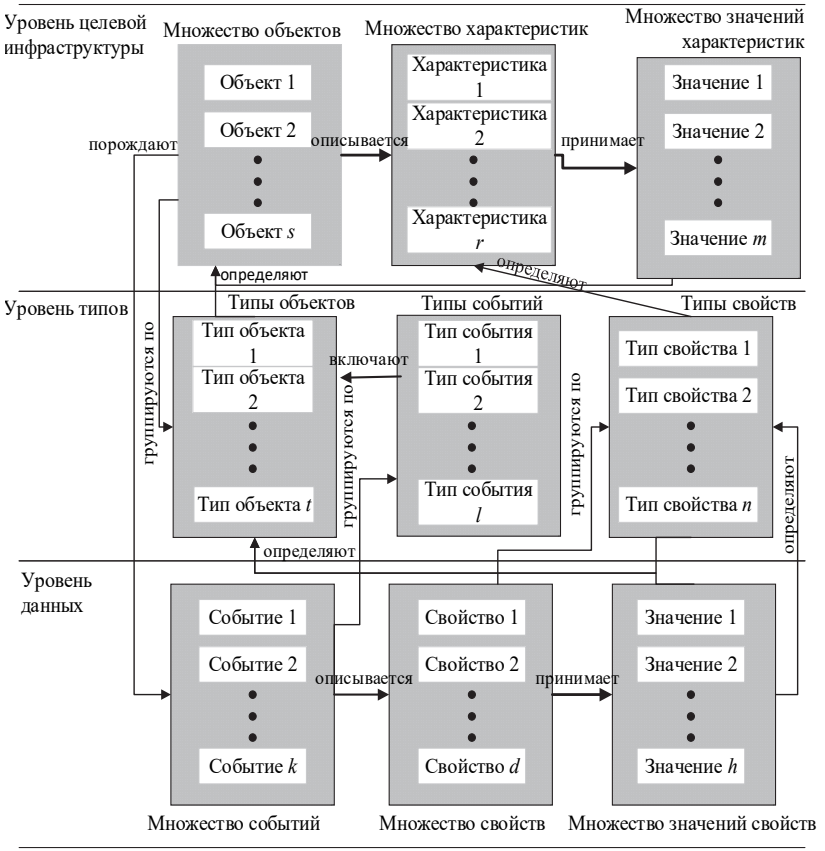

Рис. 1. Схема методики выделения типов объектов и их иерархии

Стрелками, сходящимися в одну точку, показана необходимость анализа нескольких сущностей для определения текущей (начало стрелки). Подробнее каждый этап описан в соответствующем подразделе.

3.1. Сбор и обработка входных данных. Предположим, что изначально анализируемая инфраструктура не определена. В качестве исходных данных используем множество событий, представленных в журналах объектов информационной системы. Рассмотрим подробнее исходные данные для выполнения процесса корреляции и основные понятия, используемые в рамках разработанной методики.

Исходными данными для выполнения статистического анализа является множество событий безопасности E журнала L:

$$E^L = \{e_1, e_2, ..., e_n\}.$$

Под событием понимается факт либо результат выполнения какого-либо действия на любом из его этапов: попытка (событие отказа), начало действия (событие старта), промежуточный результат действия (события, продолжительные по времени), конечный результат выполнения действия (завершено корректно, завершено с ошибкой). Каждое событие e состоит из множества свойств p и их значений v, которыми описывается определенное действие [40]:

$$e = \{p_i, v_i\},$$

где p_i — свойство события, $p_i \in P^e$, $P^e \subset P$, P — множество всех свойств событий:

$$P = \{p_1, p_2, ..., p_d\},$$

где d — общее количество свойств; v_i — значение свойства события, $v_i \in V^{p_i}$, V^{p_i} — множество значений свойства p_i:

$$V^{p_i} = \{v_1, v_2, ..., v_h\},$$

h — количество возможных значений свойства p_i; $V^{p_i} \subset V$, V — множество всех значений всех свойств:

$$V = \{V^{p1}, V^{p2}, ..., V^{pd}\}$$

В каждый момент времени данные множества являются конечными и не пустыми, хотя в течение времени они могут изменять свою мощность: $P \neq \emptyset$, $V^{p_i} \neq \emptyset$, $\forall i \in \{1, 2, ..., d\}$.

Для корректного проведения дальнейшего анализа свойства событий и их значения должны быть нормализованы. В противном случае качество типизации свойств и выявление типов объектов могут быть существенно снижены. В рамках данного исследования не рассматривается задача автоматизации этого этапа, и задача выполняется на основе экспертной оценки.

Собранные данные делятся по типам принимаемых ими значений на категориальные и количественные. В зависимости от типа значений на следующем этапе используются разные подходы корреляции.

3.2. Выделение типов объектов и их иерархии. В работе для определения информационных объектов целевой инфраструктуры и их характеристик используются методы корреляции. Для описания подхода к корреляции событий на основе статистического анализа с целью выявления основных активов системы и иерархии связей между ними необходимо задать модель неопределенной инфраструктуры и определить ее связь с исходными данными.

При дискретно-непрерывном наблюдении (анализе истории наблюдений) целевая инфраструктура I, состав и архитектура которой изначально не определены, состоит из s информационных объектов O [40]:

$$O^I = \{o_1, o_2, ..., o_s\},$$

имеющих некоторую продолжительность жизни. Их состояние описывается с помощью одной или нескольких характеристик x и их значений:

$$o = (x_i, v_i),$$

где $o \in O^I$, $x_i \in X$, $X = \{x_1, x_2, ..., x_r\}$, $|X| \geq 1$, r — общее числа возможных характеристик объектов; $v_i \in V^{x_i}$, V^{x_i} — множество возможных значений характеристики x_i:

$$V^{x_i} = \{v_1, v_2, ..., v_k\},$$

k — количество возможных значений характеристики x_i; $V^{x_i} \subset V$, V — множество всех значений всех характеристик:

$$V = \{V^{x1}, V^{x2}, ..., V^{xk}\}.$$

Информационные объекты обязательно связаны друг с другом отношениями принадлежности. То есть каждый объект обязательно является частью более высокоуровневого объекта и (или) содержит в себе более низкоуровневые объекты. Также связь между объектами определяется за счет их непосредственного взаимодействия друг с другом. Это позволяет сформировать иерархию объектов.

Предполагается, что коллекция характеристик X^o, которыми описывается информационный объект *o*, однозначно задает тип информационного объекта *ot*, то есть каждая характеристика *x* из множества *X* содержится только в одном типе информационного объекта *ot* [40]. Типом объекта может быть «процесс» (сервис), «файл», «сессия» (сетевая, пользовательская), «сенсор», «хост» (сетевое оборудование, персональный компьютер), и другие.

Заключительным элементом модели неопределенной инфраструктуры *I* является множество отношений *R* между объектами *O*. Однако определение данного множества выходит за рамки настоящей работы и будет рассмотрено в дальнейших исследованиях.

Таким образом, модель *M* неопределенной инфраструктуры *I* состоит из множества информационных объектов *O*, а также множеств их типов *OT* и отношений *OR*:

$$M^I = \langle O, OT, OR \rangle .$$

Исходя из вышеописанного, методика выделения типов объектов и их иерархии должна преобразовывать исходные данные процесса корреляции к множествам объектов и их типов модели неопределенной инфраструктуры:

$$\{E, P, V\} \xrightarrow{f} \{O, OT\} .$$

Для выделения типа данных (смыслового) используется метод корреляции данных на основе динамического анализа событий. Этап выделения типов объектов и их иерархии включает следующие шаги: (1) выделение типов свойств событий на основе анализа их возможных значений; (2) выделение типов объектов на основе типов свойств; (3) определение иерархии типов объектов на основе показателя общего использования.

На первом шаге множество свойств $P = \{p_1, p_2, ..., p_m\}$, где m — общее количество свойств всех событий, которые могут принимать

множество значений $V = \{V^{p1}, V^{p2}, ..., V^{pm}\}$, анализируется для выделения множества типов свойств *pt*, которое определяется как:

$$PT = \{pt_1, pt_2, ..., pt_n\},$$

где *n* — общее количество типов.

Множества *P* и *PT* связаны отношением принадлежности, то есть каждый тип *pt* представляет подмножество свойств *p*:

$$pt = \{p_1, p_2, ..., p_n\},$$

где *l* — количество свойств, описывающих тип *pt*.

Для определения принадлежности свойств определенному типу данных мы предлагаем использовать показатель парной вариативности их значений. Формула для вычисления данного показателя отличается для статичных и динамичных свойств. Под статичным понимается свойство, значение которого в среднем меняется намного реже, чем значение динамичного свойства (например, имя процесса), под динамичным понимается свойство, значение которого меняется несколько раз за то время, пока значение статичного свойства остается неизменным (например, идентификатор процесса). Для определения динамичности свойств на предмет изменения значений анализируются все значения всех свойств.

В роли показателей динамичности мы предлагаем использовать: (1) *PV* — абсолютное значение вариативности свойства, выраженное количеством возможных наблюдаемых (принимаемых) значений, и (2) *EPV* — средняя использование значений для каждого свойства, выраженная отношением количества событий, в которых наблюдается данное свойство, к общему числу возможных значений *PV:*

$$EPV(p) = \left|E^p\right| / PV . \quad (1)$$

В случае статичных свойств, показатель принадлежности свойств одному типу задается следующим образом:

$$\mu_{pt} = \frac{\left|V^{p_i} \cap V^{p_j}\right|}{\left|V^{p_i} \cup V^{p_j}\right|}, \quad (2)$$

где *V* — множество значений, принимаемых свойством, то есть принадлежность свойств p_i и p_j одному типу определяется эквивалентностью их значений.

В случае динамичных свойств типы определятся путем анализа времени жизни и вариативности значений свойств на этом отрезке.

На втором шаге выполняется выделение групп свойств, предположительно характеризующих типы объектов на основе парного использования, и последующего уточнения этих групп за счет использования корреляции количественных значений и определения однотипных свойств внутри выделенных групп. Под парным использованием свойств p_i и p_j подразумевается отношение случаев их совместного использования к общему количеству их использований в событиях. Показатель парного использования между свойствами определяется по формуле:

$$\mu_{ot} = \frac{\left| E^{p_i} \cap E^{p_j} \right|}{\left| E^{p_i} \cup E^{p_j} \right|}, \qquad (3)$$

Для дальнейшего уточнения выделенных типов объектов дополнительно используется определение однотипных свойств. Кроме того, возможно применение методов ранговой корреляции. Для этого значения свойства ранжируются по показателю использования в порядке убывания (отдельно для каждого свойства) и сопоставляются соответствующее его порядковому номеру.

На третьем шаге для определения типов объектов и их уровня в иерархии вычисляется значение общего использования их свойств (чем выше использование, тем выше объект в иерархии). Оно определяется как отношение количества использований свойства в событиях к общему количеству всех событий. Отношения между типами объектов определяются видами содержащих их событий. Можно выделить два основных вида: изменение состояния объекта и взаимодействие между объектами. Наличие взаимодействия определяет наличие связи между типами объектов.

3.3. Оценка критичности активов. Под критичностью актива (информационно-технологического объекта) будем понимать сравнительную частоту его использования в системе по сравнению с другими объектами.

Определение критичности объектов реализуется на основе выполнения следующих шагов:

1) Определение места типа объекта в иерархии типов объектов.

2) Определение относительной критичности объекта в соответствии с его местом в иерархии и по отношению к однотипным объектам.

Критичные объекты инфраструктуры предлагается определять как наиболее используемые объекты (показатель общего использования). На

первом шаге для определения места типа объекта в иерархии вычисляется показатель общего использования: $\mu=E/n$, где E — количество использования свойств типа объекта в событиях, n — общее количество событий. Типы объектов упорядочиваются сверху вниз, от максимально используемого к минимально используемому. Место l_{ot}, или уровень типа объекта *ot* в иерархии определяется как его порядковый номер.

На втором шаге критичность Cr определяется нормализацией уровня типа объекта в иерархии по общему количеству уровней l: $Cr = l_{ot}/l$. То есть имеется в виду относительная критичность информационно-технологического объекта определенного типа в системе.

Вышеописанные шаги позволяют определить уровень критичности относительно остальных типов (чем выше объект в иерархии, тем выше его использование, и, соответственно, тем выше его относительная критичность), то есть критичность типа объекта (актива) определяется относительным показателем его общего использования.

Необходимо отметить, что предлагаемый подход позволяет автоматизированно определить относительную критичность объектов инфраструктуры, однако требует дальнейшего усовершенствования и уточнения, в том числе за счет определения зависимостей между объектами и связанным с этим влиянием на критичность объектов. Кроме того, в данном случае не учитываются редко используемые критичные объекты, что также планируется рассмотреть в будущих исследованиях.

В дальнейшем планируется связать рассматриваемую динамическую инфраструктуру со статической, в том числе со стандартизированным определением программно-аппаратного обеспечения и уязвимостей. Это позволит связать предлагаемый показатель критичности объектов с вероятностью их компрометации и, следовательно, определять оценки рисков для объектов информационной системы.

4. Реализация и эксперименты. Реализация разработанной методики проводилась с использованием языка Python 3.5 и библиотек numpy, scipy и pandas, а результаты были визуализированы с помощью GraphViz и модулей matplotlib, pyplot и seaborn.

Для выполнения экспериментов использовалась вычислительная платформа с одним 6-ядерным процессором Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60GHz. и 64 GB RAM.

4.1. Исходные данные для анализа. В условиях неопределенной целевой инфраструктуры множества событий E, свойств P и значений V изначально являются пустыми. Пользуясь историческими записями журналов событий либо накапливая информацию в реальном времени, следует первоначально определить множества свойств P и их возможных значений V.

Для проверки теоретических положений раздела 3 на данный момент эксперименты проводились в рамках одного хоста, в дальнейшем планируется провести тестирование для различных видов информационных систем. Исходными данными для проведения экспериментов являлись события системного журнала безопасности хоста под управлением ОС Windows 8. Подсистема журналирования операционной системы была настроена на сбор максимального количества типов событий безопасности. Анализируемый журнал обладает следующими характеристиками:

– количество событий ~ 6700000;

– количество оригинальных типов событий — 44 (из более чем 250 заявленных разработчиками для журнала безопасности [41]);

– количество свойств событий (включая свойства с единственным нулевым наблюдаемым значением) — 111;

– количество пар свойств, совместно наблюдаемых в событиях — 1379;

– количество возможных значений свойств без объединения повторяющихся вариантов между разными свойствами — 241024;

– общее количество возможных значений всех свойств — 213040;

– размер данных журнала — 7ГБ в формате XML, 1,25ГБ в формате CSV;

– время записи журнала — 36 дней;

– точность привязки событий к масштабу времени — десятые доли микросекунд.

На основе представленных характеристик анализируемого журнала можно сделать ряд предварительных выводов, а именно: (1) только 11% пар свойств (пункт 4) из возможного гипотетического количества пар наблюдаются в событиях совместно; (2) 12% возможных значений свойств наблюдаются как минимум в 2х свойствах (пункт 5 и 6); (3) количество событий в несколько десятков раз преобладает над количеством уникальных значений, что характеризует полноту журналирования поведения объектов инфраструктуры.

4.2. Предварительная обработка данных. Этап предварительной обработки исходной информации подразумевает нормализацию свойств событий, а также их значений для корректного проведения дальнейшего анализа (см. раздел 3). В настоящее время данная задача выполнена на основе экспертной оценки, а результаты нормализации описаны ниже.

Типы событий 5156 «The Windows Filtering Platform has allowed a connection» и 5157 «The Windows Filtering Platform has blocked a connection» содержат свойство «ProcessID», наименование которого

отличается от прочих свойств идентификатора процесса «ProcessId». Подобная малозначительная неточность может существенно снизить как качество типизации свойств, так и выявление типов объектов. С другой стороны, значительное отклонение структур свойств, описывающих информационные объекты, в данных типах событий может привести к неточным результатам анализа парного использования при однозначном сопоставлении указанных наименований. Возможным решением данной проблемы является приведение наименований свойств к однообразному регистру, а дальнейший анализ должен учитывать возможные пропуски в исходных данных при несогласованных структурах свойств типов событий. В описываемом случае наименование свойства «ProcessID» было приведено к «ProcessId».

В ходе анализа было установлено, что для 6 свойств событий наблюдалось единственное семантически-нулевое значение '-'. Такие свойства были исключены из дальнейшего анализа. Также указанное значение наблюдалось в 23 свойствах, имеющих прочие возможные значения, а само значение исключалось из списка возможных.

С нашей точки зрения, подобный пропуск в исходной информации может свидетельствовать о неверно составленной структуре типов событий, в которых данные свойства используются. Также для ряда свойств наблюдались единственные нулевые значения, представленные в разных форматах, например: «S-1-0-0» и «{00000000-0000-0000-0000-000000000000}». Однако данные значения несут определенную семантическую нагрузку, поэтому в дальнейшем анализе свойства, в которых они наблюдаются, были использованы.

4.3. Эксперименты. В рамках экспериментов, показатели, предложенные в разделе 3, использовались для выделения типов свойств (парная вариативность), типов объектов (парное использование) и их иерархии (общее использование свойств). Кроме того, были проведены эксперименты по определению динамичных свойств, позволяющих выделить критичные активы.

Первоочередным этапом анализа свойств событий является определение их показателей динамичности (предпосылки к выполнению данного этапа будут описаны ниже).

Как было определено в разделе 3, в роли таких показателей предлагается использовать *PV* (абсолютное значение вариативности свойства, выраженное количеством возможных наблюдаемых значений) и *EPV* (среднее использование значений для каждого свойства, определяемую по формуле 1).

Пример вычисленных показателей динамичности для некоторых наиболее показательных свойств представлен на рисунке 2.

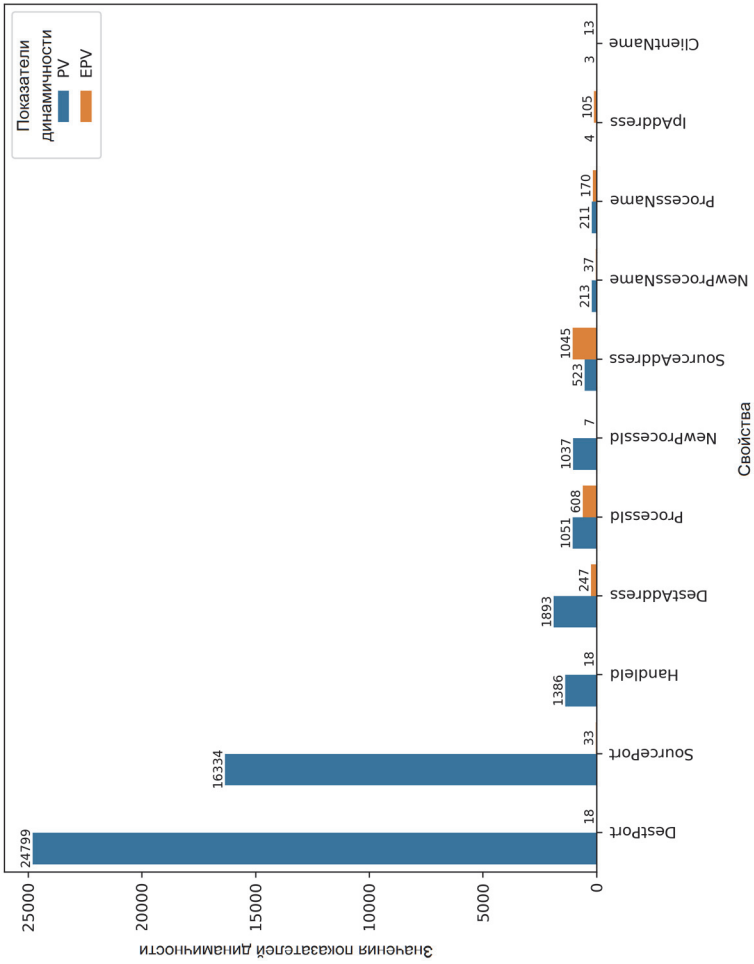

Рис. 2. Гистограмма показателей динамичности свойств событий

Труды СПИИРАН. 2019. Том 18 № 5. ISSN 2078-9181 (печ.)
ISSN 2078-9599 (онлайн), www.proceedings.spiiras.nw.ru

На данном рисунке видно, что наиболее динамичными являются свойства «DestPort» и «SourcePort». Однако такая оценка является достаточно грубой, поскольку динамичность свойства должна быть оценена во времени. Действительно, с уменьшением показателя абсолютной вариативности для некоторых свойств наблюдается рост среднего использования их значений, тогда как остальные свойства сохраняют тенденцию со значительным преобладанием вариативности.

Для подобных ситуаций следует гипотетически определить граничные значения показателей динамичности для:

(1) абсолютно статичного свойства, имеющего одно возможное значение;

(2) абсолютно динамичного свойства, все наблюдаемые значения которого являются уникальными в его пределах.

Указанные случаи описаны в таблице 1. Результаты измерений показателей динамичности некоторых свойств в течение времени (за 10 первых дней записей анализируемого журнала по дням) представлены на рисунке 3. На рисунке в каждый момент времени измеряются описанные показатели динамичности за весь прошедший период от начала наблюдений.

Таблица 1. Граничные значения

Показатель / Вид свойства	PV	EPV
Статичное свойство	$PV \to 1$	$EPV \to \left\|E^p\right\|$
Динамичное свойство	$PV \to \left\|E^p\right\|$	$EPV \to 1$

Очевидно, что для большинства свойств показатель вариативности за рассматриваемый промежуток времени достигает определенных предельных значений. Данный факт не свидетельствует о достижении возможного максимума для последующих измерений, однако по данным графикам возможно оценить динамичность свойств по отношению друг к другу.

Явным недостатком предлагаемой оценки динамичности свойств являются абсолютные величины показателей. Другими словами, адекватно сравнивать динамичность свойств на основе данных показателей целесообразно при относительно высоком парном использовании свойств. В противном случае необходимо ввести относительную величину динамичности, что является одним из направлений дальнейших исследований.

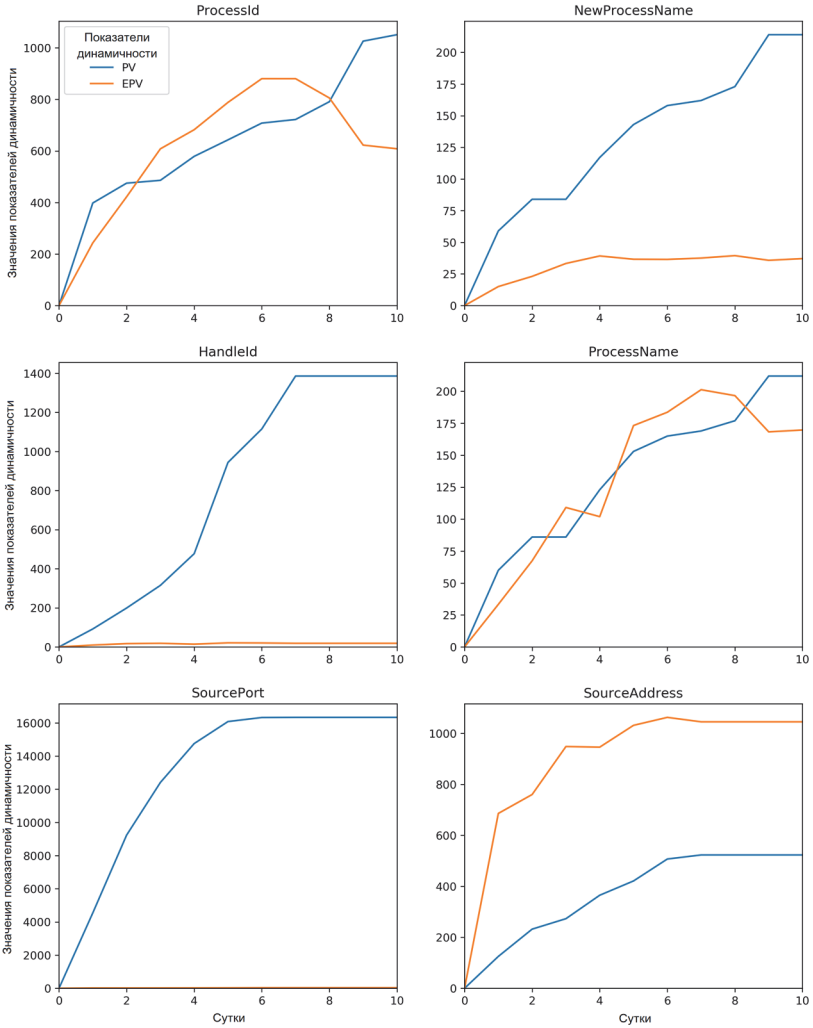

Рис. 3. Показатели динамичности свойств событий безопасности в течение времени за наблюдаемый период

4.4. Определение типов свойств. Задача выделения типов свойств событий, представленная в разделе 3, необходима для формирования списка характеристик информационных объектов, тогда как набор подобных характеристик будет в дальнейшем определять тип объекта описания. Для группировки свойств событий в типы используется показатель общей парной вариативности (формула 2).

Труды СПИИРАН. 2019. Том 18 № 5. ISSN 2078-9181 (печ.)
ISSN 2078-9599 (онлайн), www.proceedings.spiiras.nw.ru

Эксперименты показали, что не все значения показателей общей парной вариативности являются достаточными для автоматизированного принятия решения о принадлежности свойств к одному типу, что является ошибкой первого рода при исключении данной связи и, соответственно, нарушении верной типизации.

Причинами подобных ошибок могут являться: (1) высокая общая вариативность; (2) недостаточная общая вариативность одного из свойств. Однако более критичным случаем при типизации свойств является слияние двух разных типов событий, что является ошибкой второго рода при определении принадлежности разнотипных свойств к одному типу. В частности, такие ошибки могут быть вызваны схожестью форматов описания для разнотипных свойств.

Исходя из того, что вариативность является показателем динамичности свойства, преодоление описанных ошибок возможно при динамическом рассмотрении парной вариативности. Данная операция подразумевает сравнение возможных значений свойств в коротком интервале. Исходя из гипотезы, что при наблюдении определенного значения события в текущий момент времени, в окрестностях данного момента, объект, описываемый с помощью данного значения, является «живым». Таким образом, динамика изменения значений свойств событий определена временем жизни таких объектов. Парное сравнение значений свойств с привязкой к масштабу времени позволяет более точно определить парную вариативность свойств, и минимизировать появление описанных ошибок. В свою очередь, жизненный цикл объекта, описываемого с помощью конкретного значения свойства, может быть вычислен по диаграмме его наблюдения во времени.

Пример подобного графика представлен на рисунке 4.

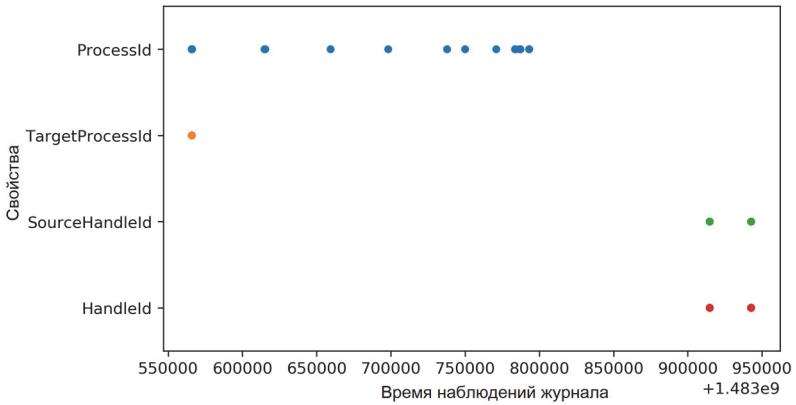

Рис. 4. Пример жизненных циклов объектов ОС Windows

По оси ординат на рисунке отображены свойства, которые принимают определенное значение (в данном случае «0x350»), а по оси абсцисс задано время наблюдения анализируемого журнала (36 дней) в относительных величинах.

Стоит отметить основное отличие между статичными и динамичными свойствами событий: статичные свойства характеризуют объект в общем масштабе времени, тогда как динамичное свойство характеризует объект лишь во время его жизни. Иными словами, при завершении жизненного цикла объекта динамическое свойство, которое его характеризовало, в последующих жизненных циклах будет принимать отличное от первоначального значение.

4.5. Определение типов объектов и их иерархии. Как было отмечено ранее, определение типов объектов по парному использованию свойств, которыми они описываются, основывается на гипотезе, что одинаковое использование свойств событий свидетельствует об описании характеристик одного или нескольких типов объектов одного уровня. Выдвигаемая гипотеза предварительно подтверждается проведенными экспериментами, показавшими, что отдельные группы свойств имеют равный либо очень близкий по значению показатель использования.

В ходе эксперимента по обнаружению разнотипных связей между свойствами на основе показателя их совместного использования в событиях было сформировано 18 групп свойств, а общее количество сгруппированных свойств равно 60. Стоит отметить, что свойства из одной группы используются в типах событий исключительно совместно [49].

Наиболее значимые и интерпретируемые типы объектов вместе со свойствами, которые их определяют, представлены на рисунке 5. На данном рисунке в каждой группе событий также указан показатель общего использования составляющих группу свойств, а сами группы упорядочены по данному показателю. Результаты проведенного эксперимента подтверждают гипотезу, согласно которой критичность типа объекта (актива) в ряде случаев определяется относительным показателем его общего использования (раздел 3).

Отдельно стоит рассмотреть объекты нулевого типа, которые встречаются во всех записях событий. Строго говоря, данные свойства определяют нулевой тип событий, так как даже такое пустое событие, как семантическая атомарная единица информации [42], имеет определенный заголовок со служебной информацией, например тип события «EventID» или задача «Task» [40].

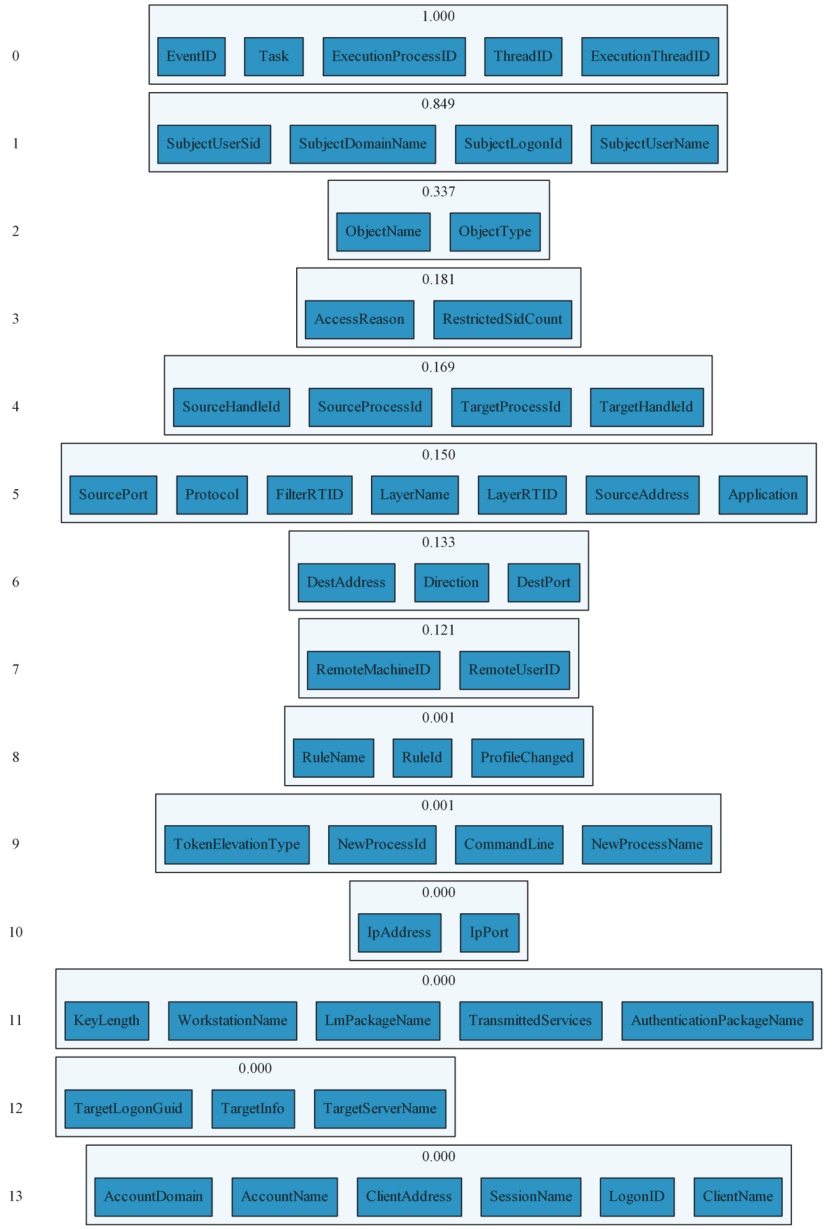

Рис. 5. Группы свойств ОС Windows, определяющие типы объектов, их уровень в иерархии и роли в типах событий

Рассмотрев сформированные группы объектов следует заключить, что большинство их них (за исключением нулевого уровня) являются достаточно семантически выраженными. Однако, представленные группы в действительности разбивают множество свойств не только по типам объектов, но и по их ролям в типах событий. Так, группы свойств 1 и 2 выделяют роли объектов «Субъект» и «Объект» соответственно.

В то же время группа 4 явно должна быть поделена на 2 роли «Источник» и «Цель» для одного типа объектов. Вероятно, более глубокая типизация объектов, например на основе вычисления коэффициентов корреляции, позволит избежать подобных неточностей. Данное направление будет рассматриваться в дальнейших исследованиях.

Отметим, что уровень в иерархии типов объектов, представленной на рисунке 5, определяет относительную критичность соответствующих активов по отношению к другим типам объектов (активов). В частности, группа характеристик 1, означающих соответствующие типы объектов (активов) «Домен» (*SubjectDomainName*), «Пользователь» (*SubjectUserName, SubjectUserSid*) и «Сессия» (*SubjectLogonId*) явно преобладает над остальными группам (кроме нулевой). Действительно, перечисленные типы объектов являются более критичными, чем типы: «Системный указатель» (*SourceHandleId, TargetHandleId*). «Процесс» (*SourceProcessId, TargetProcessId*), «Сетевой источник» (*DestPort, DestAddress, Direction*) и другие, что частично подтверждает выдвинутую гипотезу. Однако некоторые группы (7, 13), имеющие низкий показатель общего использования, явно не отражают уровень критичности соответствующих активов. Данный факт обусловлен редкостью наблюдения событий, связанных с объектами данных групп и корректная обработка подобных случаев является дальнейшим направлением исследований.

5. Заключение. В работе рассмотрен предлагаемый авторами подход к корреляции событий безопасности для оценки критичности активов (объектов) целевой инфраструктуры и их типов с точки зрения функционирования инфраструктуры и ее отдельных элементов с максимальной детализацией, которая ограничена лишь исходными данными. Описаны основные элементы подхода, в том числе исходные данные, модель неопределенной инфраструктуры, методика выделения типов объектов инфраструктуры и методика сравнения их критичности, а также применяемые на разных этапах показатели. Применение предложенного подхода показано на примерах и экспериментах, доказывающих выдвинутые теоретические предположения. Предлагается использовать разработанную методику оценки критичности в рамках оценки защищенности для определения возможного ущерба от реализации угроз безопасности. В дальнейшем планируется формализовать предлагаемый

подход, используя теорию нечетких множеств, усовершенствовать его за счет применения интервального анализа характеристик событий и расширить для выявления разных типов отношений между объектами, а также связать с методиками оценки защищенности. Кроме того, планируется развить методику оценки критичности, для учета случаев, когда критичность объекта не определяется тем, насколько часто его используют. Например, это касается генераторов ключей шифрования. Также планируется рассмотреть возможность улучшения методики оценки критичности за счет учета политики доступа организации.

Литература

1. *Kotenko I., Doynikova E., Chechulin A.* Security metrics based on attack graphs for th olympic games scenario // 2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. 2014. pp. 561–568.
2. *Kotenko I., Doynikova E.* Countermeasure selection in SIEM systems based on the integrated complex of security metrics // 2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. 2015. pp. 567–574.
3. *Doynikova E., Kotenko I.* Countermeasure selection based on the attack and service dependency graphs for security incident management // International Conference on Risks and Security of Internet and Systems. 2015. pp. 107–124.
4. *Kotenko I., Fedorchenko A., Saenko I., Kushnerevich A.* Parallelization of security event correlation based on accounting of event type links // 2018 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP). 2018. pp. 462–469.
5. *Balepin I., Maltsev S., Rowe J., Levitt K.* Using specification-based intrusion detection for automated response // International Workshop on Recent Advances in Intrusion Detection. 2003. pp. 136–154.
6. *Jahnke M., Thul C., Martini P.* Graph based metrics for intrusion response measures in computer networks // 32nd IEEE Conference on Local Computer Networks (LCN 2007) 2007. pp. 1035–1042.
7. *Kheir N. et al.* Cost Evaluation for Intrusion Response Using Dependency Graphs // 2009 International Conference on Network and Service Security. 2009. pp. 1–6.
8. *Shameli-Sendi A., Louafi H., He W., Cheriet M.* Dynamic Optimal Countermeasure Selection for Intrusion Response System // IEEE Transactions on Dependable and Secure Computing. 2018. vol. 15. no. 5. pp. 755–770.
9. NMap reference guide. URL: http://nmap.org/book/man.html (дата обращения: 02.07.2018).
10. Nessus vulnerability scanner. URL: http://www.tenable.com/products/nessus-vulnerability-scanner (дата обращения: 02.07.2018).
11. Wireshark vulnerability scanner. URL: https://www.wireshark.org (дата обращения: 02.07.2018).
12. *Clemm A., Bansal A.* Auto-Discovery at the Network and Service Management Layer // International Symposium on Integrated Network Management. 2003. pp. 365–378.
13. *Hanemann A.* Automated IT Service Fault Diagnosis Based on Event Correlation Techniques: Diss // Imu. 2007. 343 p.
14. *Steinder M., Sethi A.S.* A survey of fault localization techniques in computer networks // Science of Computer Programming. 2004. vol. 53. no. 2. pp. 165–194.
15. *Bagchi S., Kar G., Hellerstein J.* Dependency Analysis in Distributed Systems using Fault Injection: Application to Problem Determination in an e-commerce Environment // 12th International Workshop on Distributed Systems (DSOM'2001). 2001.

16. *Agarwal M.K. et al.* Mining Activity Data for Dynamic Dependency Discovery in e-Business Systems // IEEE Transactions on Network and Service Management. 2004. vol. 1. no. 2. pp. 49–58.
17. *Ensel C.* A scalable approach to automated service dependency modeling in heterogeneous environments // Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference. 2001. pp. 128–139.
18. *Tuchs K.D., Jobmann K.* Intelligent search for correlated alarm events in databases // 2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium. 2001. pp. 285–288.
19. *Motahari-Nezhad H.R., Saint-Paul R., Casati F., Benatallah B.* Event correlation for process discovery from web service interaction logs // The VLDB Journal – The International Journal on Very Large Data Bases. vol. 20. no. 3. pp. 417–444.
20. *Hellerstein J.L., Ma S., Perng C.S.* Discovering actionable patterns in event data // IBM Systems Journal. 2002. vol. 41. no. 3. pp. 475–493.
21. *Федорченко А.В., Левшун Д.С., Чечулин А.А., Котенко И.В.* Анализ методов корреляции событий безопасности в SIEM-системах. Часть 1 // Труды СПИИРАН. 2016. Т. 4. № 47. С. 5–27.
22. *Artikis A. et al.* Scalable Proactive Event-Driven Decision Making // IEEE Technology and Society Magazine. 2014. vol. 33. no. 3. pp. 35–41.
23. *Raju B.K., Geethakumari G.* Event correlation in cloud: a forensic perspective // Computing. 2016. vol. 98. no. 11. pp. 1203–1224.
24. *Calyam P. et al.* Topology-Aware Correlated Network Anomaly Event Detection and Diagnosis // Journal of Network and Systems Management. 2014. vol. 22. № 2. pp. 208–234.
25. *Alevizos E. et al.* The Complex Event Recognition Group // ACM SIGMOD Record. 2018. vol. 47. no. 2. pp. 61–66.
26. *Sadoddin R., Ghorbani A.* Alert Correlation Survey: Framework and Techniques // Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services. 2006. pp. 37.
27. *Limmer T., Dressler F.* Survey of Event Correlation Techniques for Attack Detection in Early Warning Systems // University of Erlangen, Dept. of Computer Science, Technical Report. 2008.
28. *Xu D., Ning P.* Correlation analysis of intrusion alerts // North Carolina State University. 2006.
29. *Michelioudakis E., Artikis A., Paliouras G.* Semi-Supervised Online Structure Learning for Composite Event Recognition // Machine Learning. 2018. pp. 1–26.
30. *Han Y., Zhu M., Liu C.* A Service-Oriented Approach to Modeling and Reusing Event Correlations // 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). 2018. vol. 1. pp. 498–507.
31. *Ghorbani A.A., Lu W., Tavallaee M.* Network Intrusion Detection and Prevention: Concepts and Techniques // Springer Science & Business Media. 2009. vol. 47. 223 p.
32. *Papataxiarhis V., Hadjiefthymiades S.* Event Correlation and Forecasting over Multivariate Streaming Sensor Data // arXiv preprint arXiv:1803.05636. 2018.
33. *Астахова Л.В., Цимбол В.И.* Применение самообучающейся системы корреляции событий информационной безопасности на основе нечеткой логики при автоматизации систем менеджмента информационной безопасности // Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. 2016. Т. 16. № 1. 5 с.
34. *Tiwari R.R., Singh A.K., Singh V.* Self-learning SIEM system using association rule mining // Journal of Advanced Database Management & Systems. 2015. vol. 2. № 2. pp. 10–23.
35. *Gurer D.W., Khan I., Ogier R., Keffer R.* An Artificial Intelligence Approach to Network Fault Management // SRI International. 1996. vol. 86.

36. *Skarlatidis A., Paliouras G., Artikis A., Vouros G.A.* Probabilistic Event Calculus for Event Recognition // ACM Transactions on Computational Logic (TOCL). 2015. vol. 16. no. 2. pp. 1–37.
37. *Alevizos E., Skarlatidis A., Artikis A., Paliouras G.* Probabilistic Complex Event Recognition: A Survey // ACM Computing Surveys. 2017. vol. 50. no. 5. pp. 71.
38. *Marvasti M.A., Poghosyan A.V., Harutyunyan A.N., Grigoryan N.M.* Statistical Normalcy Determination based on Data Categorization // VMware Technical Journal 2014. vol. 3. no. 1. pp. 43–55.
39. *Zhou J., Guo A., Celler B., Su S.* Fault detection and identification spanning multiple processes by integrating PCA with neural network // Applied Soft Computing. 2014. vol. 14. pp. 4–11.
40. *Федорченко А.В.* Анализ свойств событий безопасности для обнаружения информационных объектов и их типов в неопределенных инфраструктурах // Известия высших учебных заведений. Приборостроение. 2018. Вып. 61(11). С. 997–1004.
41. Windows Security Log Events. URL: https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/Default.aspx (дата обращения: 22.11.2018).
42. *Fedorchenko A., Kotenko I., El Baz D.* Correlation of security events based on the analysis of structures of event types // 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). 2017. vol. 1. pp. 270–276.

Федорченко Андрей Владимирович — младший научный сотрудник, лаборатория проблем компьютерной безопасности, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН). Область научных интересов: безопасность компьютерных сетей, обнаружение вторжений, вредоносные программы. Число научных публикаций — 40. fedorchenko@comsec.spb.ru; 14-я линия В.О., 39, 199178, Санкт-Петербург, Российская Федерация; р.т.: +7(812)328-7181.

Дойникова Елена Владимировна — канд. техн. наук, научный сотрудник, лаборатория проблем компьютерной безопасности, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук(СПИИРАН). Область научных интересов: безопасность компьютерных сетей, методы анализа рисков компьютерных сетей, управление информационными рисками. Число научных публикаций — 71. doynikova@comsec.spb.ru; 14-я линия В.О., 39, 199178, Санкт-Петербург, Российская Федерация; р.т.: +7(812)328-7181; факс: +7(812)328-4450.

Котенко Игорь Витальевич — д-р техн. наук, профессор, заведующий лабораторией, лаборатория проблем компьютерной безопасности, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН). Область научных интересов: безопасность компьютерных сетей, в том числе управление политиками безопасности, разграничение прав доступа, аутентификация, анализ защищенности, обнаружение компьютерных атак, межсетевые экраны, защита от вирусов и сетевых червей, анализ и верификация протоколов безопасности и систем защиты информации, защита программного обеспечения от взлома и управление цифровыми правами, технологии моделирования и визуализации для противодействия кибер-терроризму. Число научных публикаций — 450. ivkote@comsec.spb.ru; 14-я линия В.О., 39, 199178, Санкт-Петербург, Российская Федерация; р.т.: +7(812)328-7181; факс: +7(812)328-4450.

Поддержка исследований. Работа выполнена при финансовой поддержке РФФИ (проекты № 19-07-01246, 16-29-09482, 18-37-20047, 18-07-01488 и 18-29-22034), стипендии Президента РФ № СП-751.2018.5 и бюджетной темы АААА-А16-116033110102-5.

DOI 10.15622/sp.2019.18.5.1182-1211

A.V. FEDORCHENKO, E.V. DOYNIKOVA, I.V. KOTENKO

AUTOMATED DETECTION OF ASSETS AND CALCULATION OF THEIR CRITICALITY FOR THE ANALYSIS OF INFORMATION SYSTEM SECURITY

Fedorchenko A.V., Doynikova E.V., Kotenko I.V. **Automated Detection of Assets and Calculation of their Criticality for the Analysis of Information System Security.**

Abstract. The research aims to develop the technique for an automated detection of information system assets and comparative assessment of their criticality for farther security analysis of the target infrastructure. The assets are all information and technology objects of the target infrastructure. The size, heterogeneity, complexity of interconnections, distribution and constant modification of the modern information systems complicate this task. An automated and adaptive determination of information and technology assets and connections between them based on the determination of the static and dynamic objects of the initially uncertain infrastructure is rather challenging problem. The paper proposes dynamic model of connections between objects of the target infrastructure and the technique for its building based on the event correlation approach. The developed technique is based on the statistical analysis of the empirical data on the system events. The technique allows determining main types of analysed infrastructure, their characteristics and hierarchy. The hierarchy is constructed considering the frequency of objects use, and as the result represents their relative criticality for the system operation. For the listed goals the indexes are introduced that determine belonging of properties to the same type, joint use of the properties, as well as dynamic indexes that characterize the variability of properties relative to each other. The resulting model is used for the initial comparative assessment of criticality for the system objects. The paper describes the input data, the developed models and proposed technique for the assets detection and comparison of their criticality. The experiments that demonstrate an application of the developed technique on the example of analyzing security logs of Windows operating system are provided.

Keywords: Assets, Asset Types, Asset Criticality, Statistical Data Analysis, Security Event Correlation, Impact, Security Assessment.

Fedorchenko Andrey Vladimirovich — junior researcher, Laboratory of Computer Security Problems, St. Petersburg Institute for Informatics and Automation of Russian Academy of Scienc-es (SPIIRAS). Research interests: computer network security, intrusion detection, malware. The number of publications — 40. fedorchenko@comsec.spb.ru; 39, 14-th Line V.O., 199178, St. Petersburg, Russian Federation; office phone: +7(812)328-7181.

Doynikova Elena Vladimirovna — Ph.D., researcher, Laboratory of Computer Security Problems, St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences (SPIIRAS). Research interests: computer network security, risk analysis methods for computer networks, information security risk management. The number of publications — 71. elenadoynikova@mail.ru; 39, 14-th Line V.O., 199178, St. Petersburg, Russian Federation; office phone: +7(812)328-7181; fax: +7(812)328-4450.

Kotenko Igor Vitalievich — Ph.D., Dr.Sci., Professor, Head of Laboratory, Laboratory of Computer Security Problems, St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences (SPIIRAS). Research interests: computer network security, including security policy management, access control, authentication, network security analysis, intrusion detection, firewalls, deception systems, malware protection, verification of security systems, digital right management, modeling, simulation and visualization technologies for counteraction to cyber terrorism. The number of publications — 450. ivkote@comsec.spb.ru;

39, 14-th Line V.O., 199178, St. Petersburg, Russian Federation; office phone: +7(812)328-7181; fax: +7(812)328-4450.

Acknowledgements. This work was partially supported by grants of RFBR (projects No. 19-07-01246, 16-29-09482, 18-37-20047, 18-07-01488 and 18-29-22034), grant of the President of the Russian Federation SP-751.2018.5 and by the budget (the project No. AAAA-A16-116033110102-5).

References

1. Kotenko I., Doynikova E., Chechulin A. Security metrics based on attack graphs for th olympic games scenario. 2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. 2014. pp. 561–568.
2. Kotenko I., Doynikova E. Countermeasure selection in SIEM systems based on the integrated complex of security metrics. 2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. 2015. pp. 567–574.
3. Doynikova E., Kotenko I. Countermeasure selection based on the attack and service dependency graphs for security incident management. International Conference on Risks and Security of Internet and Systems. 2015. pp. 107–124.
4. Kotenko I., Fedorchenko A., Saenko I., Kushnerevich A. Parallelization of security event correlation based on accounting of event type links. 2018 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP). 2018. pp. 462–469.
5. Balepin I., Maltsev S., Rowe J., Levitt K. Using specification-based intrusion detection for automated response. International Workshop on Recent Advances in Intrusion Detection. 2003. pp. 136–154.
6. Jahnke M., Thul C., Martini P. Graph based metrics for intrusion response measures in computer networks. 32nd IEEE Conference on Local Computer Networks (LCN 2007) 2007. pp. 1035–1042.
7. Kheir N. et al. Cost Evaluation for Intrusion Response Using Dependency Graphs. 2009 International Conference on Network and Service Security. 2009. pp. 1–6.
8. Shameli-Sendi A., Louafi H., He W., Cheriet M. Dynamic Optimal Countermeasure Selection for Intrusion Response System. *IEEE Transactions on Dependable and Secure Computing*. 2018. vol. 15. no. 5. pp. 755–770.
9. NMap reference guide. URL: http://nmap.org/book/man.html (дата обращения: 02.07.2018).
10. Nessus vulnerability scanner. Available at: http://www.tenable.com/products/nessus-vulnerability-scanner (accessed: 02.07.2018).
11. Wireshark vulnerability scanner. Available at: https://www.wireshark.org (accessed: 02.07.2018).
12. Clemm A., Bansal A. Auto-Discovery at the Network and Service Management Layer. International Symposium on Integrated Network Management. 2003. pp. 365–378.
13. Hanemann A. Automated IT Service Fault Diagnosis Based on Event Correlation Techniques: Diss. Imu. 2007. 343 p.
14. Steinder M., Sethi A.S. A survey of fault localization techniques in computer networks. *Science of Computer Programming.* 2004. vol. 53. no. 2. pp. 165–194.
15. Bagchi S., Kar G., Hellerstein J. Dependency Analysis in Distributed Systems using Fault Injection: Application to Problem Determination in an e-commerce Environment. 12th International Workshop on Distributed Systems (DSOM'2001). 2001.
16. Agarwal M.K. et al. Mining Activity Data for Dynamic Dependency Discovery in e-Business Systems. *IEEE Transactions on Network and Service Management*. 2004. vol. 1. no. 2. pp. 49–58.

17. Ensel C. A scalable approach to automated service dependency modeling in heterogeneous environments. Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference. 2001. pp. 128–139.
18. Tuchs K.D., Jobmann K. Intelligent search for correlated alarm events in databases. 2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium. 2001. pp. 285–288.
19. Motahari-Nezhad H.R., Saint-Paul R., Casati F., Benatallah B. Event correlation for process discovery from web service interaction logs. *The VLDB Journal – The International Journal on Very Large Data Bases*. vol. 20. no. 3. pp. 417–444.
20. Hellerstein J.L., Ma S., Perng C.S. Discovering actionable patterns in event data. *IBM Systems Journal.* 2002. vol. 41. no. 3. pp. 475–493.
21. Fedorchenko A.V., Levshun D.S., Chechulin A.A., Kotenko I.V. [An Analysis of Security Event Correlation Techniques in Siem-Systems. Part 1]. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2016. Issue 4. vol. 47. pp. 5–27 (In Russ.).
22. Artikis A. et al. Scalable Proactive Event-Driven Decision Making. *IEEE Technology and Society Magazine.* 2014. vol. 33. no. 3. pp. 35–41.
23. Raju B.K., Geethakumari G. Event correlation in cloud: a forensic perspective. *Computing.* 2016. vol. 98. no. 11. pp. 1203–1224.
24. Calyam P. et al. Topology-Aware Correlated Network Anomaly Event Detection and Diagnosis. *Journal of Network and Systems Management.* 2014. vol. 22. № 2. pp. 208–234.
25. Alevizos E. et al. The Complex Event Recognition Group. ACM SIGMOD Record. 2018. vol. 47. no. 2. pp. 61–66.
26. Sadoddin R., Ghorbani A. Alert Correlation Survey: Framework and Techniques. Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services. 2006. pp. 37.
27. Limmer T., Dressler F. Survey of Event Correlation Techniques for Attack Detection in Early Warning Systems. University of Erlangen, Dept. of Computer Science, Technical Report. 2008.
28. Xu D., Ning P. Correlation analysis of intrusion alerts. North Carolina State University. 2006.
29. Michelioudakis E., Artikis A., Paliouras G. Semi-Supervised Online Structure Learning for Composite Event Recognition. *Machine Learning*. 2018. pp. 1–26.
30. Han Y., Zhu M., Liu C. A Service-Oriented Approach to Modeling and Reusing Event Correlations. 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). 2018. vol. 1. pp. 498–507.
31. Ghorbani A.A., Lu W., Tavallaee M. Network Intrusion Detection and Prevention: Concepts and Techniques. Springer Science & Business Media. 2009. vol. 47. 223 p.
32. Papataxiarhis V., Hadjiefthymiades S. Event Correlation and Forecasting over Multivariate Streaming Sensor Data. arXiv preprint arXiv:1803.05636. 2018.
33. Astahova L.V., Cimbol V.I. [Application of the self-learning system of information security events correlation based on the fuzzy logic for the information security management systems]. *Vestnik YUUrGU. Seriya: Kompyuternye tekhnologii, upravlenie, radioehlektronika – Bulletin of the South Ural State University. Series: Computer Technologies, Automatic Control & Radioelectronics.* 2016. vol. 1 5 p. (In Russ.).
34. Tiwari R.R., Singh A.K., Singh V. Self-learning SIEM system using association rule mining. *Journal of Advanced Database Management & Systems.* 2015. vol. 2. № 2. pp. 10–23.
35. Gurer D.W., Khan I., Ogier R., Keffer R. An Artificial Intelligence Approach to Network Fault Management. SRI International. 1996. vol. 86.

36. Skarlatidis A., Paliouras G., Artikis A., Vouros G.A. Probabilistic Event Calculus for Event Recognition. *ACM Transactions on Computational Logic (TOCL).* 2015. vol. 16. no. 2. pp. 1–37.
37. Alevizos E., Skarlatidis A., Artikis A., Paliouras G. Probabilistic Complex Event Recognition: A Survey. *ACM Computing Surveys*. 2017. vol. 50. no. 5. pp. 71.
38. Marvasti M.A., Poghosyan A.V., Harutyunyan A.N., Grigoryan N.M. Statistical Normalcy Determination based on Data Categorization*. VMware Technical Journal.* 2014. vol. 3. no. 1. pp. 43–55.
39. Zhou J., Guo A., Celler B., Su S. Fault detection and identification spanning multiple processes by integrating PCA with neural network. *Applied Soft Computing*. 2014. vol. 14. pp. 4–11.
40. Fedorchenko A.V. An [Analysis of Security Event Properties for Detection of the Information Objects and Their Types in Uncertain Infrastructures]. *Izvestiya vysshih uchebnyh zavedenij. Priborostroenie – Journal of Instrument Engineering*. 2018. vol. 61(11). pp. 997–1004. (In Russ.).
41. Windows Security Log Events. Available at: https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/Default.aspx (accessed: 22.11.2018).
42. Fedorchenko A., Kotenko I., El Baz D. Correlation of security events based on the analysis of structures of event types. 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). 2017. vol. 1. pp. 270–276.