Федеральное государственное бюджетное учреждение науки
Санкт-Петербургский институт информатики и автоматизации
Российской академии наук

На правах рукописи

Ушаков Игорь Александрович

ОБНАРУЖЕНИЕ ИНСАЙДЕРОВ В КОМПЬЮТЕРНЫХ СЕТЯХ НА ОСНОВЕ КОМБИНИРОВАНИЯ ЭКСПЕРТНЫХ ПРАВИЛ, МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ И ОБРАБОТКИ БОЛЬШИХ ДАННЫХ

Специальность 05.13.19 – «Методы и системы защиты информации, информационная безопасность»

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор технических наук, профессор
Котенко Игорь Витальевич

Санкт-Петербург – 2020

Содержание

Введение

Актуальность темы диссертации. Современную жизнь сложно представить без информационного взаимодействия, затрагивающего как отдельных членов общества, так и крупные организации, в том числе реализующие интересы целого государства. Помимо очевидных получаемых преимуществ такое взаимодействие несет также ряд существенных недостатков [31]. Так, передача информации по сети подвергает ее триаде угроз информационной безопасности: конфиденциальности, целостности и доступности. При этом безопасность информации должна быть обеспечена как при её передаче через открытые сети, так и внутри компьютерной сети (КС), под которой прежде всего понимается корпоративная компьютерная сеть. Однако к информации в КС, в особенности носящей критически важное значение, могут иметь доступ внутренние сотрудники, часть из которых изначально обладает такими полномочиями, входящими в круг их должностных обязанностей. Таким образом, возникает проблема противодействия атакам на КС, как случайным, так и злонамеренным, производимым в том числе внутренними сотрудниками организации.

Существуют различные способы противодействия инсайдерской деятельности на разных этапах – до самой атаки, во время ее проведения и после атаки. У каждого из способов есть свои достоинства и недостатки, однако важным является тот факт, что информация может устаревать и ее ценность, соответственно, уменьшаться. Следовательно, оказание п***о***зднего противодействия инсайдерским атакам может оказаться бессмысленным, поскольку информация к этому времени уже будет скомпрометирована и использована третьими лицами. Так, в случае нарушения целостности данных или предоставления неправомочного доступа к данным факт обнаружения одного из подобных нарушений будет иметь для организации существенно меньший эффект, чем недопущение инсайдерской атаки в целом. Следовательно, востребованным является именно недопущение инсайдерской атаки, что может быть достигнуто путем обнаружения инсайдеров

до момента проведения самой атаки. После обнаружения инсайдеров, естественно, предполагается их нейтрализация. Нейтрализация инсайдеров может производиться либо автоматически – программными средствами, либо вручную – экспертами по информационной безопасности [7].

Согласно последним исследованиям [79], весь IP-трафик и число устройств, подключенных к сети Интернет, утроятся за следующие 5 лет. Считается, что это произойдет вследствие развития сервисов и услуг, предоставляемых телекоммуникационными компаниями. При этом особую популярность набирают: облачные сервисы в виде Platform-as-a-Service (PaaS) и Software-as-a-Service (SaaS); решения для хранения данных; аналитические системы; решения для ведения бизнеса и прогнозирования рисков; рекомендательные системы. Расширение областей применения сетевых технологий означает децентрализацию сетевой инфраструктуры в целом как в плане хранения данных, так и в плане получаемого доступа к этой инфраструктуре [6]. Это усложняет решение инженерных задач, стоящих перед специалистами информационной безопасности, поскольку становится труднее контролировать все аспекты сетевой безопасности при защите критически важных данных от угроз, исходящих как из внешней сети, так и изнутри, от самих участников сети [33, 36, 76, 79, 96, 113, 115].

Таким образом, основная сложность обнаружения инсайдеров в КС напрямую следует из современных тенденций развития информационных технологий, неразрывно связанных с постоянным увеличением параметров сетевого трафика: его объема; скорости генерации; количества источников и получателей трафика; количества логических потоков, не связанных со своими целями и задачами; увеличения уровня гетерогенности данных и др. [73]

Все это приводит к существенному усложнению анализаторов трафика, поскольку далеко не все существующие системы способны справляться с такими большими объемами и сложностью, в то время как инсайдеры скрывают свои действия в общем потоке действий законных пользователей. Кроме того, современные инсайдерские атаки являются комплексными и используют множество способов реализации и множество векторов для получения

несанкционированного доступа и компрометации информационных объектов во внутренней КС [26].

Таким образом, *основное противоречие предметной области* заключается в следующем: с одной стороны, необходимо повышение точности обнаружения инсайдеров, поскольку их атаки постоянно усложняются и комплексируются, сетевой трафик атак становится менее различим из-за роста объема всего трафика в КС, а сами инсайдеры маскируют свои действия под законные; с другой стороны, существующие модели, методики и алгоритмы обнаружения инсайдеров не обладают необходимой эффективностью работы, поскольку или имеют высокий риск пропуска инсайдера (ошибка II рода), или, наоборот, – риск отнесения к инсайдеру законного пользователя (ошибка I рода). Возможной причиной порождения данного противоречия является некоторая субъективность, присущая всем вводимым критериям инсайдерской деятельности. Так, например, часть пользователей, определенных как инсайдеры, могли просто выполнять ряд ошибочных действий: неверно ввести свой пароль, ошибочно скачать документ или отправить документ на неверный адрес, подключить чужое устройство и т.п.

Разрешение указанного противоречия может лежать в плоскости применения высокоэффективных специализированных технологий обработки сетевого трафика для сферы информационной безопасности, а также в сочетании существующих и новых способов анализа и обнаружения инсайдерской деятельности. Все это может быть достигнуто следующим образом.

Во-первых, тенденция роста популярности появления решений для работы с большими данными позволяет предположить гипотетическую востребованность данной технологии для разрешения выявленного выше основного противоречия предметной области. Так, с появлением инструментов для разработки систем, использующих концепцию больших данных [91, 93, 100, 111, 116, 130, 134, 142], встает вопрос об использовании обработки больших данных для информационной безопасности и, в частности, систем мониторинга безопасности. Становится все сложнее обнаруживать потенциальные угрозы безопасности. Пропускная способность современных систем мониторинга и предупреждения сетевых атак

перестает удовлетворять требованиям постоянно разрастающихся сетей: в связи с большим количеством поступающего трафика и низкой скоростью его обработки результаты такого анализа получаются неактуальными и не отражают реального состояния сети [74]. Используя новые и эффективные технологии для агрегации и хранения больших объемов данных, а также для организации работы системы обнаружения злоумышленника, можно добиться нужных результатов, а именно получить достаточный уровень контроля над ситуацией в КС.

Также важно учитывать, что не все модели представления данных в достаточной степени адаптированы к своевременной обработке больших объемов информации и событий. Специфика задач кибербезопасности заключается в необходимости применения новых моделей баз данных и использовании методов обработки больших данных для анализа трафика компьютерных сетей.

Во-вторых, существующим и хорошо зарекомендовавшим себя подходом к обнаружению инсайдеров (учитывая сложность строгой формализации критериев обнаружения последних и их возможности к сокрытию своих действий) является использование алгоритмов на основе правил, составленных экспертами с учетом собственного накопленного опыта и существующих «best practices» (экспертных правил).

И, в-третьих, учет комплексности проводимых инсайдерами атак, а также их распределенности: по сети (например, атака на целый ряд не связанных хостов), по объектам (например, попытка доступа к частям документа с целью сбора общей критической массы конфиденциальной информации), по времени (например, последовательность событий, связанных длительным промежутком времени), – дает возможность предположить востребованность применения методов машинного обучения, позволяющих учитывать множество, на первый взгляд, трудно связанных друг с другом параметров [150].

Все вышесказанное предполагает применение для обнаружения инсайдеров в КС подхода, основанного на использовании экспертных правил, методов машинного обучения и обработки больших данных. Этим обуславливается актуальность темы диссертационного исследования.

Степень разработанности темы. Проблеме существования инсайдерской деятельности в КС было посвящено большое количество работ как отечественных ученых (П.Д. Зегжды, И.В. Котенко, А.В. Лукацкого, А.А. Молдовяна, В.Ю. Осипова, И.Б. Саенко и др.), так и зарубежных (S. Bellovin, C. Cheh, M. Collins, F. Kammüller, Y. Shuang-Hua, X. Wang и др.). Однако, несмотря на сделанный учеными существенный задел, проблема обнаружения инсайдеров в КС не может считаться разрешенной и требует проведения новых исследований, что и осуществлено в данной работе.

Цели и задачи. Основной целью диссертационной работы является повышение защищенности КС за счет усовершенствования моделей, алгоритмов и методики обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и способов обработки больших данных.

Для достижения данной цели в диссертационной работе поставлены и решены следующие задачи:

1. анализ существующих подходов к обнаружению инсайдеров в КС, моделей, методик и алгоритмов обнаружения инсайдеров в КС на основе методов машинного обучения и обработки больших данных;
2. разработка модели представления больших данных об инсайдерских атаках в формате NoSQL (включая модель инсайдера);
3. разработка алгоритма обнаружения инсайдеров в КС, основанного на экспертных правилах;
4. разработка модели и алгоритмов комбинированного применения экспертных правил и методов машинного обучения в интересах обнаружения инсайдерских атак;
5. разработка методики обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных;
6. построение архитектуры и реализация программного комплекса системы обнаружения инсайдеров в КС на базе предложенной методики, настройка

алгоритма на основе методов машинного обучения с помощью набора данных, характеризующих действия инсайдеров по заданному множеству сценариев атак, и экспериментальная оценка разработанной методики системы обнаружения инсайдеров в КС.

Объектом исследования являются КС, в которых возможно наличие инсайдеров и атаки инсайдеров на КС.

Предметом исследования являются модели, методики и алгоритмы обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных.

Разработка модельно-методического аппарата для обнаружения инсайдеров в КС на основе комбинированного использования экспертных правил, методов машинного обучения и обработки больших данных определяет **научную задачу исследования**.

Теоретическая и практическая значимость работы. Теоретическая значимость диссертационной работы определяется ее вкладом в дальнейшее развитие теории и методов информационной безопасности, что проявляется в следующих аспектах: расширены классы атрибутов, необходимых для обнаружения инсайдеров; предложен новый подход к комбинированию двух классов алгоритмов, основанных на экспертных правилах и на методах машинного обучения, для решения задачи обнаружения инсайдеров в КС; методика реализует последовательность операций, необходимых для решения задачи обнаружения инсайдеров, основывается на модели в формате NoSQL, алгоритмах, основанных на экспертных правилах, а также алгоритмах, основанных на методах машинного обучения; архитектура реализует совокупность компонентов, их взаимосвязь, процедуру их выполнения и программную реализацию для решения задачи обнаружения инсайдеров в КС; архитектура основана на модели в формате NoSQL, алгоритмах, основанных на экспертных правилах и методах машинного обучения, предложенных в диссертации.

Практическая значимость диссертационной работы заключается в следующем:

- модель представления больших данных об инсайдерских атаках является основой для формализации данных и знаний о пользователях, устройствах, приложениях и сервисах в КС;

- модель и алгоритмы комбинированного применения экспертных правил и методов машинного обучения позволяют оперировать большими объемами данных и выявлять инсайдеров для достижения наилучших показателей эффективности; произведена настройка алгоритмов на основе методов машинного обучения по типовым сценариям инсайдеров в КС; обосновано комбинированное применение алгоритмов обнаружения инсайдеров;

- методика обнаружения инсайдеров повышает эффективность обнаружения внутренних нарушителей в КС (оперативность повышается за счет использования методов обработки больших данных; результативность – за счет совместного использования алгоритмов на основе экспертных правил и методах машинного обучения, ресурс-экономность – за счет новых высокотехнологичных программно-аппаратных решений);

- архитектура и программная реализация системы способствует эффективному обнаружению инсайдеров в КС с использованием предложенной методики обнаружения инсайдеров, обеспечивающей комбинированное применение технологий обработки больших данных, экспертных правил и методов машинного обучения.

Методология и методы исследований. Для решения поставленных задач использовались как классические, так и современные методы исследования, а именно: системный, причинно-следственный и сравнительный анализ был применен в равной степени для получения практически всех основных научных результатов; аппарат теории вероятностей и теория множеств применялись в интересах формирования математической модели представления больших данных для обнаружения инсайдеров; сбор, систематизация и анализ научно-технической информации предметной области, а также функциональный и структурный синтез, позволили создать модель и комплекс алгоритмов обнаружения инсайдеров; методы машинного обучения явились центральным звеном одного из алгоритмов

комплекса обнаружения инсайдеров; методы обработки больших данных [147, 153, 157, 160, 165, 174, 175, 180, 181, 189] легли в основу методики обнаружения инсайдеров, затрагивая, тем самым, все остальные полученные результаты; для практической оценки методики и программной реализации системы обнаружения инсайдеров был проведен компьютерный эксперимент на базе имитационного моделирования [61, 170]; основой программной реализации системы обнаружения инсайдеров послужила общая методология программирования.

Положения, выносимые на защиту. Основными положениями, выносимыми на защиту, являются:

1. Модель представления больших данных об инсайдерских атаках в формате NoSQL, обеспечивающая хранение и анализ признаков пользователей в компьютерных сетях в различные моменты времени.

2. Модель и алгоритмы комбинированного применения экспертных правил и методов машинного обучения в интересах обнаружения инсайдерских атак.

3. Методика обнаружения инсайдеров в компьютерных сетях с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных.

4. Архитектура и программная реализация системы обнаружения инсайдеров в компьютерных сетях с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных.

Научная новизна результатов диссертационной работы состоит в следующем:

1. Модель представления больших данных об инсайдерских атаках в формате NoSQL отличается от существующих возможностью обеспечения хранения и анализа признаков пользователей, полученных на базе UBA/UEBA-аналитики и характеризующих потенциальную инсайдерскую деятельность в компьютерных сетях, а также возможностью учета динамики изменения этих признаков.

2. Модель и алгоритмы комбинированного применения экспертных правил и методов машинного обучения в интересах обнаружения инсайдерских атак

отличаются от существующих применением комплексного подхода к решению задачи обнаружения инсайдеров с учетом признаков и свойств пользователей, устройств, приложений, сервисов, включая параметр времени.

3. Методика обнаружения инсайдеров отличается от существующих использованием предложенной модели представления больших данных об инсайдерских атаках, а также предложенных модели и алгоритмов комбинированного применения экспертных правил, методов машинного обучения и обработки больших данных.

4. Архитектура и программная реализация системы обнаружения инсайдеров в компьютерных сетях отличается от известных архитектур и программных средств использованием предложенной методики обнаружения инсайдеров, обеспечивающей комбинированное применение технологий обработки больших данных, экспертных правил и методов машинного обучения.

Реализация результатов работы. Отраженные в диссертационной работе исследования проведены в рамках федеральной целевой программы 2019-2020 гг. «Разработка методов, моделей, алгоритмов и программных средств, основанных на выявлении отклонений в эвристиках трафика сверхвысоких объемов, для обнаружения сетевых атак и защиты от них». Данное исследование проводится при поддержке Минобрнауки России в рамках Соглашения № 05.607.21.0322 (идентификатор RFMEFI60719X0322). Полученные результаты внедрены в учебный процесс СПбГУТ (учебные курсы: «Безопасность компьютерных сетей», «Безопасность беспроводных локальных сетей») и СПбГУТПД (учебные курсы: «Комплексная защита информации на предприятии», «Технологии и методы программирования»), применяются в рабочем процессе Роскомнадзора по Северо-Западному федеральному округу, компании ООО «Фаст Лейн». Результаты диссертационного исследования представлены в заявке, победившей на конкурсе субсидий молодым ученым, молодым кандидатам наук вузов, отраслевых и академических институтов, расположенных на территории Санкт-Петербурга, в 2019 г.

Обоснованность и достоверность полученных результатов обеспечивается за счет тщательного анализа состояния исследований предметной области, подтверждается согласованностью результатов с экспериментальными оценками, успешной апробацией основных теоретических положений диссертации на ряде научных конференций всероссийского и международного уровня, а также публикацией основных научных результатов в ведущих рецензируемых научных изданиях.

Апробация результатов работы. Основные положения и результаты работы докладывались на научных конференциях: международной конференции по интеллектуальным распределенным вычислениям IDC-2019 (Санкт-Петербург, 2019), международной конференции IEEE SMARTWORLD ATC-2017 (Сан-Франциско, 2017); Санкт-Петербургской межрегиональной конференции «Информационная безопасность регионов России» (Санкт-Петербург, 2015, 2017, 2019), Международной научно-технической и научно-методической конференции «Актуальные проблемы инфотелекоммуникаций в науке и образовании» в СПбГУТ (Санкт-Петербург, 2015–2019); XV-й Санкт-Петербургской международной конференции «Региональная информатика» (Санкт-Петербург, 2019); Российской мультиконференции по проблемам управления «Информационные технологии в управлении» (Санкт-Петербург, 2016).

Личный вклад. Все результаты, представленные в диссертационной работе, получены лично автором в процессе выполнения научно-исследовательской деятельности.

Публикации. По материалам диссертационной работы опубликовано 40 работ, в том числе 9 – в рецензируемых изданиях из перечня ВАК («Вопросы кибербезопасности», «Защита информации. Инсайд», «Труды СПИИРАН», «Труды учебных заведений связи»), 2 – в изданиях, индексируемых в международных базах Scopus и Web of Science, получено 3 свидетельства о государственной регистрации программ для ЭВМ.

Структура и объем диссертационной работы. Диссертационная работа включает введение, три главы, заключение, список литературы (190 наименований)

и 2 приложения. Объем работы – 206 страниц машинописного текста; включает 35 рисунков и 13 таблиц.

Краткое содержание работы. В первой главе проведен анализ проблемы обнаружения инсайдеров в КС. Установлены место и роль задачи обнаружения инсайдеров в КС в общем цикле обработки информации в SIEM-системе. Выполнена постановка задачи исследования и сформулирована цель исследования.

Во второй главе представлены разработанные модель представления больших данных об инсайдерских атаках в формате NoSQL, модель и комплекс алгоритмов обнаружения инсайдеров в компьютерной сети на основе экспертных правил и методов машинного обучения. Описаны типовые сценарии инсайдерских атак.

В третьей главе описываются разработанные методика, архитектура и программная реализация системы обнаружения инсайдеров в КС. Представлены результаты экспериментов и сравнение предложенной методики с существующими аналогами, а также предложения по применению разработанного модельно-методического аппарата для обнаружения инсайдеров в КС.

Глава 1. Системный анализ задачи обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных

1.1. Место и роль задачи обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных

В настоящее время решению проблемы обнаружения инсайдерских атак уделяется много внимания. Доказательством этого могут служить инсайдерские атаки, совершенные на протяжении 2019 года.

Компания DeviceLock, являющаяся российским производителем DLP-систем, провела исследование каналов инсайдерских утечек информации в российских компаниях (результаты исследования отображены на сайте http://it-text.ru/Articles/2019_06/2019_06_21_2.htm). В рамках исследования, охватившего период с января по май 2019 года, были проанализированы более 800 документов, выложенных на различные ресурсы проекта DarkNet, а также предоставленные продавцами услуги в качестве образцов предлагаемых ими данных.

Другим подтверждением роста инсайдерских атак может служить ежегодный отчет Data Breach Investigations Report (DBIR) от компании Verizon (отчет доступен на сайте https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report.pdf), предоставляющий глубокий анализ последних тенденций и изменений, связанных с инцидентами кибербезопасности. Проведя глубокий анализ отчета, можно сделать вывод о том, что последние 4 года количество инцидентов, связанных с инсайдерскими угрозами, растет на 5%, начиная с низкого показателя в 2015. Отчет за 2019 год показал, что 34% всех нарушений случилось вследствие инсайдерских атак.

Компания Cisco Systems Inc. недавно изучила тенденции утечки данных [78], применив алгоритм на базе методов машинного обучения, чтобы составить профили 150 000 пользователей в 34 странах, пользующихся услугами поставщиков облачных сервисов, с января по июнь 2019 года. Этот алгоритм учитывал не только объем загружаемых документов, но и разные переменные данные, например, время загрузки в течение дня, IP-адреса и местоположение. Профили пользователей составлялись в течение полугода, затем исследователи полтора месяца изучали аномалии, – 0,5% пользователей было отмечено как совершающие подозрительные действия по скачиванию программ. Из числа этих подозрительных скачиваний 62% приходилось на стандартные рабочие часы, 40% происходило по выходным [78].

Проанализировав отчет аналитического центра компании InfoWatch, можно отметить, что за 2019 год было зарегистрировано 1039 случаев утечки конфиденциальной информации, что на 12% больше, чем годом ранее (https://www.infowatch.ru/sites/default/files/report/analytics/russ/Global_Data_Leaks_Report_2019_half_year.pdf?rel=1). Следует заметить, что объем информации, скомпрометированной по вине хакерских и иных атак под воздействием внешнего нарушителя, уменьшился в десять раз, составив только около 0,5 млрд. записей. При этом в результате нарушений внутри организаций пострадали более 1,5 млрд. записей данных, включая персональные и платежные.

Вывод о том, что общее количество инсайдерских атак растет, подтверждается в том числе и банком данных угроз безопасности информации ФСТЭК России [1]. Проведя анализ банка данных угроз ФСТЭК России, было выявлено, что доля угроз от внутренних нарушителей с низким, средним и высоким потенциалом составляет 68% при общем количестве угроз, зарегистрированных в банке данных угроз ФСТЭК России – 216. Результаты анализа представлены в таблице 1.1.

Таблица 1.1 Общая таблица нарушителей банка данных угроз безопасности информации ФСТЭК России

Потенциал нарушителя		**Количество угроз**	**Доля угроз**
Внутренний нарушитель	с низким потенциалом	91	42%
	со средним потенциалом	52	24%
	с высоким потенциалом	3	1%
	Всего	**146**	**68%**
Внешний нарушитель		216	32%

Таким образом, на основании приведенных примеров, а также на основании анализа банка данных угроз безопасности информации ФСТЭК России [1], можно сделать вывод о неуклонном росте инсайдерских атак (рисунок 1.1), а это значит, что тема обнаружения инсайдеров в компьютерных сетях является актуальной и обоснованной.

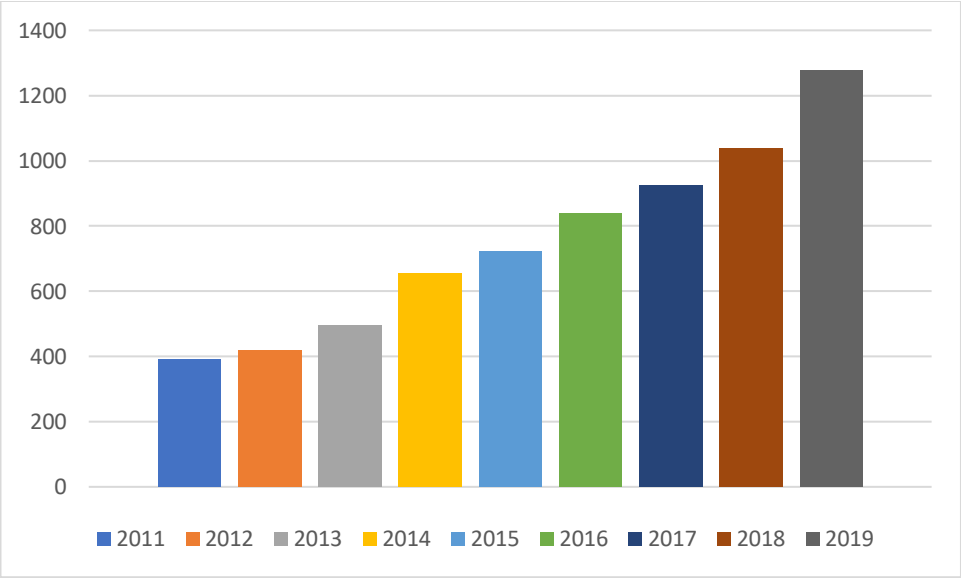

Рисунок 1.1 – Общее количество инсайдерских атак

Таксономии инсайдеров и инсайдерских угроз. В настоящее время одним из наиболее широко используемых подходов для составления таксономии инсайдеров является опросный метод. В основе этого метода лежит анализ материалов расследований инцидентов компьютерной безопасности, проводившихся специалистами по компьютерной безопасности. На основе расследованных инцидентов возможно осуществить категоризацию исследований и использовать как технические, так и психосоциальные данные.

Рассмотрим список опросов, связанных с внутренними угрозами. Salem [158, 159] ввел таксономию злоумышленников, разделив их на две категории в соответствии с имеющимися у них знаниями о целевой системе: предатели и маскарадчики. Проведя анализ литературы по обнаружению инсайдеров, можно разделить работы на три типа по виду подходов: «подходы профилирования пользователей на основе хоста», «подходы сетевого уровня» и «интегрированные подходы». Сетевой уровень и хост на основе профилирования может иметь высокую вероятность обнаружения предателей, в то время как профилирование пользователей на основе хоста может быть успешным при выявлении скрытых угроз. Авторы утверждают, что злонамеренные действия инсайдеров происходят на уровне приложений и бизнес-процессов. Hunker и Probst [104] предложили категоризацию исследования, основанную на комбинации психосоциальных исходных данных с техническими данными. Результирующие категории состоят из трех типов подходов к обнаружению угроз со стороны инсайдеров: 1) «социологический, психологический, организационный», 2) «социально-технический» и 3) «технический». Авторы подчеркнули, что для успешных методов обнаружения внутренних угроз требуется сочетание различных подходов.

Использование современной концепции BYOD в работе организаций провоцирует возникновение инсайдерских инцидентов, при которых атака осуществляется извне, но посредством устройств инсайдеров. Технические возможности атакующего значительно возрастают в том случае, если пользователь находящегося внутри периметра устройства осознанно содействует аутсайдеру,

действуя согласованно с ним. Однако, возможна ситуация, когда на устройстве сотрудника без его ведома и согласия установлены программные средства удаленного управления и специализированное ПО для сбора информации и выполнения атаки на КС. Сотрудник, находясь внутри периметра, фактически предоставляет техническую платформу для атаки, при этом зачастую неосознанно. Соответственно, технические возможности аутсайдера в этом случае будут определяться возможностями скрытого ПО. При этом атакующий также ограничен в знаниях и в возможностях подключения к КС, а значит, накопление нужной информации будет происходить менее эффективно. Использование BYOD может вызывать и непреднамеренное распространение внутри периметра вредоносного ПО с принесенного устройства вследствие его недостаточной защищенности и неопытности пользователя.

По мнению Неймана [137], аутсайдерами считаются злоумышленники, которые успешно проникли в ИС или сеть и не получили достаточно знаний, чтобы сделать их неотличимыми от инсайдеров, которыми они маскируются. Для проведения дифференциации аутсайдеров и инсайдеров автор рассматривает тест Тьюринга. Более того, инсайдерская угроза также была определена как часть более широких тем, которые возникали на протяжении всей истории компьютерных систем, таких как попытки вторжения, угрозы безопасности и контрпродуктивное поведение на рабочем месте. В настоящее время наиболее актуальные масштабы внутренней угрозы поддерживаются подразделением CERT Института программной инженерии Университета Карнеги-Меллона и основаны на базе данных более чем 1000 реальных примеров из практики.

Pfleeger [148] определяет инсайдера как "лицо, имеющее законный доступ к компьютерам и сетям организации". Отчет RAND Corp. [67] определяет инсайдера как "заведомо доверенное лицо, имеющее доступ к конфиденциальной информации и информационным системам" (ИС).

Термин "непреднамеренная внутренняя угроза" согласно [82], определяется как "текущий или бывший сотрудник, подрядчик или другой деловой партнер», который: 1) "имеет или имел санкционированный доступ к сети, системе или

данным организации" и 2) "не имел злого умысла, связанного с его действиями (или бездействием), причинившими вред или существенно увеличившими вероятность серьезного ущерба конфиденциальности, целостности или доступности информации или ИБ организации в будущем". В [125] считают, что непреднамеренная внутренняя угроза определяется как "невнимательные, самодовольные или необученные люди, которым для выполнения своей работы требуется разрешение и доступ к ИС".

Также таксономия инсайдеров может быть построена на основе анализа мотивации. Рассматривая мотивацию, Wall в [179] утверждает, что непреднамеренные внутренние угрозы делятся согласно предыдущим исследованиям на благонамеренные и небрежные. Далее автор предлагает в качестве основы для структурной оценки типологию четырех типов сотрудников, которые могут привести к утечке данных: нарушители (те, которые делают свою жизнь проще, не соблюдая политики безопасности), чрезмерно амбициозные (те, которые специально обходят меры безопасности, чтобы быть более эффективными), социальные инженеры и лица, занимающиеся кражей данных. Автор описывает различные способы утечки данных, такие как случайная утечка (потеря USB-накопителей); для удобства пользователей (копирование данных на домашние компьютеры); прочие (наличие данных на жестких дисках, поврежденных ПК, обмен данными с третьими лицами, утечка данных через открытую почту и т.д.).

Таксономия человеческих ошибок. Как уже отмечалось ранее, проблема инсайдерских инцидентов не является чисто технической, а исходит из взаимодействия человека с технической системой. При этом, помимо явных намерений осуществить определенные действия, пользователи внутри периметра могут совершать и ошибки. Reason [155] определяет человеческую ошибку как "недостижение желаемого результата в запланированной последовательности психических или физических действий, когда неудача не вызвана случайностью". Автор предлагает общую систему моделирования ошибок (GEMS). GEMS делит ошибки на следующие группы: "проскальзывания", которые представляют собой

неправильное выполнение правильной последовательности действий (сбой выполнения), и "ошибки", которые представляют собой правильное выполнение неправильной последовательности действий (сбой планирования).

Типы внутренних атак. Bellovin [63] выделяет три вида внутренних атак:

1. Злоумышленное использование доступа, которое, по мнению автора, наиболее трудно обнаружить, поскольку инсайдеры используют законный доступ, но не по назначению.

2. Обход защиты, когда инсайдеры уже прошли некоторые линии защиты (например, брандмауэры), и могут также обойти другие.

3. Создание технических проблем, когда либо элемент контроля доступа имеет неправильные настройки, либо такой элемент присутствует в доступе. Предпосылки для такого рода атак могут создавать администраторы КС, небрежно относящиеся к вопросам назначения доступа и хранения учетной информации.

Уровни внутренних угроз. Основываясь на различных потенциальных последствиях и вреде для организации, Cole и Ring [81] различают три уровня внутренних угроз:

Уровень 1: Инсайдеры с собственной мотивацией, не мотивированы никакими третьими лицами, но действуют самостоятельно по личным причинам.

Уровень 2: Завербованные инсайдеры – это лица, действующие во вред и мотивированные третьей стороной. Поскольку такой тип завербованного инсайдера представляет риск как для инсайдера, так и для вербовщика, очевидно, что предпочтительным способом для злоумышленников является использование собственного подготовленного инсайдера.

Уровень 3: Вредоносные организации, которые целенаправленно размещают инсайдеров в компании-жертве. Находя подходящего на роль инсайдера сотрудника, организации обучают его, устраивают на работу в компанию, а затем начинают использовать инсайдеров для совершения злонамеренных действий.

Типы мотивов инсайдеров. По мнению Cole и Ring [81], существует множество факторов, которые могут способствовать мотивации инсайдеров.

Наиболее популярны три основных мотивации: 1) финансовые трудности или желание дополнительного заработка; 2) политические – некоторые сотрудники с политическими убеждениями могут быть настроены весьма радикально и мотивированы причинить вред при появлении любой возможности; 3) личные.

Профилирование инсайдеров CERT. Cappelli [69] предлагают разделить вредоносные внутренние угрозы на три типа:

1) Саботаж в сфере ИТ, при котором "инсайдер использует ИТ для нанесения конкретного вреда организации или отдельному лицу". Такими инсайдерами обычно являются недовольные технически подготовленные сотрудники, обладающие административными привилегиями.

2) Кража интеллектуальной собственности. Обычно речь идет о шпионаже, который осуществляется как техническим персоналом, а также не техническим персоналом. Виновные могут красть информацию, к которой они ежедневно имеют доступ, и забирать ее с собой, когда покидают организацию.

3) Мошенничество, при котором "инсайдер использует ИТ для несанкционированной модификации данных организации в целях личной выгоды или хищения". Обычно совершается сотрудниками низшего звена с не техническим образованием, мотивом которых являются алчность или финансовые трудности, мошенничество совершается на протяжении длительного периода.

Таксономия инсайдерской угрозы на основе политик. Таксономия внутренней угрозы на основе политик может быть выведена из определений и соответствующей модели внутренней угрозы, предложенных Bishop at al. [65]. Авторы утверждают, что "политика безопасности представляет собой правила контроля доступа, применяемые в организации". Исходя из этого определения, инсайдерскую угрозу можно классифицировать с учетом двух примитивных действий: 1) "нарушение политики безопасности с использованием законного доступа", включающее действия, которые не соответствуют действующей политике безопасности; и 2) "нарушение политики контроля доступа путем получения несанкционированного доступа" – инсайдеры злоупотребляют своим

законным доступом с целью расширения своих фактических привилегий, нарушая действующую политику безопасности.

Категоризация внутренних злоумышленников по физическому присутствию. Рассматривая физическое присутствие, Neumann [136] подразделяет инсайдеров на две категории: логическую и физическую. Логический инсайдер осуществляет свои действия физически за пределами рабочего пространства организации, а физический инсайдер действует в пределах физических границ рабочего пространства организации, в том числе внешних доверенных сетей [34]. Данная классификация не определяет намерения инсайдеров, и может применяться как к непреднамеренным, так и к злонамеренным внутренним угрозам.

Человеко-ориентированная таксономия внутренних злоумышленников. Magklaras и Furnell [129, 151] предлагают ориентированную на человека таксономию внутренних злоумышленников, учитывающую три измерения: роль инсайдеров в системе, причины неправильного использования и последствия для системы.

1. Системная роль – первое измерение классифицирует людей по "типу и уровню системных знаний, которыми они обладают". Этот аспект состоит из: системных администраторов, которые полностью контролируют большую часть ресурсов ИС; продвинутых пользователей, которые не имеют привилегированного доступа, но обладают большим объемом знаний о системах и сетях организации, и способны выявить уязвимости системы [8, 149, 163]; и пользователей приложений, которые используют определенные стандартные приложения, но имеют больше прав доступа, чем обычно.

2. Причина неправомерного использования – данный аспект описывает признаки инцидентов, связанных с инсайдерской угрозой. Учитывая этот аспект, авторы делят инсайдеров на две группы: умышленные неправомерные действия по различным причинам и случайные злоупотребления, которые также могут быть классифицированы по фактической причине, оказавшей негативное влияние на поведение законного пользователя.

3. Системные последствия – этот аспект различает различные способы совершения неправомерных действий, что проявляется в наличии определенных следов в ИТ-инфраструктуре на системном уровне.

Авторы описывают три уровня, которые приписываются этим последствиям: 1) ОС – изменения в структуре файловой системы, установка несанкционированного программного обеспечения и т.д.; 2) сетевые – сетевые пакеты могут содержать несанкционированное содержимое, эксфильтрация конфиденциальных данных через службы обмена электронной почтой или файлами и т.д.; 3) аппаратное – вандализм или удаление аппаратных компонентов, установка регистраторов, изменение конфигурации критических аппаратных компонентов (например, с целью саботажа или кражи IP).

Обнаружение внутренних атак. Phyo and Furnell [151] предлагают классификацию внутренних атак, которая различает четыре уровня мониторинга системы-мишени, на которых можно обнаружить атаку:

1) сеть;
2) операционную систему;
3) приложение;
4) данные.

Эта таксономия основана на предположении, что одна внутренняя атака может проявляться на определенных уровнях системы, а следы другой внутренней атаки могут присутствовать на разных уровнях (например, нарушение целостности данных может быть очевидным на уровне приложений и данных, а утечка данных может проявляться на уровне сети и ОС).

Структурная таксономия инсайдерских инцидентов по 5W1H. Принимая во внимание приведенные выше определения и таксономию, можно составить структурную таксономию инсайдерской угрозы. Для обеспечения единого взгляда на существующие таксономии была использована методология "кто, что, где, когда, почему и как" (5W1H). 5W1H – это элементарные вопросы, касающиеся проблемы сбора информации, которые первоначально использовались для репортажей о новостях, но также имели и другие области применения. Поскольку

проблему расследования инсайдерских инцидентов можно также рассматривать как пример сбора информации, был выбран именно этот подход для структурной таксономии подобного рода инцидентов на основе злонамеренных действий (рисунок 1.2) и действий, совершенных неумышленно (рисунок 1.3).

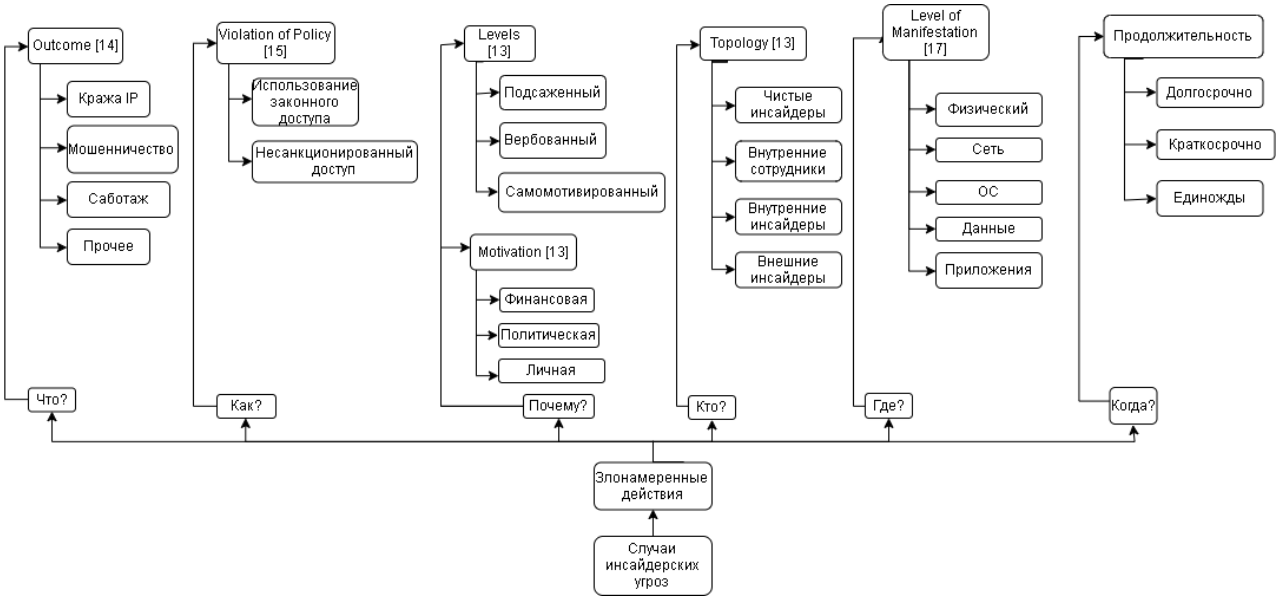

Рисунок 1.2 – Структурная таксономия инсайдерской угрозы (на основе предыдущих исследований и вопросов 5W1H)

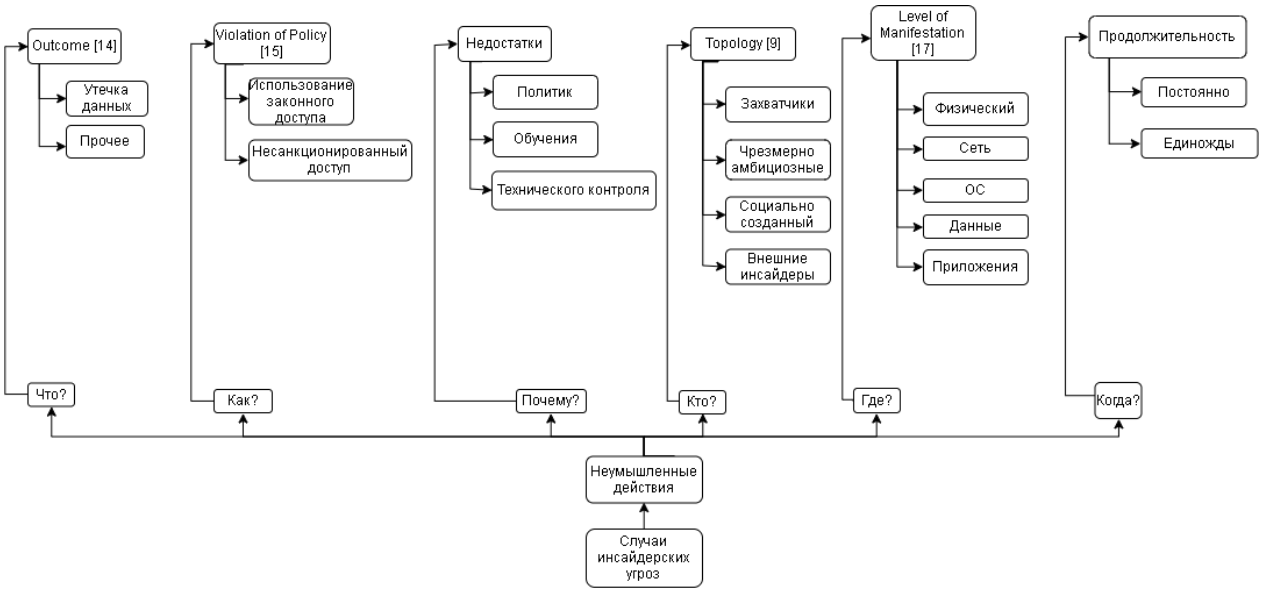

Рисунок 1.3 – Структурная таксономия инсайдерской угрозы (на основе предыдущих исследований и вопросов 5W1H)

Изучив таксономии, связанные с инсайдерской угрозой, можно перейти к новой классификации существующих инсайдерских угроз, представив их в виде четырех основных классов.

1. Угроза с использованием инцидентов и баз данных. Класс содержит справочные наборы данных для оценки подходов к обнаружению инсайдерских угроз, а также сборники материалов, описывающих реальные ситуации инсайдерских инцидентов, которые могут быть использованы для анализа и моделирования внутренних угроз или для разработки решений защиты, направленных на конкретные типы инцидентов.

2. Угроза, основанная на анализе инцидентов. Класс направлен на обобщение и моделирование всех связанных аспектов и поведение инцидентов, связанных с внутренней угрозой, включая определения и таксономию, поведенческие модели жизненного цикла внутренней угрозы, наблюдаемые показатели и критические пути с точки зрения безопасности.

3. Угроза на основе моделирования. Класс включает в себя исследования, в ходе которых проводятся эксперименты с программируемыми моделями имитируемых инсайдерских сред, либо с целью изучения влияния различных параметров моделирования на выполнение моделирования, либо с целью создания синтетических наборов данных, которые могут быть использованы для тестирования оборонных решений.

4. Угроза на основе решения защиты. Эта категория является самой большой и включает в себя статьи, в которых предлагаются решения для оценки, предотвращения или устранения внутренних угроз. Кроме того, в целях полноты охвата данная категория также включает в себя различные практические решения в области защиты. Таким образом, данная категория дает представление о спектре вариантов защиты и выявляет тенденции и идеи в развитии оборонных решений.

Рассмотрим место и роль задачи обнаружения инсайдеров в КС в общем цикле обработки информации в SIEM-системе – представлены на рисунке 1.4 (с помощью красных пунктиров).

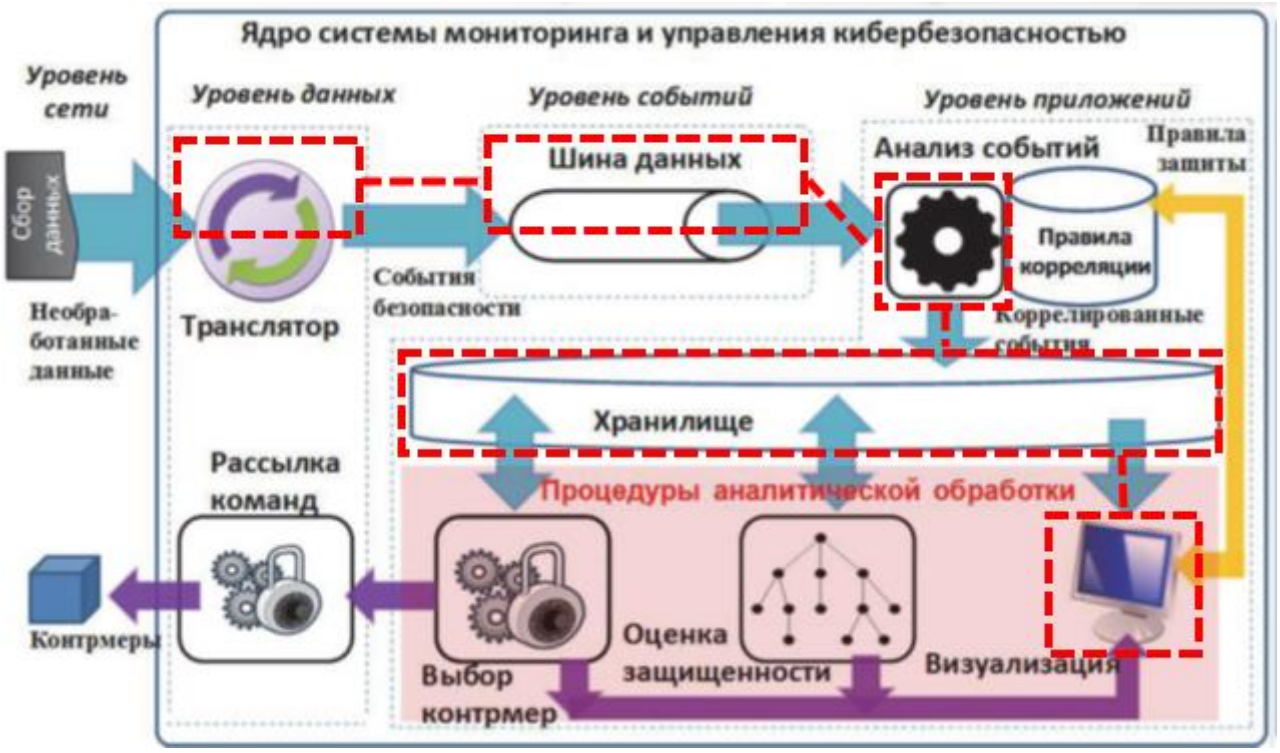

Рисунок 1.4 – Общий цикл обработки информации в SIEM-системе

Согласно рисунку 1.4, задача обнаружения инсайдеров частично относится к уровню данных и событий (за счет использования агентов сбора данных и сообщений, необходимых для анализа), уровню приложений (с применением комплекса алгоритмов обнаружения инсайдеров по собранным данным), к хранилищу (для хранения исходных, промежуточных и конечных данных) и визуализации (поскольку предоставляет результаты своей работы в виде, адаптированном для анализа экспертом ИБ, например, с помощью [105]). Таким образом, области цикла обработки, связанные с работой на уровне сети, оценкой защищенности, использованием правил защиты, выбором контрмер и рассылкой соответствующих команд, задачей обнаружения инсайдеров не затрагиваются [182].

Так, к входным данным для задач, которые решаются в настоящем исследовании, могут быть отнесены данные о сети и регистрируемые события, а к выходным данным – визуализация результатов работы в виде отчета об обнаруженных инсайдерах.

1.2. Современное состояние проблемы обнаружения инсайдеров в КС с применением методов обработки больших данных для мониторинга компьютерной безопасности

В настоящее время наблюдается значительный рост объема и многообразия обрабатываемых данных, а также скорости их обработки, связанного, прежде всего, с развитием информационных технологий и вычислительных способностей дата-центров [47]. Организации ежегодно генерируют огромные массивы разнородных данных, которые требуют эффективного управления [47, 160].

Для описания технологий оперирования с огромными массивами информации возникло понятие "большие данные" (big data) – термин, обозначающий технологии обработки и анализа структурированных и неструктурированных данных больших объемов. Большие данные описываются моделью 3V, изначально разработанной аналитиками компании Gartner в 2001 году: большой объем (volume), скорость (velocity) и неструктурированность данных (variety) [47], позже дополненной еще двумя V-параметрами: достоверностью (veracity) и значимостью (valuability) данных. Таким образом, появилась модель 5V. В последующем появились модели 9V и даже 11V.

Основной целью применения больших данных в контексте мониторинга компьютерной безопасности [50, 118,] является возможность консолидировать и анализировать данные об инцидентах компьютерной безопасности из огромного множества источников с высокой производительностью [118].

Основными факторами развития обработки больших данных явились: значительное уменьшение стоимости систем хранения данных; разработка эффективных технологий построения дата-центров и облачных вычислений; разработка инструментариев для обработки больших данных, в том числе таких, как экосистема Hadoop, Spark, базы данных NoSQL, которые позволили значительно повысить скорость обработки комплексных запросов [31].

Обработку больших данных можно разделить на две группы: (1) технологии пакетной (batch) обработки, когда данные анализируются независимо от потока данных, и (2) технологии потоковой (stream) обработки, когда данные обрабатываются в потоке.

Типичным примером системы batch-обработки является платформа Hadoop. Опишем базовые элементы данной платформы как основы для построения ставшей уже традиционной и широко используемой обработки больших данных.

Hadoop представляет разработчикам файловую систему HDFS (Hadoop Distributed File System) для обработки файлов больших размеров, программную модель MapReduce, систему YARN для управления ресурсами в кластере Hadoop, а также Hadoop Common – связующее программное обеспечение, состоящее из набора различных утилит, программных библиотек, компонентов и интерфейсов для распределенных файловых систем, используемых для поддержки различных родственных проектов Hadoop и других модулей [53, 56, 58, 59].

MapReduce – это платформа для вычисления наборов распределенных задач с использованием большого количества компьютеров (узлов, nodes), которые организуются в кластер. Принцип действия MapReduce заключается в выполнении трех стадий: классификации (Map), обработки (Shuffle) и распределения (Reduce). Саму парадигму MapReduce предложила компания Google в 2004 году. На рисунке 1.5 представлена схема функционирования MapReduce.

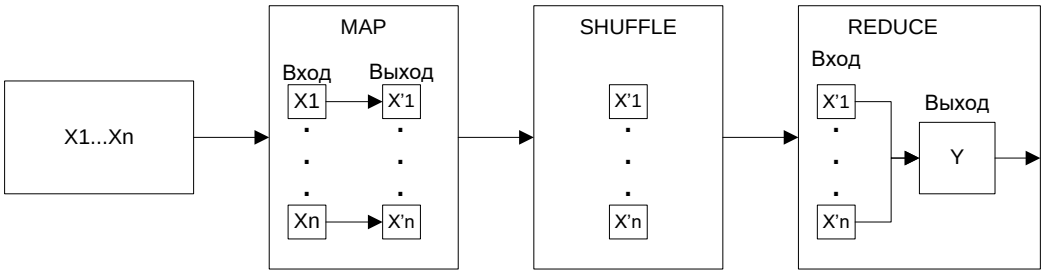

Рисунок 1.5 – Схема функционирования MapReduce

На первой стадии – планирования (Map) – происходит предварительная обработка входных данных. Для этого один из компьютеров (главный узел или

мастер) получает входные данные задачи, разделяет их на части и передает другим компьютерам (рабочим узлам) для предварительной обработки. Map – это функция, принимающая на вход одну запись вида (KEY, VALUE) и возвращающая по ней любое количество новых записей (KEY_1, $VALUE_1$), ..., (KEYn, VALUEn). В данном контексте KEY – ключ, VALUE – значение [31].

Стадия смешивания (Shuffle) проходит незаметно для пользователя. На этой стадии результаты выполнения функции планирования «разбираются по категориям». Каждая категория соответствует одному ключу ввода функции Map. В дальнейшем эти категории служат входом для стадии вывода (Reduce).

Reduce – функция, принимающая на вход все записи с данным ключом. Итерируясь по ним, она также может возвращать любое количество новых записей, которые и являются финальным результатом MapReduce-задачи.

На стадии Reduce происходит свертка предварительно обработанных данных. Главный узел получает ответы от рабочих узлов и на их основе формирует результат – решение задачи, которая изначально формулировалась. Преимущество MapReduce заключается в распределении операций предварительной обработки и свертки, что позволяет обрабатывать огромные массивы данных за приемлемое время (например, сортировка петабайта данных (10^{15} байт) может достигать нескольких часов) [31].

YARN (Yet Another Resource Negotiator) – это система управления ресурсами в кластере Hadoop. Эта система служит для управления выполнением MapReduce задач, но также может работать и с другими парадигмами распределенных вычислений.

Данный сервис состоит из множества компонентов и берет на себя задачу по контейнеризации запускаемых задач, предоставлению необходимых ресурсов и их распределению в условии конкуренции и доставки контейнера с кодом непосредственно к данным, следуя принципу локальности данных, за счет чего экономятся ресурсы сети.

Hadoop Common содержит библиотеки для управления распределенными файловыми системами, а также различные сценарии создания инфраструктуры,

необходимой для управления распределенной обработкой данных. Для удобства выполнения сценариев предусмотрен специальный интерпретатор командной строки, который называется «FS Shell» (File System Shell).

Также используются инструменты, которые позволяют запускать поверх Hadoop комплексные запросы и методы машинного обучения. Среди них можно выделить следующие инструменты: Pig – платформа и скриптовой язык для написания комплексных запросов [57], Hive – инструмент для SQL-запросов над большими данными (превращает SQL-запросы в серию MapReduce задач), RHadoop – механизмы анализа данных и методы машинного обучения для Hadoop.

Для улучшения производительности анализа данных и алгоритмов машинного обучения с использованием потоковой обработки запросов были разработаны также новые платформы, такие как Spark.

Для обработки запросов были спроектированы несколько систем управления базами данных, такие, как, например, Cassandra, CouchDB, GreenplumDatabase, HBase [54], MongoDB, Vertica и др.

С развитием технологий, требования традиционных моделей данных организации/предприятия к приложениям, базам данных и ресурсам хранения выросли, а стоимость и сложность этих моделей увеличились совместно с потребностью в обработке больших данных. Традиционная модель была расширена, чтобы охватить новые структурные элементы и решить проблемы, связанные с новыми специализированными платформами обработки информации [169, 120, 133], которые были ориентированы на удовлетворение требований, обуславливаемых большими данными.

Как правило, в существующих решениях в структуру организации/предприятия предлагается добавить три основных элемента для размещения больших данных: 1) например, платформу Hadoop, которая обеспечивает возможность хранения больших данных в распределенной и совместно используемой файловой системе; 2) возможность анализа больших объемов данных с помощью технологии MapReduce; 3) а также базы данных

NoSQL, обеспечивающие возможность получать и обновлять большой поток неструктурированных данных в режиме реального времени.

В [38] представлен пример модели больших данных центра обработки данных (ЦОД). К традиционным элементам центра обработки данных, включающим в себя базы данных, основанные на SQL, и системы хранения данных (Storage Area Network, SAN), добавляются неструктурированные данные (базы данных NoSQL), использующие в качестве платформы такую систему как Hadoop вместе с MapReduce для выполнения хранения и анализа данных.

Компоненты больших данных могут успешно интегрироваться в текущие бизнес-модели организации/предприятия. Например, компания Cisco Systems Inc. обеспечивает интеграцию в ЦОД с помощью сетевых коммутаторов Cisco Nexus, оптимизированных для больших данных [41]. Особенностью данных коммутаторов является возможность построения единой инфраструктуры ЦОД [9, 60] организации при эффективном использовании всей линейки коммутаторов. Эта линейка включает: коммутаторы уровня ядра (Nexus 7000, 9000), которые реализуют быструю коммутацию данных и являются основным элементом управления ЦОД; коммутаторы уровня агрегации, включающие в себя модели Nexus 5600 и обеспечивающие управление трафиком данных ЦОД, получаемых от уровня доступа, представленного выносными линейными платами (Fabric Extender) серии Nexus 2300.

Мониторинг компьютерной безопасности на основе обработки больших данных. Важнейшими приложениями в области компьютерной безопасности, где механизмы анализа больших данных получают в настоящее время значительное распространение, являются системы мониторинга компьютерной безопасности, в том числе, системы обнаружения и предотвращения вторжений (IPS/IDS) и системы управления информацией и событиями безопасности (SIEM-системы) [10].

Как правило, SIEM-система имеет архитектуру «агенты» – «хранилище данных» – «сервер приложений», которая разворачивается поверх защищаемой

информационной инфраструктуры. Агенты выполняют сбор событий безопасности и их первоначальную обработку.

Собранная и обработанная информация поступает в хранилище данных, где она хранится во внутреннем формате представления.

Сервер приложений реализует основные функции по обработке информации, анализируя информацию, хранимую в хранилище, и преобразует ее для выработки предупреждений или решений по защите информации [31]. На рисунке 1.4 представлен полный цикл обработки данных в перспективной SIEM-системе.

В данной архитектуре можно выделить четыре уровня: (1) уровень сети, к которому относятся компонент сбора данных с устройств в сети и компонент реализации контрмер; (2) уровень данных, к которому относятся транслятор, необходимый для надежной передачи сообщений, и компонент рассылки команд; (3) уровень событий, предоставляющий шину данных и служащий для оперативного хранения и передачи поступающей информации; (4) уровень приложений, состоящий из компонента корреляции событий, их анализа, долговременного хранилища и специальных процедур аналитической обработки. Указанная архитектура является абстрактной и не раскрывает особенностей обработки большого потока входных данных.

Можно условно выделить три поколения систем IPS/IDS и SIEM [31]:

1. Первое поколение – системы обнаружения вторжений (IDS), которые преимущественно выполняют детектирование злоупотреблений и аномалий трафика в соответствии с сигнатурами и генерируют инциденты безопасности.

2. Второе поколение – системы управления информацией и событиями безопасности (SIEM), особенностью которых является повышение уровня защищенности компьютерных сетей за счет обеспечения возможности в режиме, близком к реальному времени, манипулировать информацией и событиями безопасности и осуществлять проактивное управление инцидентами.

3. Третье поколение – системы, реализующие обработку больших данных в области компьютерной безопасности (SIEM-системы второго поколения); использование обработки больших данных позволяет обрабатывать требуемые

массивы разнородной информации за приемлемое время и снизить общее время на выполнение операций поиска инцидентов за счет более эффективных механизмов обработки данных и событий безопасности.

Таким образом, можно констатировать, что важнейшей тенденцией в области мониторинга компьютерной безопасности и управления инцидентами безопасности является применение обработки больших данных, в частности технологий Hadoop и Spark.

Входными данными элементов обработки больших данных, применяемых для мониторинга компьютерной безопасности, могут являться базы данных (о программном и аппаратном обеспечении, уязвимостях, атаках, конфигурациях, слабостях программного и аппаратного обеспечения и др.), данные о ресурсах организации/предприятия из ERP-систем, системные логи, логи приложений, данные систем обнаружения и предотвращения вторжений (IPS/IDS), данные компонентов SIEM-систем, профили поведения пользователей и др. Из общего множества событий безопасности SIEM-система должна находить и использовать для принятия решений такие, которые свидетельствуют об атаках или иных нежелательных действиях в системе [31].

В качестве платформы для обработки неструктурированных данных может, например, использоваться Hadoop в режиме кластера. Кластер HDFS имеет два типа узлов, взаимодействующих по принципу Ведущий-Ведомый (Master-Slave): сервер имен (Namenode) и множество узлов серверов данных (Datanode). Сервер имен отвечает за пространство имен файловой системы. Он хранит дерево файловой системы и метаданные для всех файлов и директорий. Эта информация представляется в виде двух файлов: образа файловой системы и лога изменений. Для сервера имен также известны все узлы серверов, на которых хранятся блоки определенного файла. Примеры источников и результатов использования обработки больших данных для мониторинга компьютерной безопасности [13, 18, 20, 22, 25, 27, 29, 44, 66, 89]:

1. Источники данных (базы данных, журналы безопасности, данные SIEM-систем, логи VPN, профили.

2. Компоненты технологии:
   - HDFS (распределенное хранилище);
   - MapReduce (распределенная обработка);
   - YARN (распределенное хранилище).

Результаты обработки (отчеты, информационные запросы и ответы на них, инструментальные и индикаторные панели.

Клиент взаимодействует с файловой системой от имени пользователя с помощью узлов Namenode и Datanode. Клиент представляет собой интерфейс файловой системы, схожий с интерфейсом Linux Portable Operating System Interface (POSIX), поэтому пользовательскому коду не нужно ничего знать о Namenode и Datanode для функционирования.

В результате обработки данных, используя функции аналитической обработки, генерируются отчеты, формируются запросы и ответы на них, а также создаются инструментальные панели (англ. dashboard), позволяющие эффективно выполнять анализ инцидентов безопасности, и индикаторные панели, отображающие текущие инциденты, показатели безопасности, возможные контрмеры и др.

В отчете [47], опубликованном Cloud Security Alliance, приводятся примеры использования обработки больших данных для целей мониторинга компьютерной безопасности. С точки зрения сетевой безопасности, как уже было отмечено, второе поколение SIEM-систем позволяет значительно повысить эффективность процедур управления инцидентами безопасности. Организации собирают терабайты информации, касательно инцидентов (события сети, события приложений и действия пользователей) для множества целей, прежде всего для последующего анализа на предмет выявления угроз безопасности.

Современные компании все чаще могут позволить себе хранить большие объемы информации, используя преимущества виртуализации и современные технологии, включая хранилища данных дата-центров.

В среднем, такая организация как HP, генерирует по одному триллиону событий в день, что составляет порядка 12 миллионов событий в секунду. В случае использования публичных облачных сервисов (public cloud) организации сталкиваются с еще большей нагрузкой на систему в целом [45, 131]. К полученным данным компании применяют системы мониторинга компьютерной безопасности, но столь большой объем данных и нерациональная работа сенсоров (данные от которых, как правило, обновляются в определенный интервал времени) не всегда приводят к необходимым результатам.

Примеры систем мониторинга, основанных на больших данных.

В настоящее время активно ведутся исследования и разработки в области мониторинга компьютерной безопасности. Рассмотрим несколько примеров систем мониторинга, основанных на использовании обработки больших данных, разрабатываемых в ряде исследовательских и коммерческих проектов.

Одним из наиболее интересных проектов является проект Cisco OpenSOC, посвященный созданию центра управления инцидентами информационной безопасности (Security Operation Center, SOC).

В OpenSOC используются следующие инструменты больших данных: Apache Hadoop – для статистической обработки и анализа данных, Apache Flume – для сбора данных, Apache Kafka – для генерации сообщений, Apache Storm – для обработки сообщений в реальном времени, Apache Hive и HBase – для хранения данных, Elastic Search – для индексации данных [30, 114, 126, 52]. Общая архитектура OpenSOC представлена на рисунке 1.6.

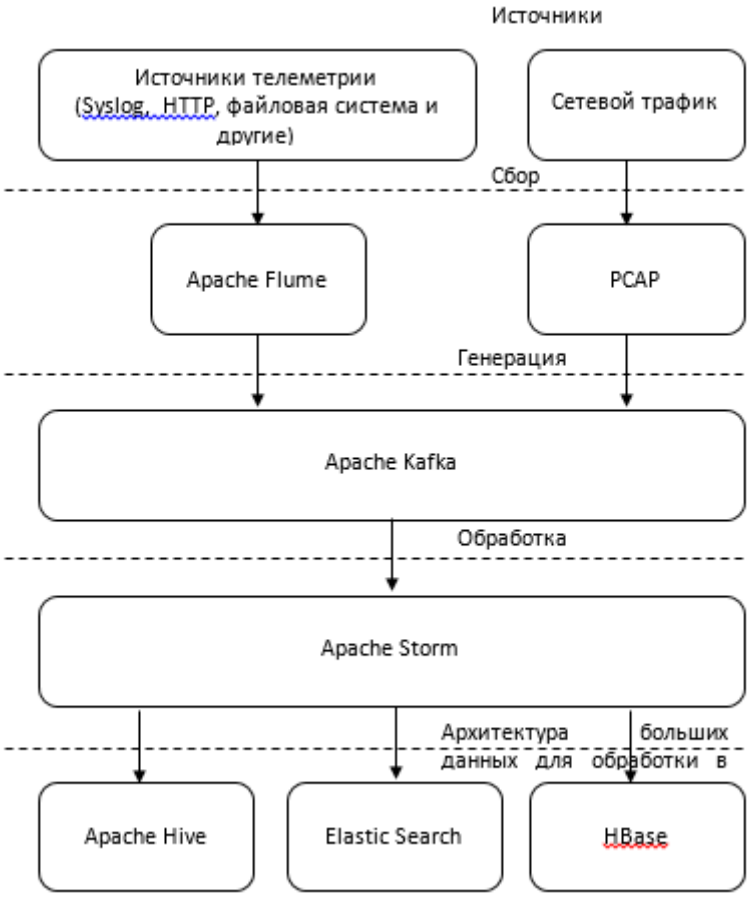

Рисунок 1.6 – Архитектура Cisco OpenSOC

Процесс функционирования OpenSOC состоит из следующих пяти фаз:

1. Данные собираются с источников телеметрии – информация о значениях измеряемых параметров, получаемых от элементов сети и включающих данные из логов, HTTP-запросов, файловой системы, коллектора NetFlow и др.

2. Данные собираются при помощи коллектора Apache Flume или в PCAP-файлы для последующей обработки. Основная задача Apache Flume – собрать данные из множества источников в централизованное хранилище.

3. Из коллектора данные обрабатываются системой обработки сообщений и разбиваются на соответствующие блоки, позволяя распределить централизованно собранный массив информации по категориям. Для этой цели в системе OpenSOC используется компонент Apache Kafka.

4. После параллельной обработки данные необходимо обработать. В системе Cisco OpenSOC для этой цели используется компонент Apache Storm. Потоковая обработка данных позволяет в режиме реального времени обрабатывать огромные массивы данных (свыше миллиона пакетов в секунду на сервер).

5. После параллельной обработки данные необходимо агрегировать, классифицировать и индексировать. В зависимости от задачи у разработчика есть несколько способов, которые можно использовать для решения поставленных задач. Apache Hive – инфраструктура, которая позволяет агрегировать данные, и представляет собой механизм генерирования запросов (аналогичный SQL), применительно к большим данным, который называется HiveQL. Elastic Search – платформа масштабируемой обработки данных в режиме реального времени. HBase – распределенная база данных для хранения структурированных массивов данных в виде больших таблиц. OpenSOC использует эту базу данных для формирования запросов к данным из больших наборов файлов.

В апреле 2015 года компания Cisco закрыла проект OpenSOC для общего доступа, приняв решение далее разрабатывать проект как закрытую платформу и продавать SOC-решение вместе с оборудованием своего производства. Однако, идеи, заложенные и реализованные в последней открытой версии OpenSOC, были положены в основу нового проекта компании Apache Metron. На рисунке 1.7 представлена архитектура Metron [56].

Рассмотрим последовательность функционирования этой системы.

На первом шаге все события, которые отслеживаются элементами телеметрии с устройств безопасности, собираются в компонент Apache Kafka.

На втором шаге каждое событие разбирается и нормализуется в стандартизованную структуру JSON. Для этого используются ключевые поля: IP-адрес источника пакета, IP-адрес хоста назначения, порт источника пакета, порт назначения, поле protocol id, поле timestamp, а также поле, содержащее тело сообщения (например, HTTP GET-сообщение). На этом шаге система может определенным образом отметить событие для его дальнейшей обработки.

После того как событие было обработано, реализуется шаг улучшения, включающий добавление дополнительной информации, которой система может дополнить уже обработанный трафик данных. Примером улучшения является добавление информации о местонахождении объекта (например, его координаты, город, страна или дополнительная информация о том, что IP-адрес хоста X соответствует web-серверам, находящимся в определенном дата-центре).

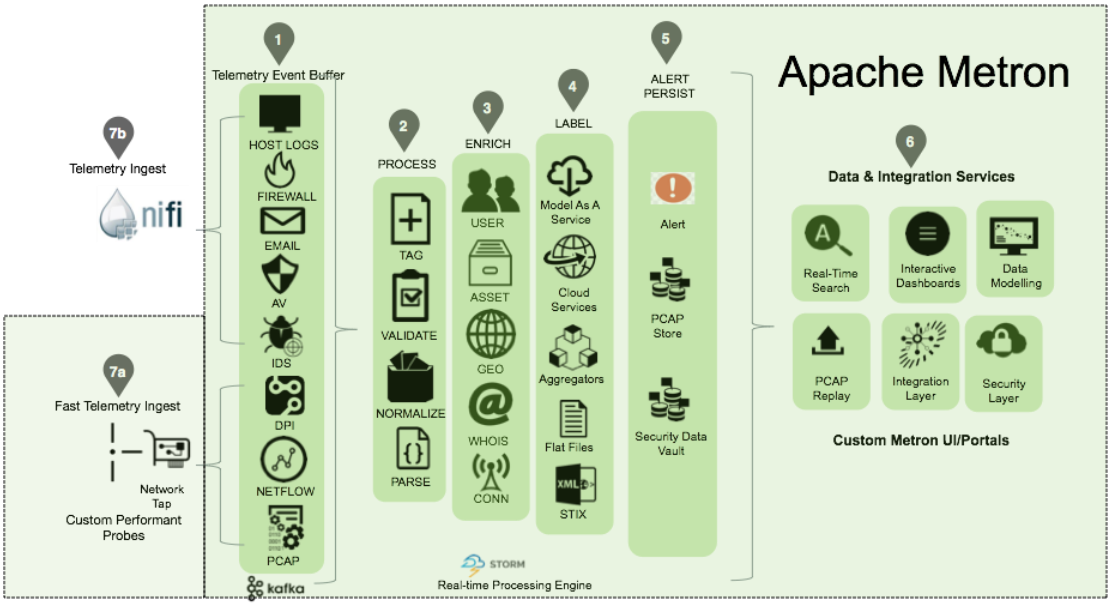

Рисунок 1.7 – Архитектура Apache Metron

На следующем шаге, после сбора дополнительной информации, выполняется интеллектуальный анализ данных, который заключается в том, что события телеметрии проходят анализ на выявление потенциальных угроз, используя, например, такое решение как Soltra [86] или другие агрегационные сервисы, анализирующие данные телеметрии.

После того как система проанализировала данные телеметрии, возможно появление сообщения тревоги, которое означает, что была обнаружена потенциальная угроза. Все данные о событиях телеметрии, дополненные и промаркированные, помещаются в систему Hadoop для длительного хранения и последующей обработки.

Для наглядного отображения инцидентов безопасности в рамках архитектуры Metron был разработан пользовательский портал, который позволяет выводить в реальном времени информацию об инцидентах безопасности. Также в систему Metron можно интегрировать дополнительные плагины, позволяющие производить моделирование данных, собранных в рамках задачи телеметрии.

Ожидается, что Metron может стать популярной масштабируемой платформой с открытым исходным кодом для решения задач безопасности с использованием обработки больших данных. Первая публичная версия была выложена в репозитории компании Apache в апреле 2016 года.

Еще одной системой в области систем мониторинга данных применительно к обработке больших данных является разработка компании Splunk Inc., которая получила название Splunk Enterprise. Данная система является одной из первых в классе КС, которая обеспечивает сбор данных и индексирование любых данных, получаемых с физических, виртуальных или облачных инфраструктур [103].

Процесс функционирования Splunk Enterprise можно разделить на несколько фаз:

1. Splunk начинает собирать всю информацию из разных источников и комбинирует полученную информацию в централизованную базу данных, используя индексы.

2. Используя индексы, Splunk способен быстро находить необходимые записи в журналах инцидентов безопасности и другой статистики, которая помещается в БД.

3. Система визуализации, встроенная в Splunk, позволяет отображать результаты запросов, которые может выполнять пользователь, используя SPL – специально разработанный синтаксис команд для активного поиска содержимого больших данных.

В ряде работ демонстрируется решение задачи анализа записей журналов инцидентов компьютерной безопасности на основе обработки больших данных. Особенностью разрабатываемой платформы является применение облачных технологий к решению задачи структурирования логов системы безопасности.

Используя коммерческое облако Amazon Cloud Environment, исследователям удалось значительно увеличить скорость обработки логов, основанных на HTTP-запросах, что еще раз подтверждает возможность эффективного решения задач обработки больших данных.

Архитектура разработанной системы состоит из следующих элементов:

1. Мастер-сервер, управляемый администратором, отвечает за взаимодействие с серверами обработки данных (slave nodes), а также серверами хранения (storage slave nodes). Мастер-сервер получает информацию о логах от серверов хранения данных, распределяет нагрузку между серверами обработки данных и выполняет обработку результатов вычислений.

2. Серверы обработки данных реализуют обработку логов, полученных от серверов хранения данных, получая инструкции от Мастер-сервера. При необходимости можно увеличить количество серверов обработки данных для повышения производительности, увеличивая параллелизм обработки данных.

3. Серверы хранения данных используются только для хранения данных логов. Возможны сценарии, когда серверы хранения объединяются с серверами обработки для анализа данных в случаях, когда не требуется выполнять комплексный анализ данных.

Схема анализа журналов инцидентов компьютерной безопасности на основе разработанного прототипа представлена на рисунке 1.8. Логика работы системы включает шесть шагов. Первый шаг – получение необходимых данных об инцидентах безопасности из серверов хранения данных. На втором шаге Мастер-сервер разделяет задачу на несколько подзадач и вычисляет параметры для распределения нагрузки между серверами обработки данных. На третьем шаге Мастер-сервер выбирает серверы обработки в соответствии с заранее заданным алгоритмом. На четвертом шаге каждый сервер обработки данных делает запрос на серверы хранения с целью получения необходимой информации для вычислений в соответствии с запросом Мастер-сервера. Следующий шаг состоит в анализе полученных данных, разделенном на два этапа: нормализация логов и

непосредственно сам анализ. Финальным шагом является сбор и обработка полученных данных от серверов обработки Мастер-сервером.

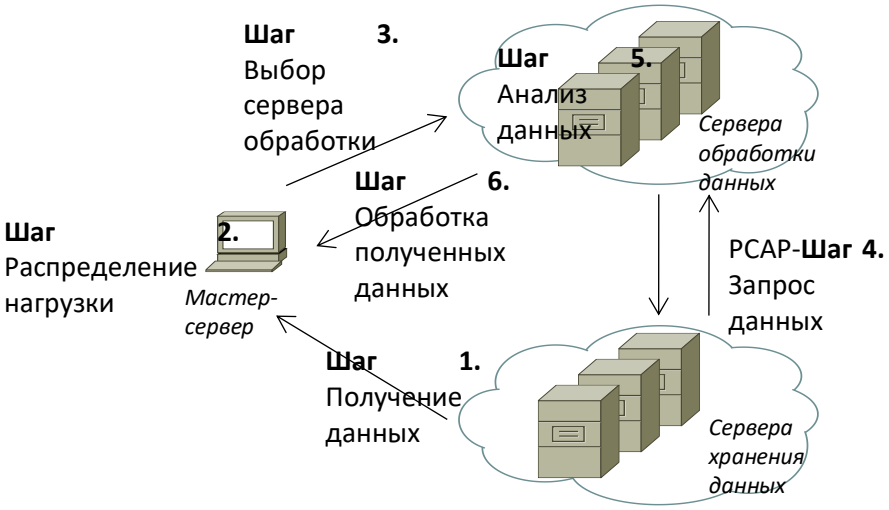

Рисунок 1.8 – Схема анализа журналов инцидентов компьютерной безопасности

В ряде работ рассмотрены вопросы мониторинга сетевого трафика на основе обработки больших данных. В статье отмечается развитие программно-конфигурируемых сетей и виртуальных инфраструктур, рост сложности обработки данных, появление новых приложений, значительное повышение производительности интерфейсов передачи данных. В связи с этим делается вывод о существенном усложнении задач проактивного мониторинга таких сетей и важности обеспечения масштабируемости предлагаемых решений. В качестве решения задач мониторинга данных в сети предлагается использовать подход, при котором функции обработки, хранения и анализа информации разделяются. Для обработки данных рассматривается архитектура, включающая как традиционную модель обработки данных, так и предлагаемую в работе интегрированную модель обработки данных. Генерируемые PCAP-файлы выгружаются в систему хранения данных, где обрабатываются в поточном режиме платформой Hadoop. Представленная модель позволяет добиться большей гибкости и эффективности обработки данных. PCAP (packet capture) – библиотека, которая позволяет

создавать программы анализа данных, поступающих на сетевую карту компьютера. Большинство приложений, использующихся для мониторинга и перехвата трафика, работают с форматом PCAP. Захваченный сетевой трафик сохраняется в формат файла PCAP, который позволяет впоследствии выполнить анализ сохраненных в нем пакетов.

На рисунке 1.9 представлена схема обработки данных, основанная на предложенной модели. Условно схему можно разделить на несколько составляющих: сетевой уровень (данные) – включает физические и виртуальные маршрутизаторы и фаерволы, серверы сбора информации/хранилища данных, платформа хранилищ (отражающих архитектуру больших данных), а также инструмент мониторинга VSS (с использованием процессора реального времени).

Условно работу системы можно разделить на три составляющих:

1. Выполняется захват и обработка данных с использованием выделенной сетевой инфраструктуры.

2. Сетевые инструменты анализируют сетевые пакеты (как в реальном времени, так и исторические записи БД).

3. Архитектура системы обработки больших данных обеспечивает поддержку виртуализации и использование вычислительных платформ, таких как Hadoop.

Рисунок 1.9 – Схема обработки данных при мониторинге сетевого трафика [53]

Разработка компании Symantec – The Worldwide Intelligence Network Environment (WINE) [87] – представляет собой платформу анализа данных значительного объема, собранных компанией Symantec (с использованием ключевых полей, включающих информацию о неизвестных исходных кодах, данные телеметрии антивирусного ПО, спам-сообщения, репутационные данные и примеры вредоносного ПО). WINE-платформа загружает и агрегирует данные, собираемые с миллионов хостов по всему миру, постоянно обновляя информацию, что позволяет исследователям получать доступ к актуальным данным.

На рисунке 1.10 представлена архитектура платформы WINE. Данные разделяются на несколько серверов хранения данных, подключенных непосредственно к серверам, которые выполняют задачи анализа данных. WINE регулярно обновляет данные от серверов Symantec, общий массив которых составляет 240 000 сенсоров, распределенных по всему миру. На основании полученных данных исследователи определяют те наборы данных, которые необходимы в конкретных исследованиях. После проведенных исследований

наборы данных архивируются на серверах хранения данных для сравнения с результатами, которые будут получены в будущем.

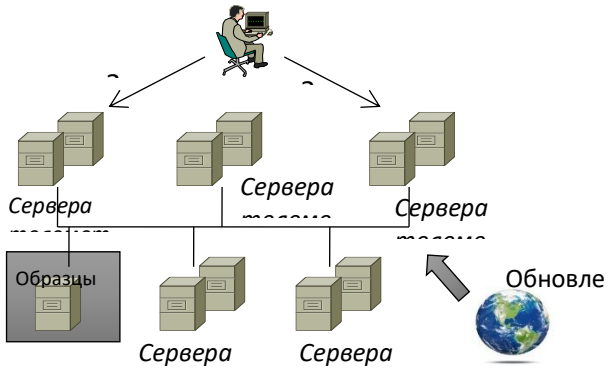

Рисунок 1.10 – Архитектура платформы WINE

Основными компонентами системы WINE являются: серверы телеметрии и репутации, которые агрегируют данные от сенсоров телеметрии, и серверы хранения данных, обеспечивающие обработку и хранение полученных данных пользователей. На основе постоянно обновляемой базы данных Symantec пользователь может смоделировать потенциальную атаку с использованием образцов вредоносного ПО, представленных на отдельном сервере. Для моделирования различных ситуаций WINE позволяет выполнять запросы в формате ANSI SQL и в среде MapReduce.

Организации все больше сталкиваются с так называемыми продвинутыми атаками (Advanced Persistent Threat, APT), когда атакующий, имея определенную заранее заданную цель по компрометации ресурсов, пытается скрыть свое присутствие в сети, используя комплексный подход к реализации атаки. Подход, описанный исследователями из корпорации AT&T, предполагает использование модели, называемой «пирамида атаки», которая базируется на построении деревьев атак. Предполагаемая схема детектирования атак группирует все инциденты безопасности, собранные организацией в контексты, после чего, используя различные алгоритмы, детектирует возможную подозрительную активность в рамках каждого контекста и между ними, используя парадигму MapReduce.

На рисунке 1.11 представлена обобщенная архитектура распределенных вычислений на базе MapReduce [70, 84, 107, 156, 185], призванная решить проблему масштабируемости при определении атак в сети организации. Как показано на рисунке, на первом этапе данные об инцидентах безопасности собираются со всех возможных источников (сетевые сенсоры, журналы инцидентов безопасности, системы IPS/IDS и др.). Далее, используя операцию Reduce 1, вычислительный сервер MapReduce берет в качестве входных параметров пары (цель, событие), а в качестве выходных данных – список с целью и набором всех событий, принадлежащих к контексту конкретной цели. Наконец, операция Reduce 2 применяется к каждой цели и набору событий, принадлежащих к соответствующему контексту. Выходом функции Reduce 2 является список целей и все потенциальные угрозы, детектируемые алгоритмом обнаружения вредоносного ПО. Результат посылается системе управления инцидентами, которая принимает решение – нужно ли генерировать сообщение тревоги, или нет.

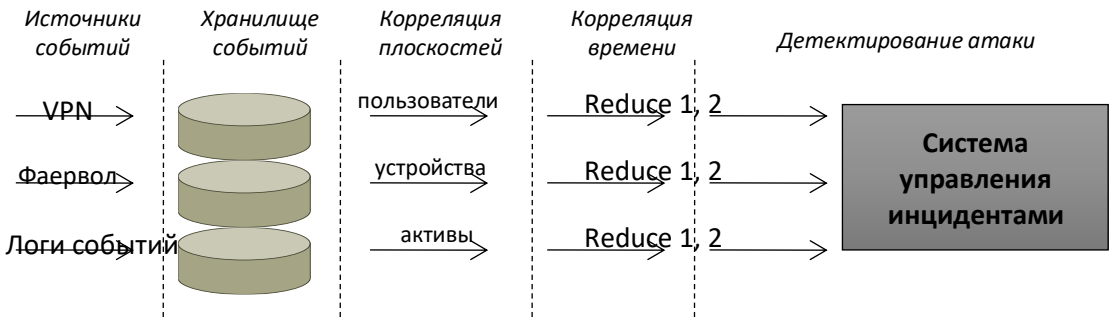

Рисунок 1.11 – Архитектура распределенных вычислений на базе MapReduce

В настоящее время появляется все больше реализаций систем мониторинга, основанных на больших данных. В данном диссертационном исследовании мы ограничились рассмотрением наиболее известных примеров таких решений.

В таблице 1.2 приведены сравнительные характеристики нескольких рассмотренных решений.

Для сравнения были выбраны следующие ключевые параметры:

1. *Объем обрабатываемых данных*. В исследуемых работах были приведены выборки данных объемом от 100 Мб, заканчивая 3,5 экзабайт (10^{18}).

2. *Количество поддерживаемых серверов*. В реализованных системах вполне естественным является то, что увеличение количества серверов приводит к уменьшению времени работы системы в целом за счет распределения нагрузки между серверами [12].

3. *Скорость обработки.* Общая скорость обработки данных зависит от объема обрабатываемого трафика [12] и возможности использования массивов жестких дисков для балансировки нагрузки трафика.

4. *Способ обработки – потоковая/пакетная*. Большинство представленных решений используют поточную обработку данных, как более перспективную, так как она позволяет в случае применения обработки больших данных для мониторинга инцидентов достичь большей скорости формирования решений. Пакетная обработка данных используется только в тех случаях, когда заявленная реализация либо не была изначально предназначена для поточной обработки, либо использовалась в качестве сравнения производительной мощности прототипа.

5. *Цели и задачи, решаемые системой*. Как видно из таблицы 1.2 цели и задачи, которые ставили перед собой исследователи, разрабатывая системы обработки информации с использованием обработки больших данных, весьма обширны. Все разрабатываемые решения объединяет единая цель – разработать архитектуру, которая смогла бы удовлетворять требованиям обработки больших потоков информации и оперативного обнаружения инцидентов безопасности.

Таблица 1.2 Сравнительные характеристики рассмотренных решений

Рассматриваемая система	**Объем обрабатываемых данных**	**Количество поддерживаемых серверов**	**Скорость / время работы**	**Способ обработки данных**	**Цели и задачи, решаемые системой**
Massive Distributed and	До 500 Мб	До 8 slave-серверов	450 секунд при	Потоковая	Разработка архитектуры

Parallel Log Analysis For Organizational Security [166]			обработке 500 Мб трафика на 8 серверах.		распределенной обработки журналов инцидентов безопасности
VSS Monitoring. Leveraging a Big Data Model in the Network Monitoring Domain [122]	Более 3.5 экзабайт	Данные не предоставлены	100 Гбит/с	Потоковая Пакетная	Разделение сетевой аналитику и системы хранения Получение большей эффективности за счет интеграции используемого оборудования, инфраструктур обработки больших данных
Toward a Standard Benchmark for Computer Security Research. The Worldwide Intelligence Network Environment [87]	Более 100 Тб	240 000 сенсоров по всему миру	Данные не предоставлены	Потоковая Пакетная	Обработки данных с миллионов хостов с использованием ключевых полей безопасности

Using Large Scale Distributed Computing to Unveil Advanced Persistent Threats [94]	Репрезентативная выборка составила 74 гигабайта данных о 144 миллионах событий	Один физический сервер с 16 ядрами процессора	Время обработки информаци - 1500 сек.	Пакетная обработка данных	Интеграция всех инцидентов безопасности, детектирование подозрительной активности

1.3. Анализ методик и алгоритмов обнаружения инсайдеров в компьютерных сетях

Рассмотрим перечень релевантных работ в области информационной безопасности для обнаружения или категорирования инсайдеров [82, 46, 76, 97, 98, 125, 136, 143] по следующим категориям:

– SIEM (Security Information and Event Management) системы и системы обнаружения атак (Intrusion Detection Systems, IDS) и противодействия атакам (Intrusion Prevention Systems, IPS);

– machine learning алгоритмы;

– системы User and Entity Behavioral Analytics (UBA/UEBA);

– rule-based архитектуры;

– решения, основанные на обработке больших данных.

SIEM системы и системы обнаружения атак и противодействия атакам.

В [48] описывается решение, которое собирает внешнюю информацию о вредоносных IP-адресах, занесенных в публичные черные списки, и внутреннюю информацию организации об инцидентах безопасности для расчета показателей репутации для внешних IP-адресов и публичных черных списков IP-адресов. Оценка репутации используется правилами SIEM для выбора типа оповещения для

каждого IP-адреса, подлежащего мониторингу. Представленное решение направлено на расширение охвата SIEM деятельности по борьбе с киберпреступностью в сети организации.

В исследовательской статье [71] описывается разработка и предлагаемое применение сигнатуры управления информацией и событиями безопасности (SIEM) для обнаружения возможной вредоносной инсайдерской деятельности, ведущей к саботажу ИТ. В отсутствие единого стандартизированного формата регистрации событий в данной статье подпись представлена в двух наиболее заметных публичных форматах: Common Event Framework (CEF) и Common Event Expression (CEE). Цель сигнатуры – определить личность злоумышленника, а также то, какой протокол удаленного подключения используется им, и происходит ли эта деятельность вне рабочего времени. Идентификация злоумышленника может быть получена с помощью следующих параметров: имя пользователя, имя VPN аккаунта или хоста. Протоколом удаленного доступа может быть: SSH (Secure Shell), Telnet или RDP (Remote Desktop Protocol). Подпись была основана на следующих ключевых полях: имя пользователя, имя VPN аккаунта, имя хоста атакующего, а также признак того, используется ли SSH, Telnet или RDP.

В работе [90] предлагается решение для проблемы инсайдеров, используя концепции поведенческой теории, профилирования личности и аудита цифровых следов. Вместо изолированного подхода было рассмотрено пересечение различных областей риска и агрегированные показатели риска по каждой из них как фактор, предсказывающий вредоносную инсайдерскую деятельность. Авторы описывают аналитическую модель, учитывающую элементы риска из различных рискованных доменов. Обработка каждого домена по отдельности приводит к недостаточным доказательствам злого умысла. Однако, когда пересечение различных индикаторов риска рассматривается как единый блок, оно предлагает значительное улучшение возможности обнаружения инсайдерской угрозы.

Как представляется, не существует решения, способного полностью устранить внутреннюю угрозу внутри организации. Кроме того, технический подход сам по себе может оказаться не самым эффективным способом

предотвращения и/или обнаружения вредоносных внутренних угроз. Одним из перспективных подходов к повышению эффективности и результативности SIEM-систем является использование методов машинного обучения.

Machine Learning алгоритмы. Алгоритмам машинного обучения (Machine Learning Algorithms) посвящено достаточно большое количество исследований. В работе [49] предлагается использовать специальный фреймворк для определения аномалий в компьютерной сети. В качестве входных данных многомерные входные данные, такие как логи взаимодействия пользователей с аппаратными средствами, записи веб-доступа и электронные письма. Для выявления взаимосвязей между многомерными объектами используются графы. Взаимодействие пользователя с устройствами иллюстрируется взвешенным неориентированным двумерным двусторонним графом G = (V; E; W), где V – множество вершин (пользователи), E – множество ребер, а W – вес ребер. Набор вершин состоит из двух типов объектов – пользователей и устройств, в то время как ребра представляют взаимодействие пользователя с устройством.

Схема работы фреймворка представлена на рисунке 1.12. Фреймворк состоит из двух основных компонентов: «Блок графической обработки(GPU – Graphical Processing Unit) и «Блок обнаружения аномалий» (ADU – Anomaly Detection Unit). Данные, полученные от разнообразных источников компьютерной сети, форматируются и подаются в графический процессор, который генерирует граф, представляющий взаимосвязи между узлами сети. Эти входные потоки могут быть из различных информационных источников с разными форматами данных. Например, данные могут быть из журналов событий (вход / выход из системы), журналов электронной почты, записей HTTP, данных доступа к социальной сети и различных записей персонала, таких как психометрические данные. На основе этой информации для каждого пользователя рассчитываются параметры графа. Поскольку конечной целью является изоляция наиболее аномальных пользователей от остальных пользователей, все атрибуты рассчитываются индивидуально для каждого пользователя. Еще одной задачей GPU является генерация порожденных подграфов каждого пользователя для разных уровней

подграфов. Некоторые свойства, такие как число вершин, число ребер, плотность, диаметр и количество пиров (Peers), рассчитываются для каждого уровня подграфов. Рассчитанные параметры графа и подграфа подаются в ADU. Параллельно с вышеуказанным процессом изменяющиеся со временем данные также подаются в ADU. Алгоритм изолирующего леса (Isolation Forest) выполняется для обнаружения аномальных пользователей в блоке ADU, а в качестве выходных данных ADU выступает оценка аномальности каждого пользователя. Эти значения используются для обнаружения и отделения возможных злонамеренных пользователей от остальных работников.

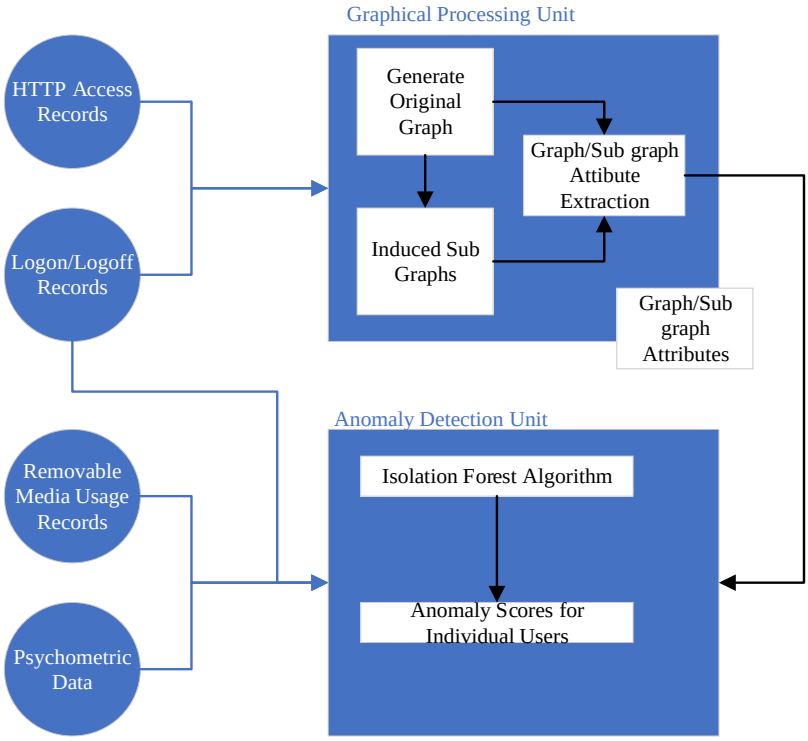

Рисунок 1.12 – Схема работы фреймворка обнаружения аномалий [49]

В качестве набора данных (data set) в [49] был использован набор, опубликованный компьютерной группой реагирования на чрезвычайные ситуации университета Карнеги-Меллона, – файл «R4.2.tar.bz». [190] Этот набор данных содержит информацию об имени пользователя, имени компьютера, URL запросы с временными метками сессий HTTP, данные о входе в систему, используемые устройства, список измененных файлов от 1000 сотрудников за более чем 17-месячный период времени.

В качестве основного алгоритма выявления аномалий выбран алгоритм Isolation Forest. Принцип работы алгоритма Isolation Forest заключается в следующем: проводится случайное разбиение пространства признаков, – такое, что в среднем изолированные точки отсекаются от нормальных, кластеризованных данных. Окончательный результат усредняется по нескольким запускам алгоритма. Суть алгоритма заключается в построении случайного бинарного решающего дерева. Корнем дерева является все пространство признаков; в очередном узле выбирается случайный признак и случайный порог разбиения, выбранный из равномерного распределения на отрезке от минимального до максимального значения выбранного признака. Критерием остановки работы алгоритма является тождественное совпадение всех объектов в узле, в этом случае решающее дерево строится полностью. Ответом в листе, который также соответствует рейтингу аномалий алгоритма, объявляется глубина листа в построенном дереве. Утверждается, что аномальным точкам свойственно оказываться в листьях с низкой глубиной, то есть в листьях, близких к корню, когда для разбиения гиперплоскостями кластера нормальных данных дереву потребуется построить еще несколько уровней. При этом количество таких уровней пропорционально размеру кластера, следовательно, пропорционально и рейтингу аномалий для лежащих в нем точках. Это означает, что объекты из кластеров малых размеров, которые потенциально являются аномалиями, будут иметь рейтинг аномалии ниже, чем из кластеров нормальных данных.

В [75] представлен общий формальный фреймворк для проведения анализа на предмет выявления вредоносных внутренних угроз, который основан на вероятностном моделировании, методах проверки и синтеза. На первом этапе фреймворк, с помощью байесовских сетей, определяет намерение инсайдеров осуществить внутреннюю атаку, а на втором этапе рассчитывает вероятность успеха внутренней атаки, используя для этого вероятностную проверку моделей. Данный подход фиксирует поведение инсайдеров и моделирует как их намерение или риск превращения в злоумышленников, так и риск того, что действия инсайдеров увенчаются успехом.

В [141] была разработана модель обнаружения внутренних угроз, используемая организациями, которые регулярно повторяют задачи через равные промежутки времени. Такими организациями являются военные или государственные учреждения. Данная модель, специально оптимизированная для организации, оценивает каждую комбинацию с точки зрения точности, AUC (площади под кривой) и TPR (истинного положительного коэффициента).

В [121] авторы изучают и оценивают работу по обнаружению внутренних угроз с использованием контролируемых и неконтролируемых алгоритмов обучения. С этой целью они изучают и анализируют данные, а также обнаружение аномалий и классификацию вредоносного поведения на основе общедоступного набора данных. В работе приводится оценка нескольких контролируемых и неконтролируемых алгоритмов обучения – HMM, SOM и DT. Основываясь на экспериментальных результатах, авторы делают вывод о том, что SOM обеспечивает лучшие результаты с точки зрения DR, FPR и поддержки человеческих аналитиков посредством визуализации данных.

В работе [77] рассматривается модель специализированного обнаружения аномалий в сети (Specialized Network Anomaly Detection, SNAD). В качестве основного алгоритма используются алгоритмы машинного обучения, базирующиеся на модели работы без учителя в рамках изучения данных электронных медицинских карт [154]. Принцип выявления аномалий заключается в сопоставлении объектов и выявлении различий между их характеристиками. Из имеющихся объектов составляется граф, элементы которого представляются бинарными матрицами, в которых 1 соответствует получению доступа к субъекту, а 0 – отказу доступа к субъекту [83, 138, 188]. Для сопоставления таких матриц используется косинусное сходство, работающее с векторными представлениями объектов. Структура SNAD состоит из двух компонентов: измерения сходства (SNAD-SM) и оценки аномалии (SNAD-AE).

В качестве набора данных был использован “EHR dataset” – набор данных онлайн-карточек пациентов больниц, содержащий сведения о 6015 пользователях

и 130457 пациентах, собираемых на протяжении 30 недель, а также данные логирования.

Для обнаружения инсайдеров авторы [65] исследовали перспективу внедрения неконтролируемой системы машинного обучения. Система оказалась склонна генерировать высокий процент ложных срабатываний. Тем не менее, путем дальнейших улучшений она может быть использована для оказания помощи судебно-медицинской экспертизе за счет снижения числа подозреваемых.

В [88] авторы представляют среду для развертывания имитируемых на месте пользовательских ботов (SUB), которые могут эмулировать действия реальных пользователей. Создав учетную запись пользователя и запустив хост в сети предприятия, SUB может быть введен в компьютерную сеть, которая работает реалистично и не мешает нормальной работе. Внедрение вредоносного поведения в SUB должно быть обнаружено инфраструктурой мониторинга внутренних угроз. Структуру SUB можно контролировать, чтобы исследовать границы развернутых систем и проверить эффективность тактики уклонения от инсайдеров.

В [68] предлагается подход, который сочетает в себе обнаружение структурных аномалий (SA) из социальных и информационных сетей и психологического профилирования (PP) отдельных лиц. SA использует технологии, включая анализ графиков, динамическое отслеживание и машинное обучение для обнаружения структурных аномалий в крупномасштабной информационной сети, в то время как PP создает динамические психологические профили из поведенческих паттернов. Угрозы выявляются путем объединения и ранжирования результатов SA и PP.

UBA/UEBA системы. Системы класса UBA и UEBA появились относительно недавно и помогают в обеспечении безопасности как внутренней, связанной с поведением сотрудников, так и внешней, например, способствуют предотвращению атаки посредством компрометации учетной записи законного пользователя. Сама аббревиатура UEBA прозвучала впервые в отчете компании Gartner в середине 2016 года [62]. Именно в этих отчетах компания советует использовать анализ поведения для выявления аномалий и предотвращения угроз.

UEBA/UBA-системы являются следующим поколением систем, которые позволяют выявлять неизвестные типы угроз, внутренних нарушителей и целевые атаки. Опираясь только на поведенческий анализ, эти системы способны определять аномалии и неочевидные взаимодействия пользователей с КС, а это, в свою очередь, предоставляет администраторам возможность своевременно реагировать на потенциальные угрозы.

В данный момент на рынке существует большое количество различных UBA- и UEBA-систем, установка которых возможна, как отдельных модулей с двухсторонней интеграцией с базой данных управления конфигурацией. Стоит учитывать, что для хранения и применения изменений в режиме близком к реальному времени потребуются большие мощности, так как система активно использует распределенные высоконагруженные платформы хранения данных [30, 114]. Перед интеграцией системы должно выполняться профилирование поведения сетевой инфраструктуры. В зависимости от ее размера, время интеграции может достигать нескольких недель.

На мировом рынке UEBA/UBA-системы представлены как в виде отдельных модулей, так и как часть систем управления информационной безопасностью. Например, решение от Splunk, основанное на UEBA Cspida, интегрировано с SIEM-решением Enterprise Security, HPE имеет свое решение – ArcSight UBA, а Microsoft – Microsoft Advanced Threat Analyics. Российские компании также поставляют свои UEBA/UBA-системы. Лаборатория Касперского имеет решение для определения скомпрометированных аккаунтов пользователей и конечных станций Kaspersky Fraud Prevention. Определять и предотвращать инсайдерские угрозы способно решение Контур информационной безопасности (КИБ) от SearchInform.

Системы UEBA/UBA служат для анализа поведения пользователей, используя алгоритмы машинного обучения и статистического анализа. Они позволяют строить модели поведения пользователей и определять отклонения от этих моделей как в режиме реального времени, так и обращаясь к уже накопленным данным. Источником данных для систем UEBA/UBA являются: журналы серверов и сетевых устройств, SIEM-систем, локальные журналы с персональных

компьютеров, данные из систем аутентификации [162], содержание переписки в социальных сетях, мессенджерах и почтовых сообщениях [167].

Подобная аналитика может быть применена, например, в следующих ситуациях (в скобках указана необходимая функциональность системы для обнаружения нарушителей) [23]:

– компрометация учетной записи обычного пользователя (необходимо обнаруживать факт получения злоумышленником контроля над учетной записью пользователя независимо от того, каким образом планируется это сделать);

– компрометация учетной записи администратора (пользователи с правами администратора могут не работать по стандартным шаблонам поведения, что усложняет аналитику; однако существует возможность определять некоторые типы атак на пользователей, обладающих доступом к критической информации);

– внутренние угрозы (если пользователь совершает операции, выходящие за рамки обычного профиля поведения, то необходимо обнаруживать такие ситуации);

– совместное использование учетных записей (использование одних и тех же учетных записей несколькими пользователями представляет опасность для безопасности, поэтому такие случаи должны идентифицироваться с указанием пользователей, совместно использующих один аккаунт);

– классификация сервисных учетных записей (предполагается автоматическое определение сервисных аккаунтов и установка отметки на них при обнаружении необычного поведения в пределах таких аккаунтов);

– неактивные учетные записи (если пользователь не входил в свой аккаунт в течение определенного времени, то возможно, что он покинул организацию и процесс деактивации учетной записи не был проведен до конца, поэтому должна обеспечиваться возможность наблюдения за сотрудниками, не использующими свои учетные записи за установленное время);

– контроль установки и удаления программ (необходимо обеспечить мониторинг появления необычного программного обеспечения (ПО) на устройстве

пользователя или удаления антивирусного обеспечения, что является прямой угрозой информационной безопасности);

– расследование нарушений инцидентов безопасности (расследование инцидента, связанного с утечкой данных или взломом сети, может занимать много времени, поэтому необходимо обеспечивать возможность обработки больших объемов данных для анализа действий всех пользователей, устройств и сервисов с течением времени в сети организации, затронутых инцидентом).

Системы класса UBA/UEBA – важный элемент по выявлению угроз, целевых атак, а также сотрудников, нарушающих внутренние правила информационной безопасности внутри компании [165]. UBA/UEBA системы нацелены на решение следующих основных задач:

– простая и расширенная аналитика в режиме реального времени;

– оперативное выявление аномалий в сети;

– определение значимости события;

– ответная реакция на события, за счет того, что администраторы имеют комплексную информацию об инциденте.

UBA/UEBA-системы становятся особенно популярны в наши дни и их интеграция в SIEM, DLP и другие системы различных производителей доказывает этот факт.

Следует также выделить ряд задач по интегрированному использованию UBA и UEBA-систем с другими механизмами защиты, например, их интеграцию с серверами аутентификации, которые в случае определения инсайдеров незамедлительно закрывают им доступ [15].

В статье [106] предложен подход, основанный на динамическом моделировании. Этот подход использует так называемую модель интеграции информации, схематично представленную на рисунке 1.13.

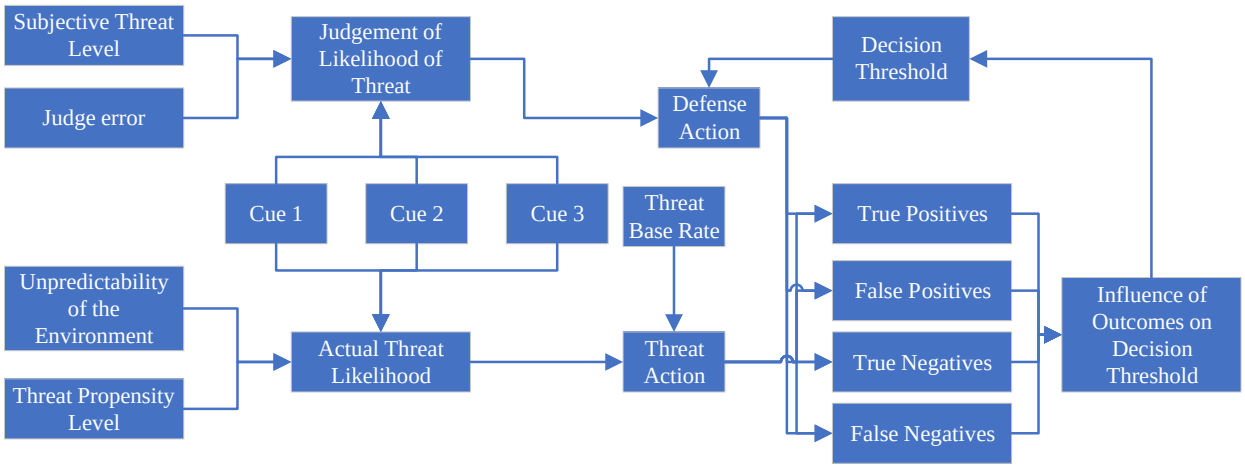

Рисунок 1.13 – Модель интеграции информации

В модели поток транзакций входит в организацию, и информационные работники обрабатывают эти транзакции. Сделки, которые осуществляет фирма, являются либо «хорошими» (отсутствует вредоносное воздействие), либо «плохими» (что указывает на возможное наличие мошенничества или ошибки). Не существует абсолютного теста на несоответствие, где «положительный» результат всегда правильно указывает на мошенничество, а «отрицательный» результат всегда правильно указывает на отсутствие мошенничества. Одной из задач, которые решают работники, является задача вынесения суждений о подозрительных транзакциях. Работники должны использовать свои знания, чтобы решить, следует ли расследовать транзакции. Они выносят суждения о транзакциях, интегрируя информацию, поступающую из разных сообщений, присутствующих в потоке транзакций (см. рисунок 1.13, Cue 1, Cue 2 и Cue 3), учитывая сложность транзакций, количество возвратов, связанных с транзакцией, и объем транзакций. В результате суждения о вероятности угроз сравниваются с порогами принятия решений, вследствие чего защитные меры либо принимаются, либо – нет. Полученные решения, в свою очередь, становятся убеждениями, которые в сочетании с организационными стимулами влияют на значения этих порогов принятия решений.

В [160] авторы разработали платформу, которая обнаруживает подозрительных инсайдеров, используя психологический триггер, который

побуждает злонамеренных инсайдеров вести себя подозрительно. Кроме того, была предложена архитектура, состоящая из диктора, монитора и анализатора. Во-первых, диктор создает событие (называемое «триггер»), которое побуждает злоумышленников вести себя подозрительно. Затем мониторы записывают подозрительные действия, такие как удаление файлов, электронной почты. Наконец, анализатор выявляет подозрительных инсайдеров, сравнивая количество удалений до и после триггера.

В [72] авторы предлагают систематическую структуру, которая использует контекстные знания о системе и ее пользователях, извлеченные из данных, собранных из системы контроля доступа в здание, для выбора подходящих моделей представления поведения движения. Затем авторы исследуют использование изученных моделей в режиме онлайн, а также информацию о планировке здания, за которым ведется мониторинг для обнаружения вредоносного внутреннего поведения. В результате работы приводится эффективность разработанной структуры, используя реальные данные, отслеживающие движение пользователей на железнодорожных транзитных станциях.

В [112] авторы предлагают методы обнаружения внутренних угроз, основанные на моделировании поведения пользователя и алгоритмах обнаружения аномалий. Основываясь на данных журнала пользователя, были созданы три типа наборов данных: сводка ежедневной активности пользователя, тематическое распределение содержания электронной почты и еженедельная история сообщений пользователя по электронной почте. Затем были применены четыре алгоритма обнаружения аномалий и их комбинации для обнаружения вредоносных действий. Результаты экспериментов показывают, что предлагаемая структура может хорошо работать в отношении несбалансированных наборов данных, в которых имеется лишь несколько внутренних угроз.

В [183] описывается методика обнаружения внутреннего нарушителя с использованием системы анализа пользовательского поведения. Созданная в рамках данной статьи платформа обладает следующими возможностями: сбор и обработка системных логов и логов приложений, извлечение записей активности

каждого пользователя из логов, создание векторов активности для каждого пользователя на основе его предыдущей активности, обнаружение аномалий. Кроме того, представлен ансамбль из нескольких алгоритмов, объединяющий OKCBM, PHHH и iForest. Эксперимент, приведенный в статье, показал, что система с ансамблем неконтролируемых алгоритмов обнаружения аномалий может обнаруживать аномальные модели поведения пользователя. В результате эксперимента определено, что OCSVM и PHHH страдают от аномалий в тренировочном комплекте, а iForest дает больше ложных срабатываний, тогда как ансамбль из трех алгоритмов имеет высокую производительность, достигающую 96,55%, и точность запоминания – в среднем 91,24%.

В работе [99] представлена методика обнаружения инсайдеров на основе регистрации событий командной соревновательной игры. Для проведения экспериментов рабочие станции были настроены со стандартным программным обеспечением (MS Office, Mozilla Firefox и Microsoft Outlook). Каждому участнику для внутренних коммуникаций был предоставлен личный адрес электронной почты. Все учетные записи участников были настроены так, чтобы не иметь прав администратора (не разрешалось устанавливать какие-либо новые программы или изменять конфигурации). В архитектуре все машины управлялись сервером контроллера домена Windows.

Каждая рабочая станция пользователя была настроена для запуска трех агентов, которые регистрировали системные вызовы, действия мыши и клавиатуры. Агенты мыши и нажатия клавиш были запрограммированы на языке Python и использовали библиотеку pyinput. Агент монитора хоста отвечал за регистрацию системных вызовов, генерируемых каждым Amazon WorkSpace. В качестве агента монитора хоста был выбран Process Monitor, поскольку это стандартный инструмент Windows для криминалистического и системного анализа. Файловый сервер служил хранилищем для сбора файлов журналов с хост-компьютеров. Журналы, хранящие информацию о поведении мыши и нажатиях клавиш, имели малый размер и обновлялись медленно, – для них был открыт прямой сетевой доступ к файловому серверу. С другой стороны, журналы,

генерируемые Host Monitor, были массивными и обновлялись очень быстро. Следовательно, чтобы предотвратить создание больших сетевых буферов и сохранить пропускную способность, журналы сжимались и отправлялись на файловый сервер каждый час. Для перехвата сетевого трафика с рабочих станций была использована атака класса «человек посередине». Работоспособность Network Proxy была обеспечена установкой сертификата в список доверенных сертификатов рабочей станции, из-за чего HTTPS-трафик мог быть перехвачен и расшифрован. Весь сетевой трафик был захвачен и упакован в формат PCAP.

В качестве исходного набора данных исследователи использовали сгенерированную в ходе работы пользователей на виртуальных машинах Amazon AWS последовательность данных, содержащую пользовательскую активность при учете трех факторов: действий мыши и клавиатуры, сетевой активности и данных об отправке e-mail сообщений. Такой подход, по мнению авторов, позволяет определить, работал ли пользователь физически за устройством или нет.

В [109] авторы объединяют формальное моделирование и анализ инфраструктур организаций с социологическим объяснением, чтобы обеспечить фреймворк для анализа внутренних угроз. Для поддержки этой структуры они используют помощника по логике Isabelle/HOL. Была получена характеристика способности инсайдеров выдавать себя за других на основе проверки протокола. Использование логики высшего порядка позволяет выразить психическую предрасположенность человека, например, мотивацию или психологическое состояние. Таким образом, поведение человека можно смоделировать для внутренних угроз в соответствии с тремя этапами социологического объяснения Вебера [171, 186].

Исследование [135] показывает возможность классифицировать и переклассифицировать пользователей по ролям рабочих групп, предоставляет возможность обнаружения и определения нормального или ожидаемого поведения рабочих групп. Исходная система классификации определяется на основе диаграмм вариантов использования. Первоначально разработанные диаграммы вариантов использования могут быть расширены для отражения действующих лиц

и их активности в системе. В работе рассмотрена организация с определенными пользовательскими ролями: юриспруденция, инжиниринг, маркетинг и продажи. Все эти пользователи могут совместно использовать общие ресурсы, включая серверы электронной почты, принтеры, базы данных клиентов, сканеры, копиры, факсимильные аппараты, веб-серверы, мобильные телефоны, маршрутизаторы, брандмауэры, базы данных патентов и точки беспроводного доступа, а также многие другие. Кроме того, предполагается, что пользователи могут действовать как внутри, так и за пределами границы организации.

Визуализация графов дает возможность классифицировать и переклассифицировать пользователей по ролям рабочих групп, что позволяет выявлять и определять нормальное или ожидаемое поведение рабочих групп. Для того чтобы идентифицировать пользователей, которые действуют за пределами своих ролей в рабочих группах, системы регистрации должны быть сконфигурированы на запись действий пользователей и их взаимодействия друг с другом. Для каждого идентифицированного действия должны быть определены связанные ресурсы, необходимые для выполнения задачи, а также методы для регистрации взаимодействия с ресурсами.

В качестве набора данных были использованы файлы журнала событий вычислительной системы по активности пользователя, а именно: изменение файлов системы, получение доступа к важным данным, отправление почты, получение доступа к принтеру и печать.

Rule-based архитектуры. В [127] рассматриваются различные подходы к постоянной оценке (CE) для выявления внутренних угроз, которые доступны правительству США, и оценивается актуальность этих подходов для ситуаций, создаваемых такими внутренними угрозами. В отчете CE определяется процесс проверки и вынесения решения, позволяющий на постоянной основе проверять личность, которая была определена как имеющая право на доступ к секретной информации или занятие должности.

В [176] представлен метод обнаружения аномального поведения путем профилирования пользователей. Авторы используют алгоритмы оценки k-средних

и плотности ядра, чтобы изучить нормальное поведение пользователя и установить нормальные профили пользователей на основе поведенческих данных. Затем они сравнивают поведение пользователя с обычными профилями, чтобы выявить ненормальные модели поведения.

В [173] авторы представляют систему, которая объединяет структурную и семантическую информацию из реальной корпоративной базы данных отслеживаемой активности на компьютерах своих пользователей для обнаружения независимо разработанных групповых вставок вредоносной инсайдерской деятельности. Также они разработали и применили несколько алгоритмов для обнаружения аномалий, основанных на предполагаемых сценариях злонамеренного поведения инсайдеров, индикаторах необычных действий, многомерных статистических шаблонах, временных последовательностях и эволюции нормальных графов [85, 101]. Представленная работа демонстрирует возможность выявления слабых сигналов, характерных для IT, с использованием нового набора алгоритмов и методов.

В [177] выполнен аналитический обзор и произведена структуризация публикаций по проблемам обнаружения инсайдеров. Согласно результатам обзора, исследования в основном осуществляются в области снижения инсайдерских угроз, теоретических оценок и обоснований, управления инсайдерскими угрозами, исследований поведения инсайдеров при создании угроз, обзора инсайдерских угроз и публикаций различного характера, которые не могут быть явно отнесены к одной из категорий, предложенных для структуризации.

Как уже упоминалось, все рассмотренные работы анализируют проблему обнаружения инсайдеров и противодействия инсайдерским атакам в отдельных направлениях и предлагают обобщения и структуризации, пригодные для выработки рекомендаций общего характера, и на их основе – внедрения более конкретных решений администраторами SIEM-систем, то есть с участием человека-специалиста по безопасности. По мере развития и распространения сетевых технологий, вне всякого сомнения, будут постоянно возникать как новые мотивации для инсайдеров, так и угрозы. Это означает, что таксономия инсайдеров

и инсайдерских угроз носит динамический характер, и, хотя в силу ограничений Тьюринга, модель для обнаружения и противодействия инсайдерам не может быть сложнее процесса, она тем не менее будет усложняться по мере усложнения процесса. Парадигма развития современных вычислительных сетей такова, что с ростом производительности происходит существенное снижение удельной стоимости вычислительных ресурсов, а это значит, что модели противодействия инсайдерам, формализованные по признакам, открытые для машинного обучения и обработки с использованием обработки больших данных, должны динамически развиваться, следуя усложнению вычислительных процессов в сети. В этой связи перспективным, по сути прорывным направлением в области SIEM-систем, будет развитие взаимодействия человек-машина таким образом, что роль администратора безопасности в большей степени будет связана с координацией и коррекцией процесса машинного обучения использующей обработку больших данных системы. Для решения задачи обнаружения и противодействия инсайдерам требуется выработка динамического таксонометрического классификатора инсайдеров и инсайдерских атак [11, 110], наиболее эффективного алгоритма машинного обучения и анализа больших данных по признакам не только состоявшихся, но также и планируемых инсайдерами инцидентов безопасности. Такой подход по сути является проактивной технологией обеспечения безопасности, позволяющей с помощью превентивных мер и оперативного реагирования эффективно решать задачу противодействия инсайдерам, и снижать ущерб от инсайдерских атак.

Обработка больших данных и алгоритмы машинного обучения могут также использоваться для создания внутри периметра ловушек различного рода, призванных привлечь внимание инсайдеров и облегчить процесс их обнаружения. Такие решения, проактивные и масштабируемые, могут быть построены на принципах, изложенных в [32].

В перспективе технологии больших данных могут применяться для противодействия инсайдерам не только путем анализа сетевых событий, но также путем анализа/распознавания эмотивных признаков лиц. Такие системы уже

разработаны и предлагаются банкам для обнаружения заемщиков, имеющих недобросовестные намерения. Широкое внедрение этих систем в России в настоящее время сдерживается их относительно высокой удельной стоимостью в сравнении с предлагаемыми на рынке аутсорсинговыми услугами детективных агентств.

Практическое применение обработки больших данных для мониторинга компьютерной безопасности будет рассмотрено далее в этой главе.

Решения, основанные на обработке больших данных.

В поисках лучшего общего классификатора, в [139] эмпирически оценивают 88 алгоритмов машинного обучения в 16 основных семействах. Они извлекают функции риска из набора данных CERT, который сочетает реальное поведение сети с отдельными описаниями угроз, а также обнаруживают прогностическую важность измерения настроения сотрудников. Среди основных семейств классификаторов, протестированных на CERT, лучший выбор предлагают алгоритмы случайных лесов, которые дают точность более 98%.

В [92] авторы подробно объясняют, как они создали новую реализацию алгоритма Random Forest на системной платформе высокопроизводительных вычислительных кластеров (HPCC) от LexisNexis. Чтобы справиться с этой сложной средой данных, был разработан инновационный подход, который отражает временную эволюцию взаимодействия пользователя с системой, чтобы создать неконтролируемую структуру обучения для обнаружения рискованного поведения инсайдеров.

В [132] авторы разработали инновационный подход, который отражает временную эволюцию взаимодействия между пользователями и системой, чтобы создать неконтролируемую систему обучения для обнаружения высокорискованного внутреннего поведения. Их метод основан на анализе двухстороннего графика взаимодействия пользователя и системы. Этот метод анализа графов потенциально способен обеспечить раннее обнаружение поведения внутренних угроз в результате взаимодействия между пользователями и системой, что позволит быстрее принимать меры по их устранению.

Исследование [123] предлагает использовать Dynamic Data Generator (DDG), который предоставляет возможность создания больших коллекций данных, состоящих из нескольких типов записей со сложными ограничениями и отношениями внутри и между записями, а также со случайностью (в пределах ограничений), отражающей демографическую статистику населения. Для каждого генерируемого набора тестовых данных DDG создает «модель вселенной», которую авторы статьи называют GAMUT (Great Automated Model Universe for Test). GAMUT содержит основные модели, относящиеся к генерируемому набору данных. Модель также поддерживает множество элементов управления конфигурациями, которые обеспечивают возможность настройки логики генерации данных в соответствии с поставленными требованиями. Это позволяет получать данные вместе с контекстом, который требуется SUT (System Under Test) для правильности работы и для достижения требуемого уровня реализма. Этот подход позволяет создавать согласованные разнородные хранилища данных, содержащие многочисленные выходные файлы. Эти файлы содержат данные в различных форматах, полученные от GAMUT. Благодаря этой конструкции DDG способен генерировать сложно различимые ошибки.

Схема работы GAMUT представлена на рисунке 1.14, на котором изображен обобщенный алгоритм обработки данных в системе. Данные (Requirements) собираются во множество (Data Set) и становятся входным элементом генератора данных, затем они поступают в обработчик с преобразованием и подвергаются валидации в модуле оценки (Scoring Module), выдавая, наконец, итоговый отчет (Test Report).

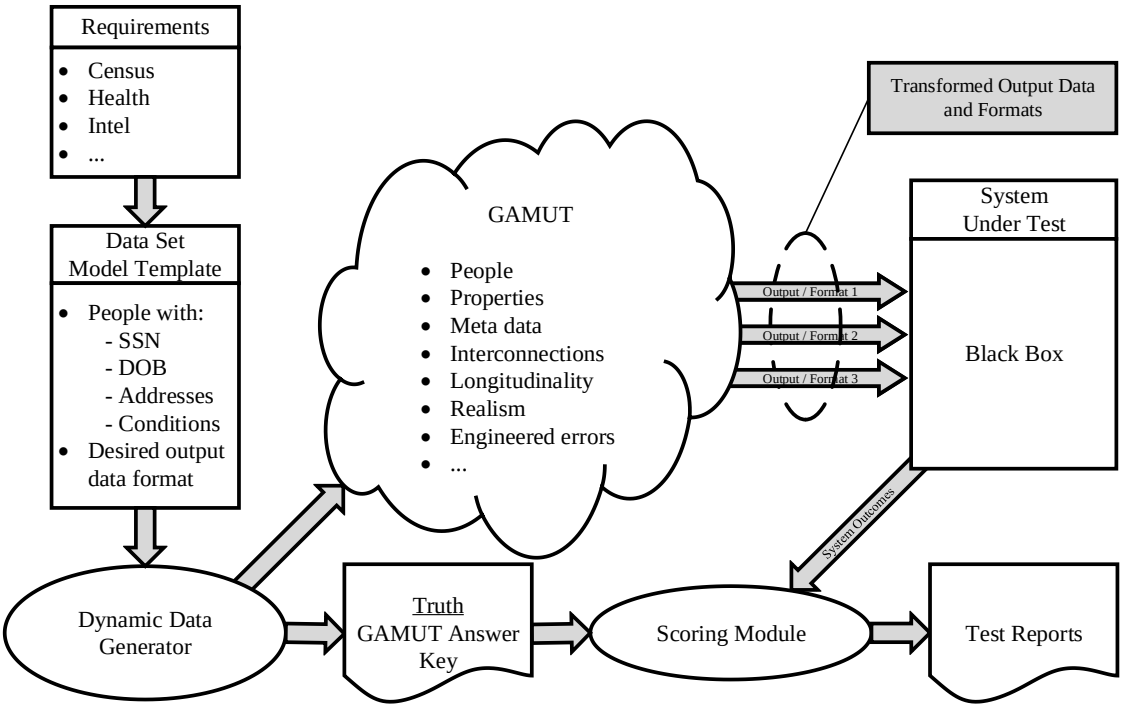

Рисунок 1.14 – Схема работы системы GAMUT

Для исходного набора данных авторы используют набор данных, опубликованный компьютерной группой реагирования на чрезвычайные ситуации университета Карнеги-Меллона.

В качестве основного алгоритма используется алгоритм, генерирующий большие наборы случайных данных, содержащие синтетические данные о пользователях, их параметры, сведения о соединениях и запросах, среди которых есть небольшое число аномалий. На основании этих данных строится граф отношений между пользователями, компьютерами, файлами, создается поведенческая модель и модель коммуникаций, выдвигаются сценарии угроз. Для анализа такого набора данных применяется визуализация, реализуемая через графы отношений.

В результате исследования релевантных работ в данной предметной области можно сделать вывод о том, что проблема обнаружения инсайдеров в сети является трудной задачей и пути ее решения продолжают совершенствоваться, а подходы достаточно разнообразны, чтобы прийти к единому способу обнаружения, поскольку каждая задача решается индивидуально [95, 102, 108, 117, 128]. В

большинстве работ авторы брали за основу математический аппарат технологий статистического анализа при обнаружении аномалий, использовали алгоритмы кластеризации, а метрики оценки эффективности основывались на оценке площадей под кривыми ROC-AUC (Receiver Operating Characteristic, Area Under the Curve).

Таким образом, в результате анализа релевантных работ можно использовать опыт новейших разработок в данной области относительно способов агрегации данных с последующим их анализом средствами статистического анализа и методов машинного обучения. Опираясь на правила обнаружения внутренних нарушителей, требуется создать комплексный подход для обнаружения инсайдеров.

1.4. Требования к системе обнаружения инсайдеров в компьютерных сетях

Сравнительный анализ исследовательских работ в области обнаружения инсайдерских атак в КС позволил определить требования к системе обнаружения инсайдеров, в основу реализации которых должен быть положен модельно-методический аппарат, разрабатываемый в настоящей работе. Данные требования можно разделить на две группы: функциональные и нефункциональные. Функциональные требования представляют собой перечень функций, которые должна выполнять система. Нефункциональные требования описывают целевые характеристики системы, такие как ограничения по времени, меры ошибок, полноты, точности и т.д.

Определим множество функциональных требований к системе обнаружения инсайдеров, реализующей разрабатываемые в данной работе подходы, следующим образом:

– учет опыта и наработанной базы используемых ранее и в настоящее время систем обнаружения инсайдеров в КС;

– учет специфики КС и действий инсайдеров в них;

– прогнозирование возможной инсайдерской деятельности еще до проведения самой атаки нарушителем;

– использование существующих научных подходов для реализации методов обнаружения инсайдеров (алгоритмы, основанные на экспертных правилах, методы машинного обучения и обработки больших данных и пр.);

– возможность последующего анализа всех внутренних данных, используемых системой в своей работе (например, для корректировки работы алгоритмов);

– возможность настройки работы алгоритмов системы экспертом по информационной безопасности в соответствии с обслуживаемой КС;

– генерация отчетов о работе системы и полученных результатах в виде, адаптированном для эксперта по информационной безопасности;

– учет специфики обслуживаемой КС, включая типовые для нее атаки инсайдеров.

Создаваемая система обнаружения инсайдеров должна учитывать успешное применение существующих подходов анализа КС [4, 5, 14, 19, 24, 42, 43, 47, 51, 75, 109, 124, 166, 184].

Множество нефункциональных требований к системе обнаружения инсайдеров можно определить, как три классические компоненты эффективности: своевременность (T), обоснованность (O) и ресурсопотребление (R) [37].

Под *своевременностью* понимается способность системы обеспечивать решение задачи – обнаружение инсайдеров – в установленный промежуток времени. Требования к своевременности могут быть заданы в формальном виде:

$$T \leq \min_{s \in S} T_s,$$

где T – время обнаружения инсайдеров разрабатываемой системой, T_S – время обнаружения инсайдеров системой s из множества всех альтернативных систем S. Для того чтобы разрабатываемая система могла использоваться в режиме, близком к реальному времени, она должна обнаруживать инсайдеров за время, не превышающее заданной границы. Данное требование к своевременности может быть задано в следующем виде:

$$P^T\left(T \le T_{def}\right) \ge P_0^T,$$

где P^T – вероятность завершения процесса работы системы по обнаружению инсайдеров за заданное время, T_{def} – допустимое время работы системы (равное $\min_{s \in S} T_s$), P_0^T – допустимое значение вероятности.

Обоснованность означает меру выполнения задачи системы, а именно – долю обнаруженных инсайдеров по сравнению с их реальным наличием в сети. Формальное соответствие системы данному критерию может быть определено посредством мер качества (которые будут описаны далее) и задано в формальном виде, как:

а) для полноты, точности, аккуратности, *F*-меры: $\begin{cases} O^i \in O \\ O^i \ge \max\limits_{s \in S} O_s^i, \end{cases}$

б) для ошибок: $\begin{cases} O^i \in O \\ O^i \le \min\limits_{s \in S} O_s^i, \end{cases}$

где O – множество мер качества, O^i – i-я мера качества, O_s^i – i-я мера качества системы s из множества всех альтернативных систем S.

Повышение обоснованности работы системы будет означать общее повышение защищенности КС, а, следовательно, и достижение цели исследования. Одной из наиболее общих мер, характеризующих качество работы, является F-мера и, следовательно, данное требование может быть записано следующим образом:

$$F\text{-}мера \rightarrow max.$$

Ресурсопотребление характеризует программные и аппаратные средства, необходимые системе обнаружения инсайдеров для решения своей задачи, а также их характеристики. Определим требование в формальном виде как совокупность

следующих показателей – количество хостов (h), средний сетевой трафик (n), объем занимаемого пространства на SSD/HDD (v), средняя нагрузка на CPU (c), средняя загрузка памяти (m):

$$\begin{cases} R^i \in R \\ R^i \leq \min\limits_{s \in S} R_s^i, \end{cases}$$

где R – множество показателей ресурсоэкономности, R^i – i-й показатель ресурсоэкономности (h, n, v, c, m), R_s^i – i-й показатель ресурсоэкономности системы s из множества всех альтернативных систем S.

Необходимо отметить, что непосредственное определение инсайдеров (как субъектов или личностей) возможно по сетевому трафику их активности в КС, а именно – по ID сетевой сессии. Для этого может быть применен соответствующий алгоритм преобразования таких ID сессий в непосредственный идентификатор инсайдеров. Поэтому сузим задачу исследования до обнаружения идентификаторов пользователей (user id), а также сетевых пакетов, объединенных в сессии и относящихся к инсайдерской деятельности.

Для определения точных характеристик инсайдеров (имена учетных записей, фамилии, должности и т.п.) могут применяться дополнительные действия; например, сопоставление IP-адресов злонамеренных сетевых пакетов с отчетами сервера журналирования о работающих за данным компьютером пользователях, анализом камер наблюдения, отметками о прохождении на территорию сотрудников и пр. Все это выходит за рамки предметной области диссертационного исследования.

Общее требование к удовлетворительности решения задачи разрабатываемой системой обнаружения инсайдеров может быть выражено при помощи следующих, достаточно известных и часто применяемых мер качества: *TP* (True Positive) – количество пользовательских сессий, определенных как инсайдерские, и являющиеся таковыми; *FP* (False Positive) – количество пользовательских сессий, определенных как инсайдерские, но не являющиеся таковыми; *TN* (True Negative) – количество пользовательских сессий, не определенных как инсайдерские и

являющиеся таковыми (то есть, которые не инсайдерские); *FN* (False Positive) – количество пользовательских сессий, не определенных как инсайдерские, но не являющиеся таковыми (то есть, которые инсайдерские [39]). Классическим синонимом FP служат ошибки I-го рода, а FN – ошибки II-го рода.

Качество обнаружения инсайдеров системой может быть оценено с помощью других, более понятных человеку мер: полноты, точности, аккуратности, ошибки, F-меры [2, 17, 3].

Полнота (r) характеризует способность системы выявлять инсайдеров, не учитывая при этом количество неверных срабатываний. Мера полноты может быть вычислена, как доля верно определенных инсайдерских сессий среди всех инсайдерских сессий:

$$r = \frac{TP}{TP+FN}.$$

Точность (p) характеризует способность системы выявлять только инсайдеров, не «захватывая» при этом легитимный трафик. Мера точности может быть вычислена, как доля верно определенных инсайдерских сессий среди всех определенных инсайдерских сессий:

$$p = \frac{TP}{TP+FP}.$$

Аккуратность (a) характеризует способность системы делать верные решения относительно определения инсайдеров. Мера аккуратности может быть вычислена, как доля верно определенных инсайдерских и не инсайдерских сессий среди всех пользовательских сессий:

$$a = \frac{TP+TN}{TP+FP+TN+FN}.$$

Ошибка (e) характеризует способность системы делать неверные решения относительно определения инсайдеров. Мера ошибки может быть вычислена, как доля неверно определенных инсайдерских и не инсайдерских сессий среди всех пользовательских сессий:

$$e = \frac{FP+FN}{TP+FP+TN+FN}.$$

F-мера (*f*), как правило, применяется для совместной оценки системы с позиции полноты и точности. F-мера может быть вычислена, как отношение удвоенного произведения полноты и точности системы к их сумме:

$$f = \frac{2 \times p \times r}{p+r}.$$

С помощью указанных мер разрабатываемая система обнаружения инсайдеров в КС может быть сравнена как с ближайшими аналогами, так и с ее собственными модификациями.

Основываясь на результатах опроса экспертов, серии проведенных экспериментов, исследовательских работах, а также на характеристиках типового серверного оборудования и автоматизированных рабочих мест, были установлены следующие требования к своевременности [45, 51, 88, 173], обоснованности [69, 74, 90, 110, 112, 141, 189] и ресурсопотреблению [49, 73, 75, 161, 183] относительно разрабатываемой системы обнаружения инсайдеров.

1) *Требования к своевременности*:

$$\begin{cases} T \leq \min\limits_{s \in S} T_s = T_{def} = 60 \text{ сек.} \\ P^T\left(T \leq T_{def}\right) \geq P_0^T = 0.98 \end{cases}$$

2) *Требование к обоснованности (а также ограничения к мерам качества)*:

- Требование: *F-мера* → *max*.
- Ограничение к полноте: $r \geq 0.90$.
- Ограничение к точности: $p \geq 0.92$.
- Ограничение к аккуратности: $a \geq 0.92$.
- Ограничение к *F*-мере: $f \geq 0.92$.
- Ограничение к ошибке: $e \leq 0.06$.

3) *Требования к ресурсопотреблению:* $R^i < R^i_{max}$, где максимальные показатели равны следующим значениям: количество хостов (R^h_{max}) = 6, средний сетевой трафик (R^n_{max}) = 100 Мб/сек., объем занимаемого пространства SSD/HDD (R^v_{max}) = 1 Тб, средняя нагрузка на CPU (R^c_{max})= 50%, средняя загрузка памяти (R^S_{max}) = 50%.

Таким образом, целевой функцией разрабатываемой системы является максимизация параметра обоснованности с учетом требований к своевременности и ресурсопотреблению; при этом, основным параметром обоснованности выбрана F-мера.

1.5. Постановка задачи исследования

Сформулирована задача исследования. Она заключается в разработке: (1) модели представления больших данных об инсайдерских атаках в формате NoSQL; (2) модели и алгоритмов комбинированного применения экспертных правил (RB-алгоритм, от англ. Rule-Based – на базе правил) и методов машинного обучения (ML-алгоритм, от англ. Machine Learning – машинное обучение) в интересах обнаружения инсайдерских атак; (3) методики обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных; (4) архитектуры и программной реализации системы обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных.

Целью исследования является повышение защищенности КС от внутренних атак. В диссертации показатель защищенности определяется через показатель обоснованности (F-мера) с учетом ограничений других показателей

обоснованности (полноты, точности, аккуратности, ошибки), а также с учетом требований к своевременности и ресурсопотреблению.

Следуя установленным требованиям к системе обнаружения инсайдеров в КС определим общий путь диссертационного исследования.

Методика обнаружения инсайдеров в КС, очевидно, должна использовать новые решения в области защиты ИБ (например, 145, 146, 187 и др.); они, во-первых, должны обладать существенной степенью высоко-технологичности, а, во-вторых, быть пока еще трудно нейтрализуемыми злоумышленниками (то есть инсайдерами). С другой стороны, накопленный опыт классических решений по противодействию инсайдерской деятельности также должен учитываться.

Таким образом, результатом исследований должно стать объединение различных успешных подходов к обнаружению инсайдеров в КС; очевидно, такое объединение должно учесть положительные стороны подходов, максимально избавившись от их отрицательных сторон. Также должны быть учтены возможные дополнительные эффекты, возникающие при объединении двух подсистем, – гипотетически, как повышающие итоговый результат, так и понижающие его; последнее, очевидно, должно быть минимизировано.

Для формирования пула возможных подходов, из которых может быть составлена результирующая методика, необходимо проанализировать как существующие на данный момент и активно используемые подходы, так и являющиеся перспективными, но возможно, применяемыми для близких задач. В результате можно будет выделить набор моделей, методик и алгоритмов, которые могут быть использованы в диссертационном исследовании. При этом, обобщающим звеном подходов можно считать их предрасположенность для работы с атрибутами поведения пользователя в сети, определяемыми посредством анализа сетевого трафика (как на уровне сетевых пакетов, так и исходя из логов работы сетевых сервисов и пр.).

Исходя из огромного и постоянно растущего количества генерируемых в единицу времени атрибутов, которыми может быть описана работа пользователя в КС, целесообразным может оказаться использование обработки больших данных.

Тем самым смогут быть решены как текущие, так и, возможно, будущие задачи сбора и хранения данных. В интересах хранения данных, очевидно, потребуется разработка собственной базы данных, адаптированной для хранения информации о поведении пользователей.

Используя модель представления больших данных в формате NoSQL, можно разработать соответствующие алгоритмы обнаружения аномалий в данных, сигнализирующих о действиях злоумышленников – инсайдеров КС. В первоначальном варианте достаточно разработать по одному алгоритму, условно противопоставимому друг другу с позиций используемого ими подхода. Так, в качестве первого алгоритма подходящим может оказаться алгоритм, построенный на основе экспертных правил. Под экспертными правилами понимаются правила, жестко заданные экспертом на основании логики, законов области применения и практического опыта эксперта. В качестве второго алгоритма потенциально востребованным является алгоритм, использующий в своей работе результаты машинного обучения, – то есть знающий типовые сценарии поведения инсайдеров и умеющий выявлять незначительные отклонения от этих сценариев.

Результатом объединения этих алгоритмов будет комплексная методика обнаружения инсайдеров в КС, в создании которой и заключается диссертационная работа; в меру своей специфики, оба алгоритма могут работать параллельно.

Для практического использования созданной методики необходима разработка архитектуры программного комплекса системы обнаружения инсайдеров в КС, включающая в себя ее основные модули, информационные и управляющие потоки, а также алгоритмы работы. Реализация такой архитектуры в виде программного комплекса позволит непосредственно проверить работу методики на практике.

Необходимым условием достижения цели исследования должно стать проведение сравнения разработанной методики с аналогами, результаты которого обоснованно покажут степень повышения защищенности КС.

Следует иметь в виду, что поскольку каждый из алгоритмов на выходе будет выдавать свое множество обнаруженных инсайдеров (заданных связанным с

каждым из них ID сетевой сессии), – простое объединение этих множеств может быть некорректным. Так, например, может оказаться, что один из алгоритмов в принципе будет работать лучше другого и объединение их выходов лишь ухудшит конечный результат. Следовательно, помимо сравнения разработанной методики с аналогами, потребуется сравнение различных вариаций методики, связанных с формулами вычисления конечного результата (то есть, множества инсайдеров) – по результату работы алгоритмов на основе экспертных правил I_{RL} или методов машинного обучения I_{ML}.

Конечный результат I_{Res} может быть подсчитан с помощью одной из четырех следующих формул: как объединение множеств результатов алгоритмов – (1) $I_{Res} = I_{RB} \vee I_{ML}$, как их пересечение – (2) $I_{Res} = I_{RB} \wedge I_{ML}$ или как результат одного из алгоритмов (очевидно, наилучшего) – (3) $I_{Res} = I_{RB}$ или (4) $I = I_{ML}$. Конечный выбор наилучшей вариации методики может быть сделан на основании введенных ранее мер: полноты, точности, аккуратности, ошибки и F-меры. Учтем также тот факт, что алгоритм на базе методов машинного обучения может быть представлен на нижнем уровне в виде базовых классификаторов: DT, NB, k-NN, SVM; на верхнем уровне – в виде композиций базовых классификаторов: голосование большинством (PV), взвешенное голосование (WV) и мягкое голосование (SV), а также Adaboost [172]. Выбор способа классификации будет влиять на результаты работы методики.

Исходя из предполагаемого хода диссертационного исследования, опишем задачи, решение которых будет необходимо:

1) разработка модели представления больших данных об инсайдерских атаках в формате NoSQL (включая модель инсайдера);
2) разработка алгоритма обнаружения инсайдеров в КС, основанного на экспертных правилах;
3) разработка модели и алгоритмов комбинированного применения экспертных правил и методов машинного обучения в интересах обнаружения инсайдерских атак;

4) разработка методики обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных;
5) построение архитектуры и реализация программного комплекса системы обнаружения инсайдеров в КС на базе предложенной методики, настройка алгоритма на основе методов машинного обучения с помощью набора данных, характеризующих действия инсайдеров по заданному множеству сценариев атак, и экспериментальная оценка разработанной методики системы обнаружения инсайдеров в КС.

На содержательном уровне научную задачу диссертационного исследования можно сформулировать следующим образом: разработать модельно-методический аппарат (модель представления больших данных об инсайдерских атаках в формате NoSQL, модель и комплекс алгоритмов обнаружения инсайдеров, методику обнаружения инсайдеров, архитектуру системы обнаружения инсайдеров), реализующий повышение защищенности КС. Разработка и экспериментальная проверка функционирования программной реализации соответствующей системы обнаружения позволит определить результативность модельно-методического аппарата.

Входными для задачи исследования являются следующие данные, описывающие поведения пользователей $Users$:

$$\{Netflow, Applications, Data, Scanners, Servers, Devices\},$$

где *Netflow* – статистические данные о взаимодействии хостов сети, *Applications* – пользовательские приложения, *Data* – сырые данные (необработанная последовательность байт), *Scanners* – сканеры, реализующие сбор информации о сети, *Servers* – серверы, предоставляющие сервисы пользователям в компьютерной сети (DHCP, RADIUS, DNS), *Devices* – пользовательские устройства.

Требуется найти внутренних нарушителей (инсайдеров) $\{Insiders\}$. Определение инсайдеров осуществляется на основе атрибутов поведения пользователей. Поведение инсайдера может быть формализовано на основе

введения порогов, задающих разные характеристики действий, выполняемых инсайдером, например, объем загруженных файлов. Поведение инсайдера задается с помощью модели инсайдера, что может быть описано в следующем виде:

$$I =< R, L, Q, G >,$$

где R – критерии атрибутов инсайдера, L – уровни доступа, Q – квалификация инсайдера, G – цель инсайдера.

Таким образом, научная задача может быть описана следующим образом: для имеющегося набора входных данных о поведении пользователя найти следующий кортеж:

$$< Mod, Alg, Met, Arch >,$$

где Mod – модель представления больших данных об инсайдерских атаках в формате NoSQL, включающая модель инсайдера; Alg – модель и алгоритмы комбинированного применения экспертных правил и методов машинного обучения в интересах обнаружения инсайдерских атак; Met – методика обнаружения инсайдеров в КС, $Arch$ – архитектура и программная реализация системы обнаружения инсайдеров в КС.

При этом необходимо добиться максимизации показателя F-меры комплекса алгоритмов при ограничениях следующих мер: (1) полноты, (2) точности, (3) аккуратности, (4) ошибки; с учетом требований к своевременности и ресурсопотреблению.

1.6. Выводы по главе 1

1. Проведен анализ задачи обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных. Рассмотрены различные подходы к созданию систем обнаружения инсайдеров. На базе проведенного анализа сделан общий вывод о

необходимости и важности разработки собственной системы обнаружения инсайдеров в КС с учетом приведенных требований.

2. Проведены исследования решений по мониторингу компьютерной безопасности на основе обработки больших данных, которые должны удовлетворять следующим требованиям:

– адаптивная и высокомасштабируемая обработка событий, обеспечивающая управление большими объемами данных о безопасности в реальном или близком к реальному времени;

– межуровневая корреляция событий безопасности, поступающих из неоднородных источников;

– высокая доступность и отказоустойчивость сбора данных о событиях безопасности.

3. Выполнена постановка задачи исследования, которая включает в себя разработку модельно-методического аппарата для обнаружения нарушителей информационной безопасности в КС внутреннего периметра с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных.

4. Сформулирована цель исследования – повышение защищенности КС за счет усовершенствования методик, моделей и алгоритмов обнаружения инсайдеров КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных.

Глава 2. Модели и алгоритмы обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных

2.1 Модель представления больших данных об инсайдерских атаках в формате NoSQL

Современные инсайдерские атаки являются комплексными и используют множество способов реализации и векторов атак для получения несанкционированного доступа и компрометации информационных объектов во внутренней сети. Инсайдером может быть любой пользователь сети. Следовательно, в системах защиты от атак необходимо выполнять процедуры анализа и контроля действий пользователей, называемые профилированием поведения пользователей.

В существующих исследованиях и разработках эти процедуры получили название аналитики поведения пользователей – User Behavior Analytics (UBA) и User and Entity Behavior Analytics (UEBA) [23, 35]. Формально системы UBA и UEBA принадлежат к одному и тому же классу систем, но между ними есть одна фундаментальная разница. UBA-системы используют информацию, содержащую только данные о пользовательской активности, следовательно, фокусируются на пользователях и их ролях. UEBA-системы вместе с данными, которые применяются в UBA-системах, учитывают информацию о системном окружении (сетевой трафик, системы хранения данных, рабочие станции и программное обеспечение). Это дает возможность UEBA-системам профилировать не только пользователей, но и состояние программного и аппаратного обеспечения в целом. Именно это позволяет UEBA-системам распознавать более обширный класс угроз [62, 97, 164, 167].

Для реализации UBA и UEBA необходима система управления базами данных (СУБД), способная легко масштабироваться и обладающая высокой скоростью обработки запросов. Для этой цели в настоящее время используются СУБД NoSQL (Not only SQL) [16, 24, 28, 40, 119, 140, 152, 178]. Решения на основе NoSQL в целом предоставляют масштабируемый и гибкий способ решения задач, которые ранее управлялись реляционными базами данных. Примером СУБД NoSQL является OrientDB [80, 144], которая объединяет в себе возможности документо-ориентированной и графо-ориентированной баз данных (БД). Это означает, что она обладает полными графическими возможностями в сочетании с функциями, обычно присутствующими только в базах данных документов.

В диссертации рассматривается построение модели представления больших данных об инсайдерских атаках в формате NoSQL для обнаружения инсайдеров в КС. Задача состоит в том, чтобы собрать максимальное количество данных из системы, сформировать с их помощью профили поведения пользователей и определить по совокупности собранной информации, поведение каких пользователей отличается от нормального поведения. Далее на основе этой информации можно выявить возможных инсайдеров и способы реализации ими несанкционированных действий. Основная цель состоит в том, чтобы показать возможность создания и использования агрегированной модели представления больших данных об инсайдерских атаках в формате NoSQL, которая учитывает поведение пользователей для последующего использования этой модели для обнаружения нарушителей информационной безопасности.

Для построения модели представления больших данных об инсайдерских атаках в целях обнаружения инсайдеров в КС рассмотрим источники собираемых данных. В качестве источников данных для аналитики поведения пользователей возьмем все клиентские устройства, подключенные как к беспроводным, так и к проводным компонентам КС. В предложенной модели предполагается осуществлять контроль над всем оборудованием, находящимся в локальной сети, и над теми устройствами, наличие которых не предполагалось владельцем сети,

например, мобильных телефонов, ноутбуков и прочих устройств, имеющих возможность беспроводного подключения.

На рисунке 2.1 приводится пример источников для сбора сведений, которыми могут выступать данные, передаваемые по сети, различные приложения, а также используемые устройства.

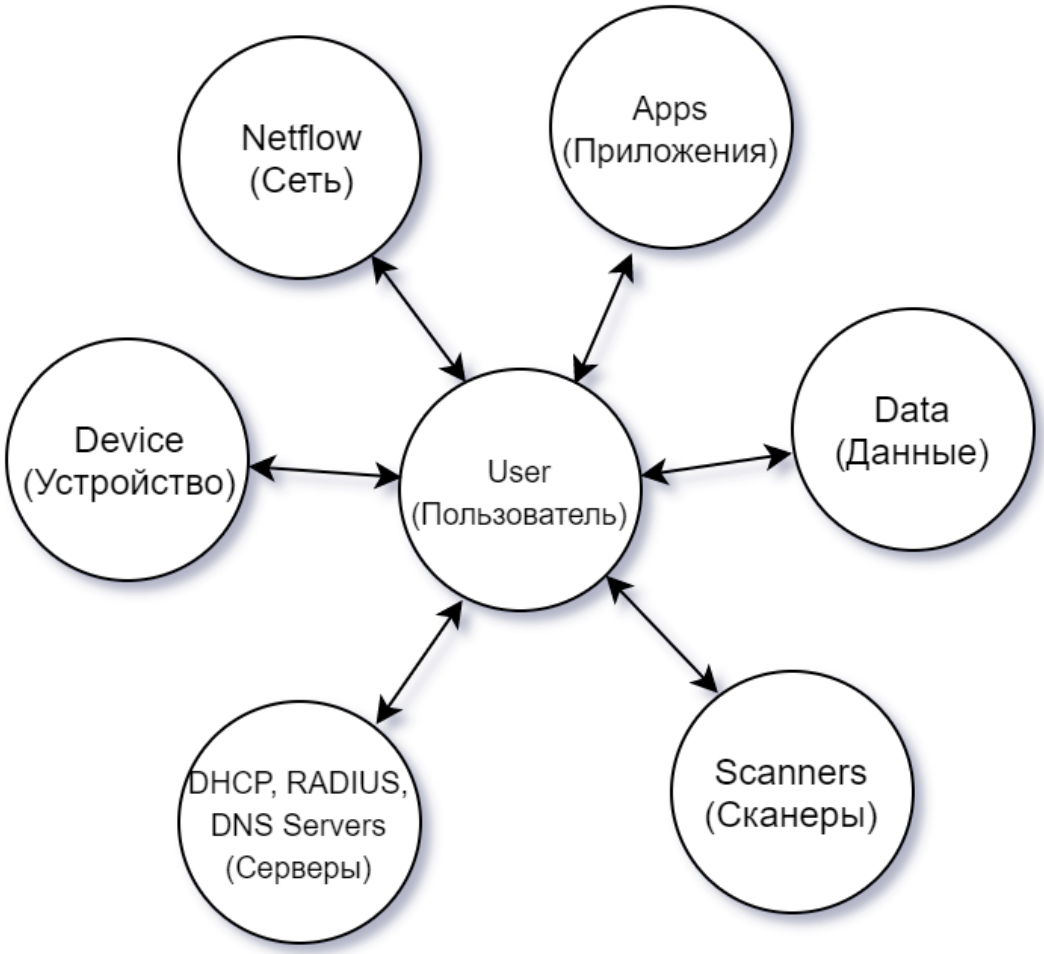

Рисунок 2.1 – Источники для сбора данных, описывающих поведение пользователя в КС

Формальный вид модели представления больших данных об инсайдерских атаках имеет следующий вид:

$$M = \langle A, I \rangle ,$$

где A – элементы, представляющие собой атрибуты поведения пользователя, I – модель инсайдера и критерии (определены в п. 2.2 диссертации), которые позволяют определить текущего пользователя к категории инсайдеров.

Представим выделенные атрибуты поведения пользователей и их взаимосвязи формально:

$$A = \langle DataSources, Users, Data, Parser \rangle.$$

Перечислим элементы, входящие в этот кортеж:

– $DataSources = \langle Netflow, Application, Scanner, Server, Device \rangle$ – источники данных, каждый элемент которых представляет собой соответственно сетевой поток, приложение, файл операционной системы, сканер, сервер, устройство;

– $Users = \bigcup_{i=1}^{P} User_i$ – пользователи, $User_i = \langle UserID_i, Attr_i, Sessions_i \rangle$ – триплет, представляющий собой соответственно идентификатор пользователя, атрибуты пользователя и соответствующие ему сессии;

– $Data = \{0,1\}^+ = \{0,1,00,01,10,11,000, \dots\}$ – данные, представляющие собой всевозможные битовые цепочки, хранящиеся на источниках данных;

– $Parser: Data \times DataSources \times Time \rightarrow Sessions$ – отображение, формирующее сессию из сырых данных в зависимости от типа источника этих данных и времени.

Модель инсайдера может быть описана в следующем виде:

$$I =< R, L, Q, G >,$$

где R – критерии атрибутов, состоящие из набора признаков, по которым принимается решение об отнесении пользователя к множеству инсайдеров (например, регламентированный график работы, допустимая нагрузка на сеть, оценка работы с информационными ресурсами); L – уровни доступа, определяющие права пользователей в КС, нарушение которых будет означать потенциальную инсайдерскую деятельность (например, оператор, инженер или администратор); Q – квалификация инсайдера, определяющая необходимый уровень подготовки для проведения атаки (например, хакер – обладающий знаниями об уязвимостях информационных систем, средствах их эксплуатации, методами сокрытия следов нарушений; вандал – владеющий в основном методами и механизмами слома элементов КС, в том числе физического; неблагонадежный

пользователь – не умеющий наносить сознательный вред КС и циркулирующей в ней информации, но подверженный получению, искажению и распространению информации условно-законными способами без злого умысла); *G* – цель инсайдера, определяющая основной вектор инсайдерской деятельности, связанный с соответствующими угрозами нарушения конфиденциальности, целостности и доступности информации (например, сбор не предназначенной для пользователя информации; вынос конфиденциальной информации за периметр организации и/или ее передачу третьим лицам; модификация информации, включая ее полное уничтожение и т.п.).

Рассмотрим элементы, представляющие собой атрибуты поведения пользователей. Источники данных содержат информацию о пользователях в сыром виде. Для преобразования этих данных в сессии применяется отображение *Parser*. Например, в случае сетевого потока данное отображение позволяет выделить TCP-соединение или HTTP-сессию, а в случае приложения / файла сформировать характеристики сессии ОС, в рамках которой это приложение было установлено / файл был создан.

Предположим, что на одном из источников данных *datasrc* были сгенерированы данные *data* в определенный момент времени *time*. Тогда идентификатор пользователя *uid*, в рамках одной из сессий которого были сгенерированы эти данные, определяется следующим образом:

$$uid \in \{userid | \langle userid, attr, sessions \rangle \in Users \wedge$$

$$Parser(data, datasrc, time) \in sessions\}$$

В исследуемых данных выделим основные атрибуты поведения пользователей, необходимые для исследования их активности и детектирования аномалий с целью возможного обнаружения инсайдеров.

Приведем пример атрибутов поведения пользователей:

– **UserFields** – поля, принадлежащие пользователю (не зависят от конкретной сессии);

– **ID** – уникальный номер пользователя;

– **User-Agent** – атрибуты конечного пользователя, которые позволяют определить информацию о его типе, операционной системе, вендоре и т.д.;

– **Login** – логин пользователя;

– **Pass** – пароль пользователя;

– **T** – пороговое значение, указывающее степень доверия к пользователю (является предустановленным числом, которое может быть скорректировано оператором системы обнаружения инсайдеров);

– **Rights** – права доступа на чтение, изменение, редактирование файлов;

– **TotalAuth** – общее число попыток входа (необходимо для оценки количества попыток входа в систему с различных устройств);

– **Active Directory-пробы** – пробы, основанные на полученных данных из AD;

– **AD-host** – признак того, является ли хост членом домена;

– **AD-domain** – домен подключения;

– **AD-operation-system** – текущая операционная система;

– **AD-version OS** – текущая версия операционной системы;

– **AD-service pack** – текущий пакет обновлений операционной системы;

– **Radius-пробы** – список проб, основанных на обращении конечного устройства к Radius серверу;

– **Calling-Station-ID** – MAC-адрес конечной точки;

– **NAS-IP-Address** – IP-адрес сетевого устройства доступа, которое является аутентификатором в сети;

– **NAS-Port** – номер физического порта аутентификатора;

– **Framed-IP-Address** – IP-адрес конечной точки;

– **Acct-Session-ID** – уникальный идентификатор сессии учета;

– **Acct-Session-Time** – время, в течение которого конечная точка получает сервис;

– **Acct-Terminate-Cause** – если сессия или соединение разорвано, данное поле будет содержать информацию о причине;

– **On-For-Login-Auth** – используется для пересылки меток QoS в пакетах аутентификации;

– **DHCP-проба** – список проб, основанных на DHCP-обращении конечного устройства;

– **Dhcp-class-identifier** – сообщает платформу устройства или информацию об ОС;

– **Dhcp-client-identifer** – отображает MAC-адрес конечного устройства;

– **Dhcp-user-class-id** – параметр некоторых ОС (MAC/Windows), который является уникальным корпоративным идентификатором клиента;

– **Dhcp-requsted-address** – IP-адрес устройства;

– **Dhcp-server-identifier** – идентификатор сервера;

– **Dhcp-parameters-request-list** – уникальный идентификатор типа устройства;

– **Dhcp-message-type** – тип DHCP-сообщения;

– **DNS-пробы** – проба, основанная на получении информации от DNS сервера;

– **DNS-FQDN** – полное доменное имя;

– **NMAP-пробы** – сканирование открытых портов на конечном устройстве;

– **Session** – список сессий с информацией о них (представляет из себя список из кортежей типа **SessionData**);

– **SessionData** – поля, принадлежащие конкретной уникальной сессии и содержащие информацию о ней;

– **SessionID** – уникальный номер сессии;

– **Changes** – список файлов, к которым был осуществлен доступ со времени входа в систему;

– **Auth** – число попыток входа в систему перед успешной аутентификацией в данной сессии;

– **LogPass** – связки логин-пароль, используемые при попытке входа в систему;

– **Sites** – использование нестандартных сетевых ресурсов (указывает, посещал ли пользователь во время сеанса подозрительные веб ресурсы: пользователи, не обладающие высокими привилегиями и использующие необычные ресурсы, могут оказаться нарушителями информационной безопасности);

– **Periph** – используемые периферийные устройства (поле содержит список всех используемых в ходе сессии устройств);

– **Time** – время, прошедшее от начала сессии, или время выхода из системы (подозрительными могут считаться сессии, совершенные в необычное для пользователя время);

– **Progs** – наличие на компьютере приложений, осуществляющих подозрительную активность (это могут быть утилиты, совершающие в фоновом режиме повышенную сетевую активность);

– **NoAV** – указывает на отсутствие антивирусных программ на устройстве, либо на ситуацию, когда антивирусные базы сильно устарели (является важным критерием при детектировании инсайдеров);

– **LogType** – способ входа в систему (стандартный или нестандартный; может быть осуществлен с рабочего места или через виртуальную частную сеть);

– **Geo** – географическое положение (подозрительной может считаться резкая смена географического положения при работе в системе);

– **DeviceID** – уникальный номер устройства;

– **AppID** – уникальный номер приложения;

– **DeviceFields** – поля, принадлежащие конкретному устройству и не зависящие от пользователя;

– **Name** – имя устройства;

– **OS** – название операционной системы, версия, текущий пакет обновлений;

– **Vendor** – наименование фирмы-изготовителя;

– **AppFields** – поля, принадлежащие конкретному приложению на устройстве;

– **AppName –** название ПО;

– **Version** – версия ПО;

– **Developer** – разработчик ПО;

– **Ports** – порты, используемые приложением;

– **Netflow** – список потоков, который содержит информацию о сетевой активности пользователя в конкретной сессии;

– **Source IP address** – IP-адрес источника;

– **Destination IP address** – IP-адрес назначения;

– **Next-Hop IP address** – IP-адрес следующего маршрутизатора, на который будет передан сетевой поток;

– **Input ifIndex** – SNMP индекс интерфейса, через который маршрутизатор получает сетевой поток;

– **Output ifIndex** – SNMP индекс интерфейса, через который маршрутизатор передает сетевой поток;

– **Packets** – общее количество полученных пакетов в рамках потока;

– **Bytes** – общее количество байт, полученных в рамках потока;

– **Start time of flow** – время начала потока;

– **End time of flow** – время окончания потока;

– **Source port** – порт источника;

– **Destination port** – порт назначения;

– **TCP Flags** – TCP флаги;

– **IP protocol** – номер IP протокола;

– **ToS** – тип сервиса;

– **Source AS** – номер автономной системы IP источника;

– **Destination AS** – номер автономной системы IP назначения;

– **Source Mask** – маска сети IP источника;

– **Destination Mask** – маска сети IP назначения;

– **Padding** – отступы для эффективного использования всей длины заголовка;

– **Source VLAN** – номер VLAN источника;

– **Destination VLAN** – номер VLAN назначения;

– **Source MAC** – MAC-адрес источника;

– **Destination MAC** – MAC-адрес назначения

– **Income Traffic [SIZE]** – массив значений объема полученного сетевого трафика за каждый час в течение последних 30 дней (таким образом, SIZE = 24 * 30 = 720);

– **Outcome Traffic [SIZE]** – массив значений объема отправленного сетевого трафика за каждый час в течение последних 30 дней (таким образом, SIZE = 24 * 30 = 720)

– **Traffic Time** – время окончания сбора информации о сетевом трафике (используется, как отправная точка) для полей **Income Traffic/Outcome Traffic**.

Эти и другие поля вместе со взаимосвязями между ними изображены на рисунке 2.2.

В списке полей БД содержатся следующие значения: Ключ и Документ (таблица 2.1).

Таблица 2.1. Пример модели представления больших данных в формате NoSQL

Ключ	Документ
1	{**"Session"**: "1001", **"ID"**: "1", **"User-Agent"**: "Chrome", **"Login"**: "Ivan", **"Pass"**:"Ivan 123", **"T"**:"3", **"Income Traffic"**: [5,5,5, … 6,5,5], **"Outcome Traffic"**: [1,1,1, … 2,1,1], **"Traffic time"**: "01-01-2020 18:00:00", **"Rights"**:"Read|Write", "TotalAuth":"", "Active Directory-пробы":"", "AD-host":"", "AD-domain":"", "AD-operation-system":"", "AD-version OS":"", "AD-service pack":"", "Radius-пробы":"", "Calling-Station-ID":"", "NAS-IP-Address":"", "NAS-Port":"", "Framed-IP-Address":"", "Acct-Session-ID":"", "Acct-Session-Time":"", "Acct-Terminate-Cause":"", "On-For-Login-Auth":"", "DHCP-проба":"", "Dhcp-class-identifier":"", "Dhcp-client-identifer":"", "Dhcp-user-class-id":"", "Dhcp-requsted-address":"", "Dhcp-server-identifier":"", "Dhcp-parameters-request-list":"", "Dhcp-message-type":"", "DNS-пробы":"", "DNS-FQDN":"", "NMAP-пробы":"", "SessionData":"", "SessionID":"", "Changes":"", "Auth":"", "LogPass":"", "Sites": [], "Periph":"", "Time":"", "Progs ":"", "NoAV":"", "LogType":"", "Geo":"", "DeviceID":"", "AppID":"", "DeviceFields":"", "Name":"", "OS":"", "Vendor":"", "AppFields":"", "AppName":"", "Version":"", "Developer":"", "Ports":"",

	"Netflow":"", "Source IP address":"", "Destination IP address":"", "Next-Hop IP address":"", "Input ifIndex":"", "Output ifIndex":"", "Packets":"", "Bytes":"", "Start time of flow":"", "End time of flow":"", "Source port":"", "Destination port":"", "TCP Flags":"", "IP protocol":"", "ToS":"", "Source AS":"", "Destination AS":"", "Source Mask":"", "Destination Mask":"", "Padding ":"", "Source VLAN":"", "Destination VLAN":"", "Source MAC":"", "Destination MAC - ":"", "UserFields":""}
2	{"Session":"1002”, "ID":"2", "User-Agent": "Firefox", "Login": "Petr", "Pass":"Pet_456", "T":"3", "Income Traffic": [0,0,0, … 0,0,10], "Outcome Traffic ": [10,0,0, … 0,0,0], ... }
…	…

Примечание. В приведенном списке полей БД для примера заполнены лишь основные из них.

Таблица 2.2. Пример модели и критериев представления инсайдера в формате NoSQL

Ключ	**Документ**

Критерий 1	{
	"Name": "Функционирование сети",
	"Условия": {
	"Максимальное количество обращений к хостам (шт./мин)": [1000, "Запрещено"],
	"Максимальное количество нетиповых запросов (шт./мин)": [1000, "Запрещено"],
	"Нестандартные завершения сессии": "Запрещено"
	}
	}

Критерий 2	{
	"Name": "Функционирование Active Directory",
	"Условия": {
	"Наличие узлов – не являющихся членами домена": "Запрещено",
	"Отсутствие доменного имени": "Запрещено"
	}
	}

Критерий 3	{ "Name": "Заражение ПК",

	"Условия": {
	"Необходимые версии ПО": [["Windows 10", "Windows server 2019"], "Запрещено"]
	"Опасные сайты": [["vk.com", "casino.ru", "cia.gov"], "Подозрительно"],
	"Предел нетипичной активности приложений (%)": ["70", "Подозрительно"],
	"Нестадартные подключения": ["Подозрительно"]
	}
	}

Критерий 4	{
	"Name": "Человеческий фактор",
	"Условия": {
	"Новые периферийные устройства": ["Подозрительно"],
	"Нестандартные устройства – смена параметров": [["ОС", "MAC", "IP"], "Подозрительно"]
	}
	}

Критерий 5	{
	"Name": "Аутентификация",
	"Критерии": {
	"Максимальное количество неудачных аутентификаций": [[3, "Подозрительно"], [10, "Запрещено"]]
	}
	}

Критерий 6	{
	"Name": "Сбор и утечка информации",
	"Условия": {
	"Максимальное превышение количества отправленной за пределы периметра организации информации (Гб/час)": [[1, "Подозрительно"], [10, "Запрещено"]],
	"Максимальное превышение количества собранной в рамках периметра организации информации (Гб/час)": [[1, "Подозрительно"], [10, "Запрещено"]]
	}
	}

Критерий 7	{ "Name": "Рабочий распорядок пользователей",

	"Критерии": {
	"Начало сессии (время)": [["пн|вт|ср|чт", ["09:00-18:00"], "Подозрительно"], ["пт", ["09:00-13:00", "14:00-18:00"], "Подозрительно"]],
	"Конец сессии (время)": [["пн|вт|ср|чт", ["09:00-18:00"], "Подозрительно"], ["пт", ["09:00-13:00", "14:00-18:00"], "Подозрительно"]]
	}
	}

Критерий 8	{
	"Name": "Ограничения",
	"Условия": {
	“Высокая важность”: 3,
	“Критичная важность”: 6,
	“Минимальный уровень доступа”: “Operator”,
	“Максимальный уровень доступа”: “Administrator”,
	“Квалификация инсайдера”: [“Хакер”, “Вандал”, “Неблагонадежный пользователь”],
	“Цель инсайдера”: [“Сбор информации”, “Вынос информации”, “Модификация информации”],
	}
	}

Для пояснения идеи работы моделей приведем примеры отображения в них инсайдера (U_I) и законного пользователя (U_L). Предположим, инсайдерская деятельность первого пользователя заключается в скачивании большого размера трафика с внутренних ресурсов организации и последующем посещении подозрительных сайтов (например, с целью отправки обработанных данных вне периметра организации). Соответственно, второй пользователь может делать то же самое, но в значительно меньших объемах – то есть вести законную деятельность (например, в рамках должностных обязанностей).

Для обнаружения такого рода атак в модели инсайдера (описанных в терминах NoSQL базы данных: Ключ – Документ) будет использованы следующие критерии:

Таблица 2.3. Пример критериев и условий модели представления инсайдера в формате NoSQL

Ключ	**Документ**

Критерий 3	{
	"Name": "Заражение ПК",
	"Условия": {
	"Опасные сайты": [["vk.com", "casino.ru", "cia.gov"], "Подозрительно"],
	}
	}

Критерий 6	{
	"Name": "Сбор и утечка информации",
	"Условия": {
	"Максимальное количество собранной в периметре организации информации (Гб/час)": [[1, "Подозрительно"], [10, "Запрещено"]]
	}
	}

Критерий 8	{
	"Name": "Ограничения",
	"Условия": {
	“Высокая важность”: 3,
	“Квалификация инсайдера”: “Неблагонадежный пользователь”],
	“Цель инсайдера”: [“Сбор информации”, “Вынос информации”],
	}
	}

Несоответствие поведения пользователей данным критериям может сигнализировать об инсайдерской деятельности.

Рассмотрим вид модели представления больших данных об инсайдерских атаках для поведения U_I (с Id сессии 1001) и U_L (с Id сессии 2001) в части ее полей, связанных с этими критериями:

Таблица 2.4. Пример полей модели представления больших данных в формате NoSQL для инсайдера и законного пользователя

Ключ	Документ
1	{"Session":"1001”, "ID":"1", "Income Traffic": [1,1,1,1,1,1,25], “Traffic time”: “01-01-2020 18:00:00”, “Sites”: [“cia.gov”, “yandex.ru”, “google.ru”]}
2	{"Session":"2001”, "ID":"2", "Income Traffic": [1,1,1,1,1,1,5], “Traffic time”: “01-01-2020 16:00:00”, “Sites”: [“gov.ru”, “yandex.ru”, “google.ru”]}

Как видно из таблицы 2.4, пользователь, ассоциированный с сессией 2001 (строка с ключом 2), соответствует пользователю U_L, поскольку не удовлетворяет критериям инсайдера: скачивает на 4 Гб больше по сравнению с дневной нормой и посещает разрешенные сайты. С другой стороны, пользователь, ассоциированный с сессией 1001 (строка с ключом 1) соответствует пользователю U_I, поскольку удовлетворяет критериям инсайдера: скачивает на 24 Гб больше по сравнению с дневной нормой и посещает запрещенные сайты (например, cia.gov – сайт ЦРУ США).

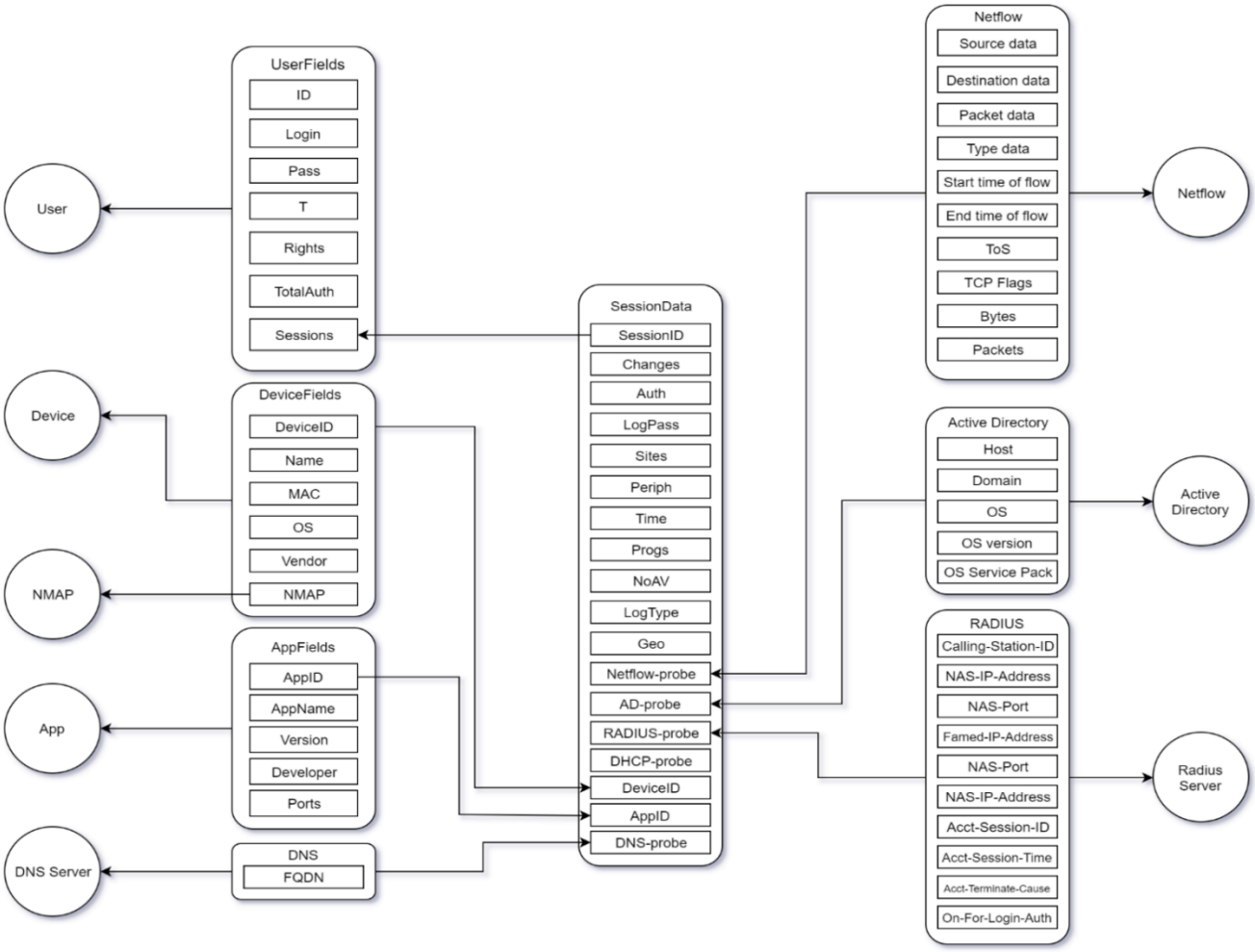

Рисунок 2.2 – Взаимосвязь между основными атрибутами поведения пользователей

Представим в формальном виде атрибуты, необходимые для исследования активности и детектирования аномалий с целью возможного обнаружения инсайдеров:

$$Devices = \bigcup_{i=1}^{n} Device_i, Device_i = \langle DeviceID, Name, OSType, Vendor \rangle$$

В представленном кортеже *Devices* имеются следующие поля: *DeviceID*, *Name*, *OSType* и *Vendor*. А сам кортеж состоит из объединения множеств, содержащих данные по заданным полям.

$$Apps = \bigcup_{j=1}^{m} App_j, App_j = \langle AppID, AppName, Version, Developer, Ports \rangle$$

В кортеже *Apps* имеются следующие поля: *AppID*, *AppName*, *Version*, *Developer*, *Ports*.

$$Sessions = \bigcup_{k=1}^{p} SessionData_k, SessionData_k = \left\langle \begin{matrix} SessionID, Changes, Auth, \\ LogPass, Sites, Periph, \\ Time, Progs, NoAV, \\ LogType, Geo, DeviceID, \\ AppID \end{matrix} \right\rangle$$

Кортеж *Sessions* состоит из следующих полей: *SessionID*, *Changes*, *Auth*, *LogPass*, *Sites*, *Periph*, *Time*, *Progs*, *NoAV*, *LogType*, *Geo*, *DeviceID*, *AppID*.

$$Attr = \bigcup_{l=1}^{F} Attr_l, Attr_l = \langle Login, Pass, T, Rights, TotalAuth \rangle$$

Кортеж *Attributes* состоит из следующих полей: *Login*, *Pass*, *T*, *Rights*, *TotalAuth*.

$$Netflowprobe = \cup_{l=1}^{F} Netflowprobe_l,$$

$$Netflowprobe_l = \left\langle \begin{matrix} SourceIPaddress, DestinationIPaddress, \\ Next - hopIPaddress, InputifIndex, OutputifIndex, \\ Packets, Bytes, StartTimeofFlow, EndTimeofFlow, \\ SourcePort, DestinationPort, TCPFlags, IPprotocol, ToS, \\ SourceAS, DestinationAS, SourceMask, DestinationMask, \\ Padding, SourceVLAN, DestinationVLAN, \\ SourcMAC, DestinationMAC \end{matrix} \right\rangle$$

Кортеж *Netflow probe* состоит из следующих элементов: *Source IP address*, *Destination IP address*, *Next – hop IP address*, *Input if Index*, *Output if Index*, *Packets*, *Bytes*, *Start Time of Flow*, *End Time of Flow*, *Source Port*, *Destination Port*, *TCP Flags*, *IP protocol*, *ToS*, *Source AS*, *Destination AS*, *Source Mask*, *Destination Mask*, *Padding*, *Source VLAN*, *Destination VLAN*, *Source MAC*, *Destination MAC*.

$$ADprobe = \bigcup_{l=1}^{F} ADprobe_l, ADprobe_l = \langle \begin{matrix} ADhost, ADdomain, ADOS, \\ ADverOS, ADServPack \end{matrix} \rangle$$

Кортеж *AD probe* состоит из следующих элементов: *AD host*, *AD domain*, *AD OS*, *AD ver OS*, *AD ServPack*.

$RADIUS\ probe = \cup_{l=1}^{r} RADIUS\ probe_l,\ RADIUS\ probe_l = <$
огра$Calling\ Station\ ID, NAS\ IP\ Address, NAS\ Port, Framed\ IP\ Address, Acct\ Session\ ID$,
$Acct\ Session\ Time$, $Acct\ Terminate\ Cause, On\ For\ Login\ Auth >$;

Кортеж *RADIUS probe* состоит из следующих элементов: *Calling Station ID, NAS IP Address, NAS Port, Framed IP Address, Acct Session ID, Acct Session Time, Acct Terminate Cause, On For Login Auth*.

$DHCP\ probe = \cup_{l=1}^{r} DHCP\ probe_l,\ DHCP\ probe_l = <$
$Dhcp\ class\ identifier, Dhcp\ client\ identifer, Dhcp\ user\ class\ id,$
$Dhcp\ requsted\ address, Dhcp\ server\ identifier, Dhcp\ parameters\ request\ list,$
$Dhcp\ message\ type >$;

Кортеж *DHCP Probe* состоит из следующих элементов: *DHCP class identifier, DHCP client identifier, DHCP user class id, DHCP requested address, DHCP server identifier, DHCP parameters request list, DHCP message type*.

$DNS\ probe = \cup_{l=1}^{r} DNS\ probe_l,\ DNS\ probe_l = < DNS\ FQDN >$;

Кортеж *DNS probe* состоит из следующих элементов: *DNS FQDN*.

$NMAP\ probe = \cup_{l=1}^{r} NMAP\ probe_l,\ NMAP\ probe_l = < Open\ ports >$;

Кортеж *NMAP probe* состоит из следующих элементов: *Open ports*.

Для корректного функционирования необходимо также предварительно подготовить профили нормального поведения пользователей, на основании отклонения от которых определяется, относится ли пользователь к инсайдерам. Чем больше будет количество полей и количество профилей нормального поведения, тем более точно будут определяться инсайдеры в сети. Инсайдеров в приведенной схеме можно определить по следующим параметрам: нестандартное устройство, с которого выполняется подключение; использование приложений, ранее неиспользовавшихся пользователем; время завершения сессии в нерабочее время; отсутствие антивирусного ПО; проявление во время сессии подозрительной сетевой активности от устройства; корректировка нестандартных файлов во время сессии; количество попыток аутентификации перед созданием конкретной сессии.

Модель построена с использованием NoSQL подходов, отличающихся от используемых в традиционных СУБД именно ориентированностью на управление большими данными. Так, вместо ACID требований по атомарности, согласованности, изолированности и долговечности/надежности (Atomicity, Consistency, Isolation, Durability), в NoSQL основной аспект смещен в сторону базовой доступности, гибкого состояния и согласованности в конечном состоянии. Также, отличительными характерными чертами NoSQL-решения является линейная масштабируемость, разнородность типов хранилищ, возможность разработки без задания схемы базы данных и др.

2.2 Алгоритм обнаружения инсайдеров в КС с использованием экспертных правил

Для обнаружения инсайдеров в КС был разработан алгоритм, основанный на экспертных правилах. Предпосылкой для создания экспертных правил послужил вариант типовой политики безопасности в организации (применительно к поведению пользователей в КС), который был создан на основании мнений экспертов в данной области. Очевидно, что приведенная политика безопасности является лишь одним из возможных примеров. Тем не менее, предложенный подход формирования алгоритма может быть применен и к большинству реально действующих политик.

Политика состоит из набора базовых правил, с каждым из которых ассоциировалось множество критериев определения соответствия правилам, а также степень критичности не соответствия им (таблица 2.5).

Таблица 2.5 – Пример возможной политики информационной безопасности поведения пользователя в КС организации

Правило политики	**Критерий соответствия правилу**	**Степень критичности**
1. Предотвращение нарушения функционирования компьютерной сети внутри организации.	Создание большого количества обращений к хостам (за малое время – в течение 1 минуты – около 1000 сообщений).	Запрещено
	Создание большого количества нетиповых запросов в объеме около 1000 запросов в минуту (с флагами TCP Reset, SYN, SYN + FIN).	Запрещено
	Нестандартное завершение сессии.	Запрещено
2. Обеспечение корректного функционирования Active Directory.	Наличие узлов, не являющихся членами домена – является ли устройство корпоративным.	Запрещено
	Отсутствие доменного имени.	Запрещено
3. Предотвращение и определение факта заражения ПК вредоносным ПО.	Несоответствие ОС, ее служб и обновлений необходимым версиям – обновление ПО должно быть осуществлено в течение одного дня после выхода обновления.	Запрещено
	Посещение опасных/запрещенных сайтов – в соответствии со списком запрещенных URL адресов организации.	Подозрительно
	Наличие приложений с нетипичной активностью – для каждого типового приложения задана стандартная активность. Например, приложение Skype активно работает в фоновом режиме.	Подозрительно
	Появление нестандартных подключений – в КС появились приложения или устройства, которые используют нестандартные порты подключения.	Подозрительно

4. Предотвращение человеческого фактора.	Подключение новых периферийных устройств.	Подозрительно
	Появление нестандартных устройств – смена ОС, MAC, IP и пр.	Подозрительно
5. Предотвращение перебора логина и пароля.	Выполнение более 3 неудачных аутентификаций.	Подозрительно
	Выполнение более 10 неудачных аутентификаций.	Запрещено
6. Предотвращение утечки информации за периметр организации.	Отправка большого количества данных вне периметра организации – превышение объема переданной информации за 1 час составляет около 10 Гб по сравнению со средне-дневной отправкой за предыдущие 6 месяцев.	Запрещено
	Отправка данных в объеме более 1 Гб за час вне периметра организации по сравнению со средне-дневной отправкой за предыдущие 6 месяцев.	Подозрительно
7. Предотвращение нелегитимного сбора информации с внутренних ресурсов организации.	Получение большого количества данных внутри периметра с внутренних ресурсов организации – превышение объема полученной информации за 1 час составляет около 10 Гб по сравнению со средне-дневным трафиком, полученным за предыдущие 6 месяцев.	Запрещено
	Получение данных в объеме более 1 Гб за час с внутренних серверов за 1 час по сравнению со средне-дневной отправкой за предыдущие 6 месяцев.	Подозрительно

8. Подотчетность действий пользователей.	Начало сессии в нерабочее время (с 18.00 вечера до 9.00 утра; совещание в пятницу с 13.00 до 14.00).	Подозрительно
	Завершение сессии в нерабочее время (с 18.00 вечера до 9.00 утра; совещание в пятницу с 13.00 до 14.00).	Подозрительно

Примечание. В таблице 2.5 в столбце **Степень критичности** указана важность критерия для соответствия политике безопасности. Так, значение «Запрещено» означает недопустимость несоответствия сессии пользователя данному критерию, а значение «Подозрительно» – необходимость обратить внимание на факт несоответствия (например, если пользователь одновременно посещает опасные сайты, подключает к сети собственный ноутбук, работает на выходных и устанавливает приложение с нетипичной сетевой активностью), что может служить дополнительным внешним признаком для распознавания инсайдера.

Согласно таблице 2.5, было сформировано 8 правил политики безопасности (касательно сетевой активности пользователей и их устройств), проверка каждого из которых осуществляется по 2 – 4 критериям.

Важным следствием из правил и критериев является то, что в каждом правиле соответствие одной части критериев может быть определено точно, а соответствие другой части – с некоторой степенью вероятности. Так, например, для правила политики №1 (первая строка таблицы), создание большого количества обращений к хостам и нетиповых запросов определимо условно (по причине условности термина – *большое количество*), в то время как «нестандартное завершение сессии» определяется однозначно (поскольку этому соответствует *неожиданный* разрыв сессии). Тем не менее, исходя из того, что в реальной работающей системе бывают нештатные сбои (например, нарушение типовой последовательности пакетов из-за перегруженности сети; скачивание на ПК больших по объему обновлений крупных пакетов; кратковременное нарушение функционирования сети из-за инсталляции на сервер пакета обновления для уязвимости нулевого дня и его принудительной перезагрузки; кратковременная потеря и появление подключения в нерабочее

время), даже при точном соответствии сетевой активности пользователя критериям, непосредственное нарушение политики безопасности может отсутствовать. Недопущение таких ошибочных, непреднамеренных срабатываний, требует от алгоритмов обнаружения инсайдеров учета огромного количества параметров КС, неявно связанных и сложно определимых, что выходит за рамки предметной области.

На основании предложенной политики информационной безопасности и предложенного подхода был разработан следующий алгоритм на базе экспертных правил. Алгоритм условно можно разделить на: (1) основную ветку выполнения, отвечающую за выбор области проверки для применения правил, и (2) подпрограммы, отвечающие за детализацию отдельных правил.

Блок-схема основной ветки алгоритма представлена на рисунке 2.3.

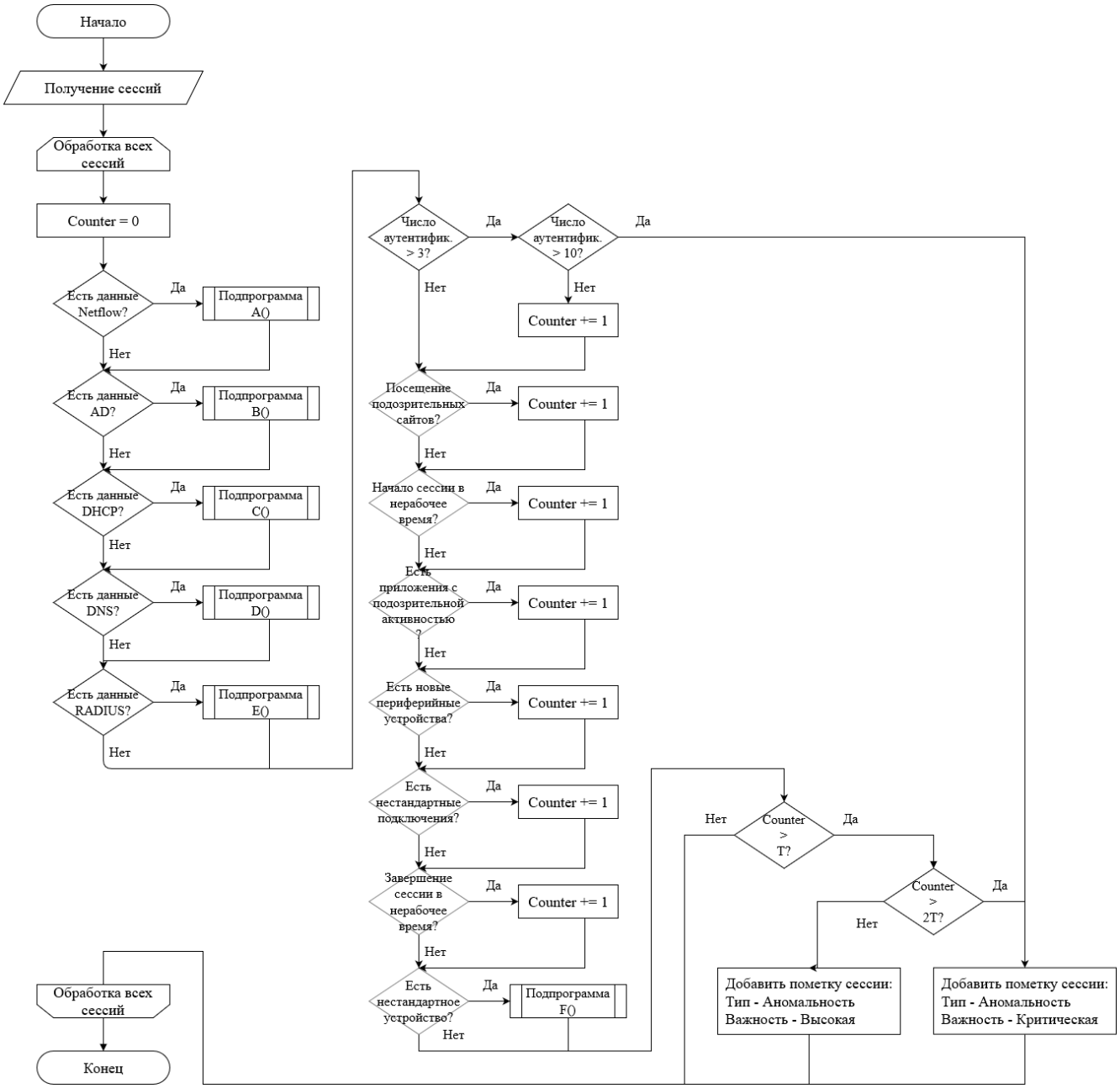

Рисунок 2.3 – Блок-схема алгоритма определения аномалий на основе экспертных правил

Алгоритм начинается с синтаксического анализа информации и разделения на отдельные сессии.

В каждой сессии проверяется наличие данных от различных проб. Если Netflow данные присутствуют, то задействуется модуль по их анализу; для этого в него предаются данные по текущей сессии. По данной сессии запрашиваются все Netflow пакеты и происходит их комплексный анализ: так, например, происходит

детектирование угрозы типа сканирования сети. То есть, если за короткий промежуток времени происходит большое количество единичных обращений к разным адресам, то к данной сессии добавляется пометка о сканировании сети. С помощью данного модуля также можно обнаружить: DoS атаки, фальшивые TCP reset атаки типа SYN-флуд, SYN-FIN, распространение вредоносного кода, а также утечку данных. Для анализа резких изменений объема сетевого трафика используются поля модели – массивы *Income Traffic []* и *Outcome Traffic []*.

Аналогично работают модули AD и RADIUS, которые способны определить нестандартное завершение сессии, неактуальную версию ОС или неопознанных пользователей.

После проверки наличия данных от проб проверяется число аутентификаций. Если количество попыток аутентификации больше 10, то сессия сразу же помечается, как аномальная с уровнем важности «критическая». Если количество попыток аутентификации меньше 10, но больше 3, тогда к счетчику (Counter) прибавляется единица.

Далее проверяются такие параметры как посещение подозрительных сайтов, наличие новых периферийных устройств, завершение сессии в нерабочее время, наличие приложений с подозрительной сетевой активностью, нестандартный тип подключений.

Каждый из данных параметров увеличивает счетчик на единицу. В конце проверяется конечное устройство: если это обычный пользовательский ПК, подключенный через кабель к зарегистрированному порту, и находящийся в защищенном периметре организации, то такое устройство определяется как стандартное. Если устройство определено как нестандартное, в этом случае запускается дополнительный модуль, в котором собирается информация со всех проб; кроме этого, данное нестандартное устройство проверяется с помощью NMAP пробы. Данные от различных проб анализируются и выявляется несоответствие либо MAC-адресов, либо операционной системе, что может говорить о попытке скрыть исходные данные устройства.

Помимо этого, в данном модуле проверяются потенциально уязвимые устройства, для чего используется результат NMAP-проб, который содержит информацию об открытых портах. Например, если тип устройства был определен как Web-сервер, а на нем были открыты нестандартные порты, например, порт 445, в этом случае устройство помечается как потенциально уязвимое. В некоторых случаях даже открытый 22 порт может говорить об уязвимости. Так как доступ по данному порту должен быть только у администраторов сервера и, если бы конфигурация была правильной, NMAP-пробе не удалось бы подключиться по нему.

В итоге значение счетчика сравнивается с уровнем доверенности пользователя (параметр T в модели), сессия которого анализировалась. По результатам сравнения, а также в процессе работы к сессиям могут добавляться пометки, содержащие тип потенциальных инсайдерских действий и уровень их важности; сессии с такими пометками будут считаться инсайдерскими.

Тип действия в пометке к сессии может принимать следующие значения, ассоциируемые с потенциальными атаками:

1. DoS атака
2. SYN-FIN запрос
3. SYN-флуд
4. Аномальная сессия
5. Неактуальная версия ОС
6. Неопознанный пользователь
7. Несоответствие MAC адреса
8. Несоответствие ОС адреса
9. Нестандартное завершение сессии
10. Отсутствует доменное имя
11. Распространение вредоносного кода
12. Сканирование сети
13. Сбор информации
14. Утечка информации

15. Уязвимое устройство

16. Фальшивые TCP Reset

Важность действия в пометке к сессии может принимать следующие значения:

1. Средняя – действия являются подозрительными, однако они могут быть следствием случайных сбоев в работе КС и не быть причиной инсайдерской деятельности.

2. Высокая – хотя причиной действий потенциально и являются инсайдеры, тем не менее действия, скорее всего, напрямую не приведут к нарушению информационной безопасности в КС;

3. Критическая – действия могут привести к существенному нарушению информационной безопасности в КС и требуют немедленной реакции (от ручного анализа до автоматической блокировки пользователя).

Описание подпрограмм, согласно их обозначениям на рисунке 2.3 и с использованием псевдокода, приведено далее.

Подпрограмма A

```
// Получение данных Netflow по текущей сессии
Data = GetCurrentSessionData(NetFlow)
// Если большое количество единичный обращений к разным адресам
IF Data.OneSourceToManyTarget.Num > MAX_OSTMT_NUM
// за короткий промежуток времени
AND Data.OneSourceToManyTarget.Time < MIN_OSTMT_TIME
  // То добавить пометку к сессии:
  // Тип “Сканирование сети” с важностью “Средняя”
  THEN Date.AddSessionMark(Сканирование сети, Средняя)
// Конец условия
FI
// Если большое количество пакетов к одному хосту
IF Data.ManySourceToOneTarget.Num > MAX_MSTOT_NUM
// за короткий промежуток времени
AND Data.ManySourceToOneTarget.Time < MIN_MSTOT_TIME
  // То добавить пометку к сессии:
```

```
  // Тип “DoS атака” с важностью “Высокая”
  THEN Date.AddSessionMark(DoS атака, Высокая)
// Конец условия
FI
// Если большое количество обращений со значением флага TCP Reset
IF Data.TcpPackets.HasFlags(TCP_Reset).Num > MAX_TCP_RESET_FLAG
  // То добавить пометку к сессии:
  // Тип “Фальшивые TCP Reset” с важностью “Средняя”
  THEN Date.AddSessionMark(Сканирование сети, Средняя)
// Конец условия
FI
// Если большое количество обращений со значением флага SYN запросов
IF Data.TcpPackets.HasFlags(SYN).Num > MAX_SYN_FLAG
// за короткий промежуток времени
AND Data.TcpPackets.HasFlags(SYN).Time < MIN_TCP_WITH_SYN_FLAG_TIME
  // То добавить пометку к сессии:
  // Тип “SYN-флуд” с важностью “Средняя”
  THEN Date.AddSessionMark(SYN-флуд, Средняя)
// Конец условия
FI
// Одновременное обращение со значением флагов SYN и FIN
IF Data.TcpPackets.HasFlags(SYN, FIN).Num > 0
  // То добавить пометку к сессии:
  // Тип “SYN-FIN запрос” с важностью “Высокая”
  THEN Date.AddSessionMark(SYN-FIN запрос, Высокая)
// Конец условия
FI
// Начало цикла: Перебор всех адресов трафика
FOR_EACH Address IN Data.AddressList
  // Если резкое увеличение
  IF Date.FlowsFrom(Address).Rate > MAX_FFR_RATE
  // исходящего трафика на разные адреса
  AND Date.FlowsFrom(Address).Targets.List > MAX_FTTA_LIST
    // То добавить пометку к сессии:
```

```
    // Тип “Распространение вредоносного кода” с важностью “Высокая”
    THEN Date.AddSessionMark(Распространение вредоносного кода, Высокая)
  // Конец условия
  FI
// Конец цикла
EACH_FOR
// Начало цикла: Перебор всех адресов трафика
FOR_EACH Address IN Data.AddressList
  // Если резкое увеличение
  IF Date.FlowsFrom(Address).Rate > MAX_FFR_RATE
  // исходящего трафика на конкретные адреса
  AND Date.FlowsTo(Address).Source.List > MAX_FTSA_LIST
    // То добавить пометку к сессии:
    // Тип “Утечка информации” с важностью “Высокая”
    THEN Date.AddSessionMark(Утечка информации, Высокая)
  // Конец условия
  FI
// Конец цикла
EACH_FOR
// Начало цикла: Перебор всех адресов трафика
FOR_EACH Address IN Data.AddressList
  // Если резкое увеличение
  IF Date.FlowsTo(Address).Rate > MAX_FTR_RATE
  // входящего трафика с конкретных адресов
  AND Date.FlowsFrom(Address).Source.List > MAX_FFSA_LIST
    // То добавить пометку к сессии:
    // Тип “Сбор информации” с важностью “Высокая”
    THEN Date.AddSessionMark(Сбор информации, Высокая)
  // Конец условия
  FI
// Конец цикла
EACH_FOR
```

Подпрограмма B

```
// Получение данных ActiveDirectory по текущей сессии
```

```
Data = GetCurrentSessionData(ActiveDirectory)
// Если хост не член домена
IF Date.Host IS_NOT_IN Domain.HostList
  // То добавить пометку к сессии:
  // Тип “Неопознанный пользователь” с важностью “Средняя”
  THEN Date.AddSessionMark(Неопознанный пользователь, Средняя)
// Конец условия
FI
// Если ОС не соответствует политике безопасности
IF Date.OS IS_NOT_IN SecurePolitic.OSList
// Или версия не соответствует политике безопасности
OR Date.Version IS_NOT_IN Domain.VersionList
// Или пакет обновлений не соответствует политике безопасности
OR Date.Update IS_NOT_IN Domain.UpdateList
  // То добавить пометку к сессии:
  // Тип “Неактуальная версия ОС” с важностью “Средняя”
  THEN Date.AddSessionMark(Неактуальная версия ОС, Средняя)
// Конец условия
FI
```

Подпрограмма C

```
// Получение данных DHCP по текущей сессии
Data = GetCurrentSessionData(DHCP)
// Если отсутствует FQDN имя
IF Date.FQDN IS_NOT_IN Domain.FQDNList
  // То добавить пометку к сессии:
  // Тип “Отсутствует доменное имя” с важностью “Средняя”
  THEN Date.AddSessionMark(Отсутствует доменное имя, Средняя)
// Конец условия
FI
```

Подпрограмма D

```
// Получение данных DNS по текущей сессии
Data = GetCurrentSessionData(DNS)
// Если отсутствует DNS имя
IF Date.FQDN IS_NOT_IN Domain.FQDNList
```

```
  // То добавить пометку к сессии:
  // Тип “Отсутствует доменное имя” с важностью “Средняя”
  THEN Date.AddSessionMark(Отсутствует доменное имя, Средняя)
// Конец условия
FI
```

Подпрограмма E

```
// Получение данных RADIUS по текущей сессии
Data = GetCurrentSessionData(RADIUS)
// Если подозрительный обрыв сессии
IF NOT Date.NormalSessionClose
  // То добавить пометку к сессии:
  // Тип “Нестандартное завершение сессии” с важностью “Средняя”
  THEN Date.AddSessionMark(Нестандартное завершение сессии, Средняя)
// Конец условия
FI
```

Подпрограмма F

```
// Получение данных MAC по текущей сессии
Data = GetCurrentSessionData(MAC)
// Получение данных ОС по текущей сессии
Data += GetCurrentSessionData(OS)
// Запуск Nmap-сканирования портов текущего устройства – N проб
Data += NMapScaner.RunForCurrentDevice(N)
// Если MAC адреса разные для различных проб (более 1-го адреса)
IF Data.MACList.Length > 1
  // То добавить пометку к сессии:
  // Тип “Несоответствие MAC адресов” с важностью “Высокая”
  THEN Date.AddSessionMark(Несоответствие MAC адресов, Высокая)
// Конец условия
FI
// Если ОС разные для различных проб (более 1-го типа)
IF Data.OSList.Length > 1
  // То добавить пометку к сессии:
  // Тип “Несоответствие ОС” с важностью “Средняя”
  THEN Date.AddSessionMark(Несоответствие ОС, Средняя)
```

```
// Конец условия
FI
// Если открыты нестандартные порты для текущего устройства
IF Data.OpenedPortList IS_NOT_IN Global.StandartPortList
  // То добавить пометку к сессии:
  // Тип “Потенциально уязвимое устройство” с важностью “Средняя”
  THEN Date.AddSessionMark(Потенциально уязвимое устройство, Средняя)
// Конец условия
FI
```

Вся работа алгоритма определения аномалий с использованием правил может быть описана с помощью следующих шагов:

Основная ветка алгоритма

Шаг 1. Разделение входного потока данных на список сессий.

Шаг 2. Обработка каждой сессии в цикле (счетчик доверенности пользователя обнуляется). По окончанию обработки сессий происходит выход из алгоритма. Последующие шаги выполняются последовательно (с вызовом подпрограмм по условию).

Шаг 3. Проверка наличия данных Netflow. В случае их присутствия запускается Подпрограмма A; в ином случае счетчик увеличивается на 3.

Шаг 4. Проверка наличия данных ActiveDirectory. В случае их присутствия запускается Подпрограмма B; в ином случае счетчик увеличивается на 3.

Шаг 5. Проверка наличия данных RADIUS. В случае их присутствия запускается Подпрограмма C; в ином случае счетчик увеличивается на 3.

Шаг 6. Проверка наличия данных DHCP. В случае их присутствия запускается Подпрограмма D; в ином случае счетчик увеличивается на 3.

Шаг 7. Проверка наличия данных DNS. В случае их присутствия запускается Подпрограмма E; в ином случае счетчик увеличивается на 3.

Шаг 8. Проверка превышения 3-х попыток аутентификации пользователем. В случае превышения 10-ти попыток аутентификации к сессии добавляется пометка “Аномальная с Критической важностью” и происходит переход на Шаг 2.

В случае превышения 3-х попыток, но менее 10-ти попыток, счетчик увеличивается на 1.

Шаг 9. Проверка посещения пользователем подозрительных сайтов. В случае подтверждения посещения счетчик увеличивается на 1.

Шаг 10. Проверка начала или завершения сессии пользователем в нерабочее время. В случае подтверждения этого счетчик увеличивается на 1.

Шаг 11. Проверка наличия у пользователя приложений с подозрительной активностью. В случае подтверждения этого счетчик увеличивается на 1.

Шаг 12. Проверка появления новых периферийных устройств пользователя. В случае подтверждения этого счетчик увеличивается на 1.

Шаг 13. Проверка обнаружения нестандартных подключений у пользователя. В случае подтверждения этого счетчик увеличивается на 1.

Шаг 15. Проверка подключения пользователем нестандартных устройств. В случае подтверждения этого запускается Подпрограмма F.

Шаг 16. Сравнение счетчика с уровнем доверия пользователя. Если значение счетчика больше или равно уровню доверия пользователя, но меньше его удвоенного значения, то к сессии добавляется пометка “Аномальная с Высокой важностью”. Если значение счетчика больше или равно удвоенного значения уровня доверия пользователя, то к сессии добавляется пометка “Аномальная с Критической важностью”.

Подпрограмма A

Шаг 1. Получение Netflow данных по текущей сессии.

Шаг 2. Проверка адресов единичных обращений пользователя и их временных меток. В случае большого количества адресов за короткий промежуток времени, к сессии добавляется пометка “Сканирование сети со Средней важностью”.

Шаг 3. Проверка сетевых пакетов от пользователя и их временных меток. В случае большого количества пакетов на один порт за короткий промежуток времени, к сессии добавляется пометка “DoS атака со Средней важностью”.

Шаг 4. Проверка флагов TCP пакетов от пользователя. В случае большого количества обращений с флагом Reset, к сессии добавляется пометка “Фальшивые TCP Reset со Средней важностью”.

Шаг 5. Проверка флагов TCP пакетов от пользователя и их временных меток. В случае большого количества запросов с флагом SYN за короткий промежуток времени, к сессии добавляется пометка “SYN-флуд со Средней важностью”.

Шаг 6. Проверка флагов TCP пакетов от пользователя. В случае большого количества обращений одновременно с флагами SYN и FIN, к сессии добавляется пометка “SYN-FIN запрос с Высокой важностью”.

Шаг 7. Проверка потоков трафика, исходящего от пользователя. В случае резкого увеличения количества трафика, направленного на разные адреса, к сессии добавляется пометка “Распространение вредоносного кода с Высокой важностью”.

Шаг 8. Проверка потоков исходящего от пользователя трафика. В случае резкого увеличения количества трафика, направленного на конкретный адрес, к сессии добавляется пометка “Утечка информации с Высокой важностью”.

Подпрограмма B

Шаг 1. Получение данных Active Directory по текущей сессии.

Шаг 2. Проверка принадлежности AD-хоста пользователя членам домена. В случае не подтверждения принадлежности AD-хоста, к сессии добавляется пометка “Неопознанный пользователь со Средней важностью”.

Шаг 3. Проверка соответствия программного обеспечения пользователя политике безопасности. В случае несоответствия операционной системы, версии или пакета обновлений политике, к сессии добавляется пометка “Неактуальная версия ОС со Средней важностью”.

Подпрограмма C

Шаг 1. Получение DHCP данных по текущей сессии.

Шаг 2. Проверка наличия FQDN имени пользователя. В случае отсутствия имени, к сессии добавляется пометка “Отсутствует доменное имя со Средней важностью”.

Подпрограмма D

Шаг 1. Получение DNS данных по текущей сессии.

Шаг 2. Проверка наличия FQDN имени пользователя. В случае отсутствия имени, к сессии добавляется пометка “Отсутствует доменное имя со Средней важностью”.

Подпрограмма E

Шаг 1. Получение RADIUS данных по текущей сессии.

Шаг 2. Проверка корректности завершения сессии пользователя. В случае подозрительного обрыва сессии, к ней добавляется пометка “Нестандартное завершение сессии со Средней важностью”.

Подпрограмма F

Шаг 1. Получение MAC и ОС по текущей сессии.

Шаг 2. Запуск NMAP-сканирования (проб) по текущему устройству пользователя.

Шаг 3. Проверка совпадения MAC адресов для различных проб. В случае несовпадения адресов, к сессии добавляется пометка “Несоответствие MAC адреса с Высокой важностью”.

Шаг 4. Проверка совпадения ОС для различных проб. В случае несовпадения ОС, к сессии добавляется пометка “Несоответствие ОС адреса со Средней важностью”.

Шаг 5. Проверка открытых портов устройства. В случае нестандартных портов для текущего типа, к сессии добавляется пометка “Уязвимое устройство со Средней важностью”.

Алгоритм на базе экспертных правил, очевидно, ограничен возможностями экспертов (их квалификацией, опытом, человеческими факторами), задающих логику и настройку правил; поэтому, применение только данного алгоритма в интересах обнаружения инсайдеров может оказаться недостаточно, для чего потребуется создание модели и алгоритмов комбинированного применения экспертных правил и методов машинного обучения в интересах обнаружения инсайдерских атак. При создании комплекса алгоритмов предполагается

использовать предложенный алгоритм на основе экспертных правил, а также алгоритмы, основанные на известных методах машинного обучения.

Алгоритм использует обработку больших данных в следующих частях. Во-первых, подпрограммы алгоритма производят анализ сетевых потоков хоста по топологии 1:M – «Один ко Многим», – что соответствует первому V-признаку представления больших данных – Объем (Volume). Во-вторых, в алгоритмах учитывается *резкость* изменений трафика, требующая своевременного учета всех предыдущих значений, что соответствует второму V-признаку представления больших данных – Скорость (Velocity). И, в-третьих, в основу алгоритма положена обработка параметров, характеризующих поведение пользователей с множества точек зрения, что соответствует третьему V-признаку представления больших данных – Разнообразие (Variety).

2.3 Модель и алгоритмы комбинированного применения экспертных правил и методов машинного обучения в интересах обнаружения инсайдерских атак

Исходя из поставленных функциональных требований, разрабатываемая система должна сочетать несколько успешных подходов к обнаружению инсайдеров в КС. Этого можно достичь комбинацией различных алгоритмов обнаружения инсайдеров следующим образом.

Поскольку каждый из алгоритмов на *входе* имеет одни и те же данные – атрибуты поведения пользователей, а на выходе – идентификаторы обнаруженных инсайдеров, то возможно их параллельное выполнение с *комбинированием* результатов одним из способов. Существует 4 наиболее распространенных способа комбинирования результатов с позиции работы с множествами (два последних из которых можно считать вырожденными, но тем не менее, необходимыми для рассмотрения):

1) объединение – результат работы комплекса включает в себя инсайдеров, обнаруженных любым из алгоритмов (операция – ∨);

2) пересечение – результат работы комплекса включает в себя инсайдеров, обнаруженных обоими алгоритмами одновременно (операция – ∧);

3) только первый – результат работы комплекса включает в себя инсайдеров, обнаруженных только первым из алгоритмов (операция – *I*);

4) только второй – результат работы комплекса включает в себя инсайдеров, обнаруженных только вторым из алгоритмов (операция – *II*).

Графическая интерпретация способов комбинирования приведена на рисунке 2.4 (пунктирной красной линией обозначен результат комбинирования).

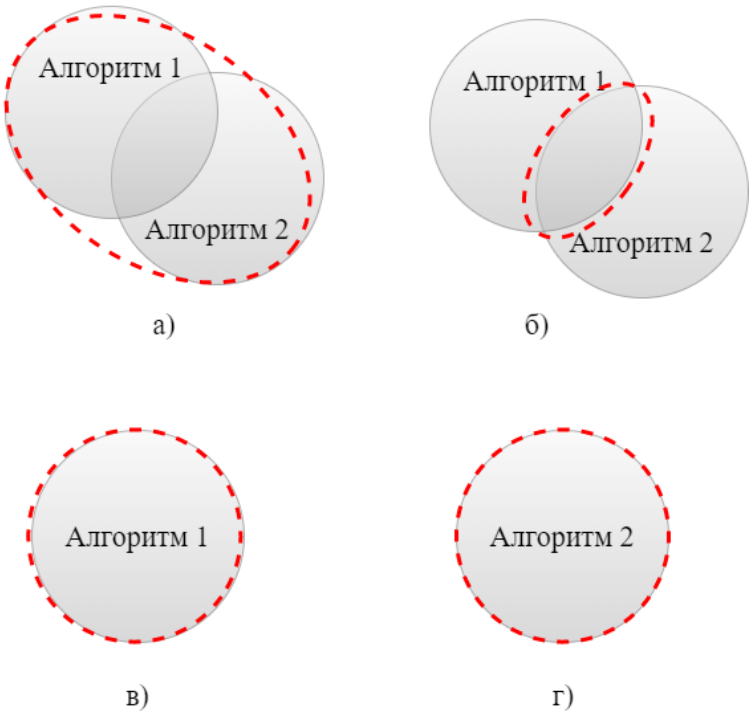

Рисунок 2.4 – Графическая интерпретация способов комбинирования: а) объединение, б) пересечение, в) только 1-й, г) только 2-й

Выбор одной из формул вычисления, связанной с соответствующим способом, очевидно должен показывать наилучшие значения мер качества работы

системы. Такой обоснованный выбор будет произведен при помощи соответствующей экспериментальной оценки.

Схематично, меры качества TP, TN, FP, TN работы любого алгоритма обнаружения инсайдеров могут быть представлены в графическом виде следующим образом (рисунок 2.5).

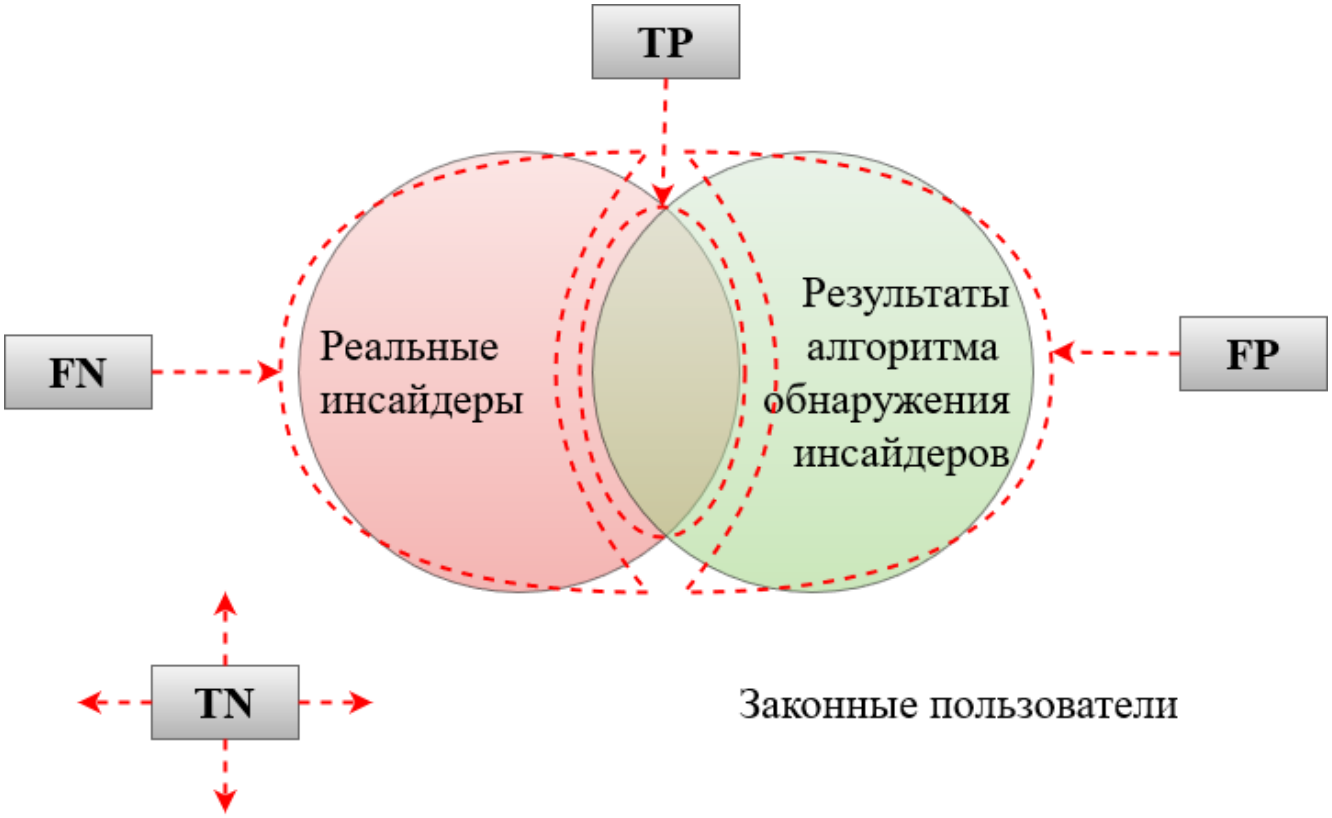

Рисунок 2.5 – Графическая интерпретация метрик качества работы алгоритма обнаружения инсайдеров

На рисунке 2.5 показаны две области инсайдеров: реально действующих в КС (красный круг) и обнаруженных алгоритмом (зеленый круг); вся белая область вокруг кругов соответствует законным пользователям. Таким образом, пересечение красного и зеленого кругов означает верное срабатывание алгоритма (мере TP), а отсутствие кругов (белая область) – верное определение алгоритмом законных пользователей (TN); при этом часть реальных инсайдеров не были обнаружены алгоритмом (FN или ошибка II-го рода – пропуск цели), а часть законных пользователей алгоритм ошибочно посчитал как инсайдеров (FP или

ошибка I-го рода – ложная тревога). Данная схема является общей и в диссертационной работе.

Исходя из вышесказанного, предложим следующую функциональную структуру комплекса алгоритмов (K_A) обнаружения инсайдеров в КС, состоящую из двух алгоритмов (A_1 и A_2), скомбинированных одним из описанных выше способов. Формальная запись комплекса алгоритмов имеет следующий вид:

$$\begin{cases} K_A = \{A_1 \otimes A_2\} \\ \otimes \in \{I, II, \vee, \wedge\} \end{cases},$$

где $\otimes$ – операции комбинирования, I – результат работы комплекса включает в себя инсайдеров, обнаруженных только первым из алгоритмов, II – результат работы комплекса включает в себя инсайдеров, обнаруженных только вторым из алгоритмов, $\vee$ – результат работы комплекса включает в себя инсайдеров, обнаруженных любым из алгоритмов, $\wedge$ – результат работы комплекса включает в себя инсайдеров, обнаруженных обоими алгоритмами одновременно.

Составим комплекс из следующих алгоритмов: в качестве 1-го возьмем алгоритм на основе экспертных правил, описанный в п.2.2, а в качестве 2-го – известный алгоритм на основе методов машинного обучения, который может иметь несколько комбинаций согласно выбранным классификаторам. Так, под алгоритмом на рисунке 2.5 будет пониматься их комплекс – на основе экспертных правил и методов машинного обучения. Комбинация алгоритмов и выбор наилучшего классификатора направлены на снижение ошибок I-го и II-го рода.

Модель комбинированного применения алгоритмов обнаружения инсайдеров в КС. Комбинированное применение алгоритмов может быть представлено в виде модели, представленной в графическом виде на рисунке 2.6.

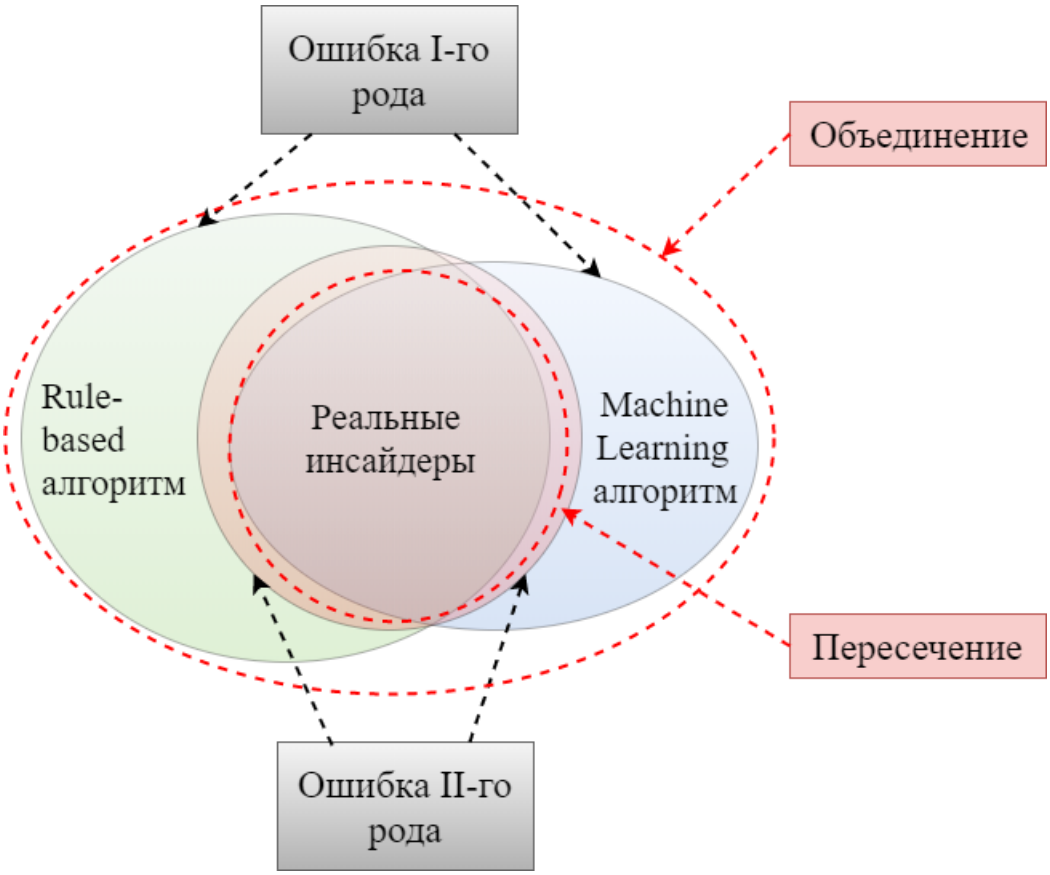

Рисунок 2.6 – Графическая интерпретация модели комбинирования алгоритмов обнаружения инсайдеров в КС

Модель отражает взаимосвязь следующих сущностей, представленных на рисунке с помощью 3-х эллиптических областей и ассоциированные с инсайдерами: красная – реальные инсайдеры в КС (I_0), зеленая – инсайдеры, обнаруженные алгоритмом на основе экспертных правил (I_{RB}), синяя – инсайдеры, обнаруженные алгоритмом на основе методов машинного обучения – для некоторого классификатора (I_{ML}).

Каждая из зеленой и синей областей (на рисунке 2.6) соответствует отдельному результату работы каждого из алгоритмов, объединение этих областей – объединению результатов работы алгоритмов, пересечение – соответственно пересечению результатов их работы. Пересечение красной области с результатом работы одного из способов комбинирования алгоритмов соответствует мере TP – то есть верно обнаруженным инсайдерам, а область вне красной, зеленой и синей областей соответствует мере TN – от есть пользователям, верно не отнесенным к

инсайдерам. По аналогии, зеленая или синяя области, не пересекаемые с красной соответствуют мере FP (или ошибке I рода) – то есть инсайдеру, не обнаруженному алгоритмом; а красная область, не пересекаемая с результатом работы одного из способов комбинирования алгоритмов соответствует мере FN (или ошибке II рода) – то есть пропуску инсайдера комплексом.

Очевидно, идеальной работой комплекса алгоритмов (то есть результатом I) будет ситуация, когда или пересечение, или объединение алгоритмов совпадет с реальными инсайдерами (то есть, когда FP и FN тождественны 0). Однако, такая ситуация редко достижима по причине того, что любой из алгоритмов может как пропускать часть инсайдеров, так и определять их неверно.

В рамках формальной записи комплекса алгоритмов модель может быть описана следующим образом (для каждой комбинации):

$$\text{а)}\quad \text{для } I = I_{RB}\colon \begin{cases} I_{TP} = I_0 \wedge I_{RB} \\ I_{TN} = \overline{I_0 \vee I_{RB}} \\ I_{FP} = I_{RB} \backslash I_0 \\ I_{FN} = I_0 \backslash I_{RB} \end{cases},$$

$$\text{б)}\quad \text{для } I = I_{ML}\colon \begin{cases} I_{TP} = I_0 \wedge I_{ML} \\ I_{TN} = \overline{I_0 \vee I_{ML}} \\ I_{FP} = I_{ML} \backslash I_0 \\ I_{FN} = I_0 \backslash I_{ML} \end{cases},$$

$$\text{в)}\quad \text{для } I = I_{RB} \wedge I_{ML}\colon \begin{cases} I_{TP} = I_0 \wedge (I_{RB} \wedge I_{ML}) \\ I_{TN} = \overline{I_0 \vee (I_{RB} \wedge I_{ML})} \\ I_{FP} = (I_{RB} \wedge I_{ML}) \backslash I_0 \\ I_{FN} = I_0 \backslash (I_{RB} \wedge I_{ML}) \end{cases},$$

$$\text{г)}\quad \text{для } I = I_{RB} \vee I_{ML}\colon \begin{cases} I_{TP} = I_0 \wedge (I_{RB} \vee I_{ML}) \\ I_{TN} = \overline{I_0 \vee (I_{RB} \vee I_{ML})} \\ I_{FP} = (I_{RB} \vee I_{ML}) \backslash I_0 \\ I_{FN} = I_0 \backslash (I_{RB} \vee I_{ML}) \end{cases},$$

где $I_{TP}, I_{TN}, I_{FP}, I_{FN}$ – множества, соответствующие каждой из мер. Очевидно, что если для идеального случая работы комплекса (то есть для $I = I_0$): $\begin{cases} I_{TP} = I_0 \\ I_{TN} = \overline{I_0} \\ I_{FP} = \emptyset \\ I_{FN} = \emptyset \end{cases}$, где

соответственно $\overline{I_0} = U - I_0$ – все законные пользователи (для всего множества пользователей КС – U).

Используя полученные выражения могут быть посчитаны основные меры качества работы комплекса алгоритмов (оперируя мощностью множества $|X|$ – соответствующему количеству его элементов, и некоторым способом комбинации – $\otimes \in \{I, II, \vee, \wedge\}$):

а) полнота: $r = \frac{|I_0 \wedge (I_{RB} \otimes I_{ML})|}{|I_0 \wedge (I_{RB} \otimes I_{ML})| + |I_0 \setminus (I_{RB} \otimes I_{ML})|}$;

б) точность: $p = \frac{|I_0 \wedge (I_{RB} \otimes I_{ML})|}{|I_0 \wedge (I_{RB} \otimes I_{ML})| + |(I_{RB} \otimes I_{ML}) \setminus I_0|}$;

в) аккуратность: $a = \frac{|I_0 \wedge (I_{RB} \otimes I_{ML})| + |I_0 \vee (I_{RB} \otimes I_{ML})|}{|I_0 \wedge (I_{RB} \otimes I_{ML})| + |(I_{RB} \otimes I_{ML}) \setminus I_0| + |I_0 \vee (I_{RB} \otimes I_{ML})| + |I_0 \setminus (I_{RB} \otimes I_{ML})|}$;

г) ошибка: $e = \frac{|(I_{RB} \otimes I_{ML}) \setminus I_0| + |I_0 \setminus (I_{RB} \otimes I_{ML})|}{|I_0 \wedge (I_{RB} \otimes I_{ML})| + |(I_{RB} \otimes I_{ML}) \setminus I_0| + |I_0 \vee (I_{RB} \otimes I_{ML})| + |I_0 \setminus (I_{RB} \otimes I_{ML})|}$;

F-мера, соответственно, вычисляется из этих 4-х мер: $f = \frac{2 \times p \times r}{p + r}$.

Методы машинного обучения. Рассмотрим следующие методы машинного обучения. На нижнем уровне в виде базовых классификаторов расположены: метод деревьев решений (DT), наивный байесовский классификатор (NB), метод k-ближайших соседей (k-NN), метод опорных векторов (SVM). На верхнем уровне в виде композиций базовых классификаторов – голосование большинством (PV), взвешенное голосование (WV) и мягкое голосование (SV), а также Adaboost.

Метод деревьев решений (DT). Метод представляет собой иерархическую структуру, содержащую числовые атрибуты и предикаты в виде нетерминальных узлов и обозначения классов в виде терминальных узлов. При прохождении вниз по дереву на основании истинности некоторого предиката для векторного компонента, выбирается один из двух путей, представленных следующей формулой:

$$F(z) = R(T, z)$$

$$R(T,z) = \begin{cases} C, & \text{если T содержит только терминальный узел} \\ R\left(T^{(L)}, z\right), & \text{если } P^{(T)}(z) \\ R\left(T^{(R)}, z\right), & \text{если не } P^{(T)}(z) \end{cases}$$

Где $T^{(L)}$ и $T^{(R)}$ – левые и правые поддеревья множества T, C – метка класса, $P^{(T)}$ – предикат из корня дерева.

Наивный байесовский классификатор (NB). Это семейство методов классификации, которые принимают одно допущение: каждый параметр классифицируемых данных рассматривается независимо от других параметров класса. Вероятностная модель для классификатора – это условная модель p(C | F1, . . ., Fn) над зависимой переменной класса C с малым количеством результатов или классов, зависимая от нескольких переменных F1…Fn. Она выражается следующей формулой:

$$F(z) = \arg\max_{c \in \Omega}[P(z|c)P(c)]$$

Метод k-ближайших соседей (k-NN) позволяет сопоставить анализируемому вектору метку класса, экземпляры которого обладают наибольшим числом среди всех K обучающих объектов, ближайших к данному вектору z. Данный подход можно представить следующим образом:

$$F(z) = \arg\max_{c \in \Omega} \sum_{i=1}^{K} [x_i' \in c]$$

где $x_1', \ldots, x_K'$ – обучающие векторы, для которых значение $\sum_{i=1}^{K} \|z - x_i'\|$ минимально среди всех обучающих векторов; Ω – классы. Можно сказать, что этот метод не требует предварительной настройки (обучения). Для его функционирования достаточно сохранить весь тренировочный образец.

Метод опорных векторов (SVM). Логика метода заключается в построении разделительной гиперплоскости, имеющей свойство равноудаленности от

объектов, находящихся ближе всего к ней и принадлежащих к разным классам. Модель представляется в виде следующей формулы:

$$F(z) = sign(-b + \sum_{i=1}^{M_s} w_i x_i^T z)$$

где w_i – весовой коэффициент, представляющий собой произведение ненулевых множителей Лагранжа и желаемых выходных значений; x_i – опорные векторы (i = 1, …, M_s); b – параметр смещения; M_s – количество опорных векторов. В этой формуле предполагается, что тренировочная выборка может быть линейно разделена, в противном случае необходимо использовать специальные преобразования – ядра.

Комбинируя вышеперечисленные классификаторы, можно получить три композиции: (1) PV – голосование большинством, (2) WV – взвешенное голосование и (3) SV – мягкое голосование, а также Adaboost.

PV представляется в следующем виде:

$$G(z) = \arg\max_{c \epsilon \Omega} \sum_{i=1}^{N} \left[F^{(i)}(z) = c\right],$$

где N = 4 число базовых классификаторов, которые предстоит скомбинировать.

Взвешенное голосование представлено в виде улучшения предыдущего варианта, где происходит взвешивание коэффициентов классификатора и выдается пропорциональная оценка правильности работы метода, выражаемая следующей формулой:

$$G(z) = \arg\max_{c \epsilon \Omega} \sum_{i=1}^{N} w_i \cdot \left[F^{(i)}(z) = c\right],$$

$$w_i = \frac{\#\{x_j | F^{(i)}(x_j) = c_j\}_{j=1}^{M}}{\sum_{k=1}^{N} \#\{x_j | F^{(k)}(x_j) = c_j\}_{j=1}^{M}}$$

где весовые коэффициенты w_i рассчитываются так, что их сумма равна 1.

Тогда мягкое голосование может быть рассмотрено как расширение предыдущей формулы, представляя весовые коэффициенты к базовым методам и их предсказаниям в виде:

$$G(z) = \arg\max_{c\epsilon\Omega} \sum_{i=1}^{N} w_{ic} \cdot \left[F^{(i)}(z) = c\right],$$

$$w_{ic} = \frac{\#\{x_j | F^{(i)}(x_j) = c_j \wedge c = c_j\}_{j=1}^{M}}{\sum_{c'\epsilon\Omega}^{N} \#\{x_j | F^{(i)}(x_j) = c_j \wedge c\prime = c_j\}_{j=1}^{M}},$$

где весовые коэффициенты вычисляются с учетом того, что их сумма равна 1 для каждого базового классификатора. Метод мягкого голосования присваивает каждому базовому классификатору вектор весов, который можно интерпретировать в виде вероятности правильного определения класса элемента.

Adaboost (общий аналог всех бустинг-методов и в частности XGB) представляется следующим образом:

$$G(X) = \arg\max_{c\epsilon S} \sum_{j=1}^{M} w_j \cdot I\left[F_j(X) = c\right],$$

$$w_j = \frac{1}{2} \cdot \ln\left(\left(\sum_{i=1}^{M} \nu_i \cdot I\left(F_j(X_i) \neq c_i\right)\right)^{-1} - 1\right).$$

Adaboost выполняет последовательное обучение классификаторов $\{F^1, \ldots, F^5\}$. Adaboost запоминает историю ошибок классификатора F^i, что позволяет настраивать новый классификатор F^{i+1} с большим акцентом, выраженным через коэффициент ν_i, на распознавание тех объектов, которые ошибочно классифицировались классификатором F^i.

Методы машинного обучения считаются одними из наиболее эффективных для обработки больших данных, поскольку используемые в методах машинного

обучения модели изначально разработаны для учета трех основных признаков больших данных: объема, скорости и разнородности. Задачей, решаемой в машинном обучении, является поиск закономерностей в эмпирических данных с последующим прогнозированием их развития, что напрямую соотносится с предсказанием возникновения инсайдерских угроз по состоянию и истории изменения сетевой активности в КС.

Псевдокод работы комплекса алгоритмов. Вычисление результатов работы комплекса алгоритмов для разных вариаций работает согласно следующим этапам.

Этап 1. В каждый алгоритм `ALGORITHM()` последовательно передается каждая из сессий пользователей (с множеством связанных параметров, хранящихся в модели представления больших данных); число всех сессий обозначим как `SESSIONS_NUM`, а массив самих сессий – как `SESSIONS[]`. Псевдокод этого представлен ниже.

```
// Начало цикла: Перебор индекса сессии от 0 до числа сессий
FOR index = 0 ... SESSIONS_NUM
  // Получение текущей обрабатываемой сессии из списка сессий по индексу
  SESSION = SESSIONS[index]
  // Вызов алгоритма обнаружения инсайдера для сессии SESSION
  // ALG_PARAM – параметр работы алгоритма
  ALGORITHM(SESSION, ALG_PARAM)
// Конец цикла
ROF
```

Каждый из алгоритмов может вызываться с параметром; так, алгоритм на базе экспертных правил не имеет параметров, а алгоритм на базе методов машинного обучения в качестве параметра принимает тип классификатора: DT, NB, k-NN, SVM, PV, WV, SV, Adaboost. Таким образом, `ALGORITHM()` отработает 8 раз – по числу классификаторов.

Этап 2. Каждый из алгоритмов помечает каждую сессию как относящуюся к законному пользователю (пусть, значением `FALSE`) или инсайдеру (пусть, значением `TRUE`); пометка заносится в собственный отдельный массив

INSIDERS_BY_ALGORITHM[] размером SESSIONS_NUM, хранящий принадлежность сессии инсайдеру (естественно, с точки зрения данного алгоритма). Псевдокод этого представлен ниже.

```
// Начало цикла: Перебор индекса сессии от 0 до числа сессий
FOR index = 0 ... SESSIONS_NUM
  // Получение текущей обрабатываемой сессии из списка сессий по индексу
  SESSION = SESSIONS[index]
  // Вызов алгоритма обнаружения инсайдера для сессии SESSION
  // ALG_PARAM – параметр работы алгоритма
  // Алгоритм возвращает FALSE для законного пользователя
  // TRUE для инсайдера
  // Результат заносится в массив отметок об инсайдерах
  INSIDERS_BY_ALGORITHM[index] = ALGORITHM(SESSION, ALG_PARAM)
// Конец цикла
ROF
```

В случае алгоритма на основе методов машинного обучения, будет заполнено 8 массивов с принадлежностью сессий к инсайдеру – по числу классификаторов.

Этап 3. К результатам работы алгоритмов на основе экспертных правил и методов машинного обучения – то есть к массивам INSIDERS_BY_ALG[] – применяются соответствующие комбинации, что означает применение к элементам двух массивов логический операций ИЛИ (для объединения), И (для пересечения), а также взятие элемента только из первого массива (для выбора первого) или только из второго массива (для выбора второго). Результат считается итогом работы комплекса алгоритмов согласно заданной вариации и заносится в массив INSIDER_BY_COMPLEX[], также имеющий размер SESSIONS_NUM.

Такие операции могут быть записаны на псевдокоде следующим образом (в коде далее префикс и постфикс переменных ALGORITHM заменен на аббревиатуру алгоритмов – RB_ALG и ML_ALG):

Вариант $I_{RB} \vee I_{ML}$: Объединение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения

```
// Начало цикла: Перебор индекса сессии от 0 до числа сессий
```

```
FOR index = 0 ... SESSIONS_NUM
  // Получение отметки об инсайдере согласно работе 1-го алгоритма
  INSIDER_BY_RB_ALG = INSIDERS_BY_RB[index]
  // Получение отметки об инсайдере согласно работе 2-го алгоритма
  INSIDER_BY_ML_ALG = INSIDERS_BY_ML[index]
  // Поэлементное объединение результатов работы алгоритмов
  INSIDERS_BY_COMPLEX[index] = INSIDER_BY_RB_ALG OR INSIDER_BY_ML_ALG
// Конец цикла
ROF
```

Вариант $I_{RB} \wedge I_{ML}$: Пересечение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения

```
// Начало цикла: Перебор индекса сессии от 0 до числа сессий
FOR index = 0 ... SESSIONS_NUM
  // Получение отметки об инсайдере согласно работе 1-го алгоритма
  INSIDER_BY_RB_ALG = INSIDERS_BY_RB[index]
  // Получение отметки об инсайдере согласно работе 2-го алгоритма
  INSIDER_BY_ML_ALG = INSIDERS_BY_ML[index]
  // Поэлементное пересечение результатов работы алгоритмов
  INSIDERS_BY_COMPLEX[index] = INSIDER_BY_RB_ALG AND INSIDER_BY_ML_ALG
// Конец цикла
ROF
```

Вариант I_{RB}: Выбор результатов работы алгоритма на основе экспертных правил

```
// Начало цикла: Перебор индекса сессии от 0 до числа сессий
FOR index = 0 ... SESSIONS_NUM
  // Получение отметки об инсайдере согласно работе 1-го алгоритма
  INSIDER_BY_RB_ALG = INSIDERS_BY_RB[index]
  // Выбор результатов работы 1-го алгоритмов
  INSIDERS_BY_COMPLEX[index] = INSIDER_BY_RB_ALG
// Конец цикла
ROF
```

Вариант I_{ML}: Выбор результатов работы алгоритма на основе методов машинного обучения

```
// Начало цикла: Перебор индекса сессии от 0 до числа сессий
FOR index = 0 ... SESSIONS_NUM
  // Получение отметки об инсайдере согласно работе 2-го алгоритма
  INSIDER_BY_RB_ALG = INSIDERS_BY_RB[index]
  // Выбор результатов работы 2-го алгоритмов
  INSIDERS_BY_COMPLEX[index] = INSIDER_BY_ML_ALG
// Конец цикла
ROF
```

Этап 4. Итоговый массив отметок об инсайдерах преобразуется к списку индексов инсайдерских сессий (`INSIDERS_BY_COMPLEX[index]`) путем выбора тех сессий, для которых стоит итоговая отметка, как принадлежащий инсайдеру; псевдокод может быть следующим:

```
// Начало цикла: Перебор индекса сессии от 0 до числа сессий
FOR index = 0 ... SESSIONS_NUM
  // Проверка принадлежности сессии по индексу к инсайдерской
  IF INSIDERS_BY_COMPLEX[index] == TRUE
    // Добавление индекса инсайдерской сессии в массив INSIDERS_INDEX[]
    INSIDERS_INDEX[] << index
  // Конец условия
  IF
// Конец цикла
ROF
```

Таким образом, в конце работы всех комбинаций результатов работы алгоритмов будет создано 25 массивов `INSIDERS_INDEX[]`, каждый из которых будет содержать список индексов сессий, информация о которых хранится в массиве `SESSIONS[]`. По индексу сессии можно получить саму сессию, по которой согласно модели вычисляется ID инсайдера.

Для выбора наилучшей комбинации, как указывалось ранее, необходима оценка качества работы алгоритмов с помощью соответствующих мер – путем сопоставления полученного множества инсайдеров исходным истинным.

Комбинации результатов алгоритмов комплекса. Согласно формальному определению комплекса, приведем возможные комбинации результатов работы его

алгоритмов и их условные обозначения (I_{RB} в данном случае будет означать результат работы алгоритма на основе экспертных правил, а $I_{ML}(C_{ML})$ – результат работы алгоритма на базе одного из методов машинного обучения согласно заданному классификатору C_{ML}):

1) I_{RB} – применение только алгоритма на основе экспертных правил;

2) $I_{ML}(DT)$ – применение только алгоритма на основе методов машинного обучения для классификатора DT;

3) $I_{ML}(NB)$ – применение только алгоритма на основе методов машинного обучения для классификатора NB;

4) I_{ML}(k-NN) – применение только алгоритма на основе методов машинного обучения для классификатора k-NN;

5) I_{ML}(SVM) – применение только алгоритма на основе методов машинного обучения для классификатора SVM;

6) $I_{ML}(PV)$ – применение только алгоритма на основе методов машинного обучения для классификатора PV;

7) $I_{ML}(WV)$ – применение только алгоритма на основе методов машинного обучения для классификатора WV;

8) $I_{ML}(SV)$ – применение только алгоритма на основе методов машинного обучения для классификатора SV;

9) I_{ML}(Adaboost) – применение только алгоритма на основе методов машинного обучения для классификатора Adaboost;

10) $I_{RB} \vee I_{ML}$(DT) – объединение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора DT;

11) $I_{RB} \vee I_{ML}$(NB) – объединение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора NB;

12) $I_{RB} \vee I_{ML}(\text{k-NN})$ – объединение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора k-NN;

13) $I_{RB} \vee I_{ML}(\text{SVM})$ – объединение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора SVM;

14) $I_{RB} \vee I_{ML}(\text{PV})$ – объединение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора PV;

15) $I_{RB} \vee I_{ML}(\text{WV})$ – объединение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора WV;

16) $I_{RB} \vee I_{ML}(\text{SV})$ – объединение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора SV;

17) $I_{RB} \vee I_{ML}(\text{Adaboost})$ – объединение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора Adaboost;

18) $I_{RB} \wedge I_{ML}(\text{DT})$ – пересечение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора;

19) $I_{RB} \wedge I_{ML}(\text{NB})$ – пересечение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора NB;

20) $I_{RB} \wedge I_{ML}(\text{k-NN})$ – пересечение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора k-NN;

21) $I_{RB} \wedge I_{ML}(\text{SVM})$ – пересечение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора SVM;

22) $I_{RB} \wedge I_{ML}(\text{PV})$ – пересечение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора PV;

23) $I_{RB} \wedge I_{ML}(\text{WV})$ – пересечение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора WV;

24) $I_{RB} \wedge I_{ML}(\text{SV})$ – пересечение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора SV;

25) $I_{RB} \wedge I_{ML}(\text{Adaboost})$ – пересечение результатов работы алгоритмов на основе экспертных правил и методов машинного обучения с использованием классификатора Adaboost.

Описанная схема работы комплекса алгоритмов приведена на рисунке 2.7.

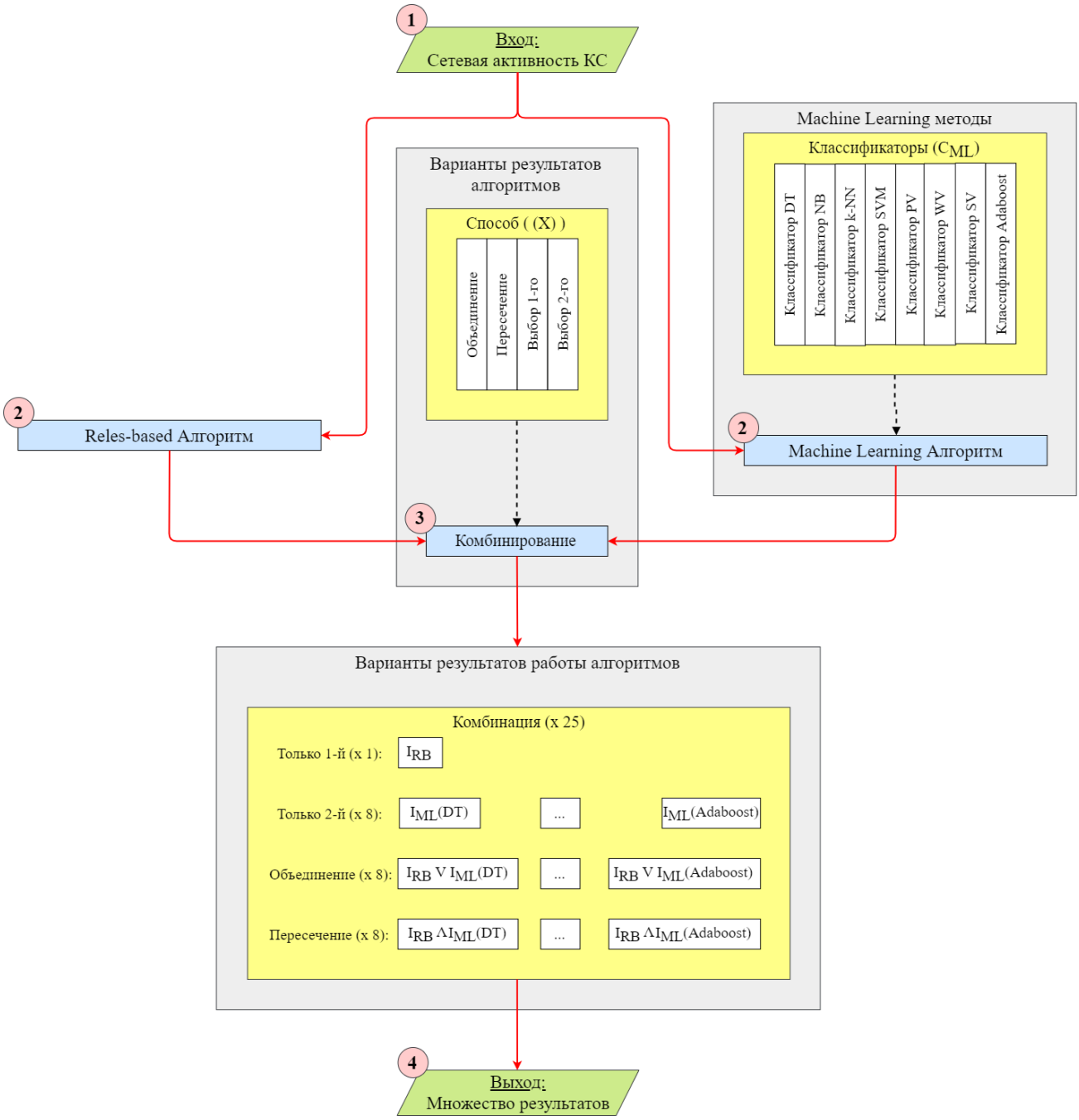

Рисунок 2.7 – Схема работы комплекса алгоритмов обнаружения инсайдеров

Согласно схеме, изображенной на рисунке 2.7, комплекс алгоритмов состоит из 4-х этапов:

Этап 1 – Ввод данных о сетевой активности КС, которые затем передаются на вход алгоритмов на основе экспертных правил и методов машинного обучения.

Этап 2 – Параллельный запуск алгоритма на основе экспертных правил и вариаций алгоритма на базе методов машинного обучения для каждого из классификаторов (1-я вариация).

Этап 3 – Комбинирование результатов обоих алгоритмов путем различных способов (2-я вариация).

Этап 4 – Вывод множества результатов работы алгоритмов по каждой из вариаций.

Схема вычислений комплекса алгоритмов. Обобщенная схема вычислений, производимых системой обнаружения инсайдеров и основанных на возможных вариациях комплекса ее алгоритмов, показана на рисунке 2.8. На схеме также учтено обучение второго алгоритма из комплекса, результаты которого будут использоваться при проведении эксперимента. На рисунке 2.8 пунктирной красной линией показан пример выбора классификатора для машинного обучения и комбинации результатов работы алгоритмов комплекса, имеющих наилучшие показатели качества работы и полученные экспериментальной оценкой.

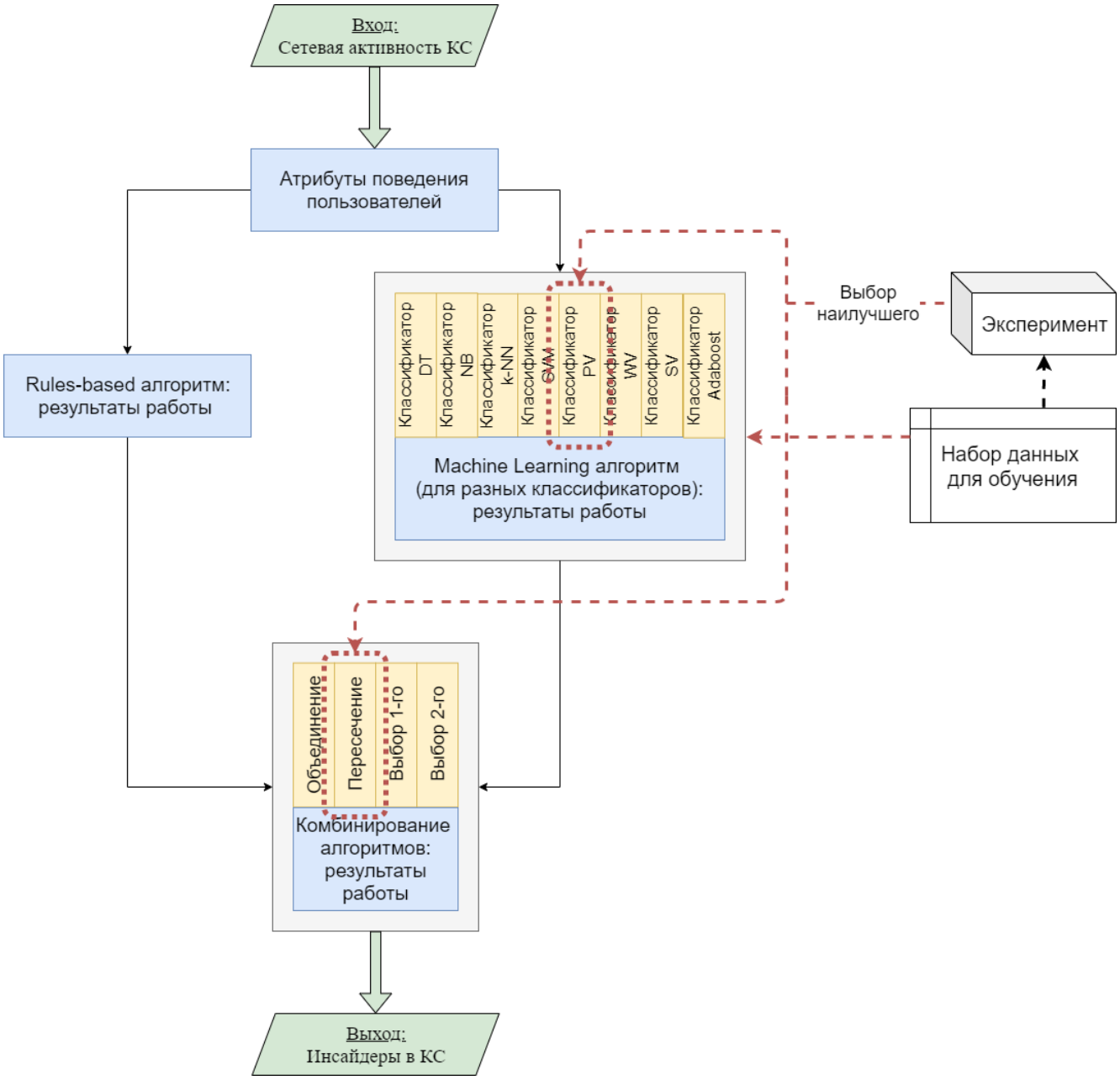

Рисунок 2.8 – Схема вычислений результата работы комплекса алгоритмов обнаружения инсайдеров

Таким образом, информационная схема работы комплекса алгоритмов основана на использовании конкретных алгоритмов вычисления и способов комбинирования результатов, обоснованный выбор которых будет получен экспериментально.

Корректная работа алгоритма на базе методов машинного обучения основана на обучении соответствующими наборами данных. Для задачи диссертационного

исследования наиболее приемлемым решением является обучение по прецедентам, заключающееся в предъявлении интеллектуальной системе набора положительных и отрицательных примеров; в качестве первых может быть выбран сетевой трафик законных пользователей, а в качестве вторых – инсайдерский трафик.

Типовые сценарии атак инсайдера. В качестве типовых сценариев инсайдерских атак были выбраны следующие семь, представляющих по мнению экспертов наибольшую распространенность среди типовых атак в КС; также указаны критерии определения атак:

1. Сканирование внутренних MAC/IP-адресов и TCP/UDP портов (например, отправка запросов на соединение всех портов подряд) – по большому количеству эхо-запросов ко всем устройствам сети и/или их портам;

2. Отказ в обслуживании с помощью «SYN-флуд» (отправка большого количества SYN-запросов сервису за короткий период) – по подсчету количества и времени отправленных SYN-запросов;

3. Неудачная попытка входа в сервис через HTTP/FTP (например, более 5 отказов аутентификации в течение 1 минуты) – по ответам сервера AAA;

4. Аномальное получение чрезмерно большого количества данных с внутренних ресурсов организации (например, получение в течение одного дня 10Гб трафика, если за предыдущие 6 месяцев получалось 100Мб данных в день) – по резкому повышению количества отправленных данных по сравнению со среднестатистическим за период;

5. Аномальная отправка чрезмерно большого количества данных вне периметра организации (например, отправка в течение одного дня 10Гб трафика, если за предыдущие 6 месяцев отправлялось 100Мб данных в день) – по резкому повышению количества полученных данных по сравнению со среднестатистическим за период;

6. Выполнение шести этапов: (1) ожидание нерабочего времени (например, когда отсутствуют другие сотрудники); (2) подключение к сети нового устройства (например, собственного ноутбука) по гостевому WiFi; (3) вход в систему аутентификации с первой попытки по чужой учетной записи (то есть, легального

пользователя); (4) скачивание умеренно большого объема информации на подключенное устройство с внутренних ресурсов сети (например, получение в течение одного часа 2 Гб трафика, если за предыдущий период в течение 6 месяцев получалось по 1 Гб данных в день); (5) отсутствие активности в течение некоторого времени (например, в течение одного часа); (6) отключение устройства от сети – по целому ряду признаков о сетевой активности, работе сервера аутентификации и т.п., а также их закономерностям;

7. Выполнение шести этапов: (1) ожидание рабочего времени (то есть выполнение сотрудником своих должностных обязанностей); (2) вход в систему аутентификации с первой попытки (то есть, использование учетной записи легального пользователя); (3) запуск на собственном компьютере программы поиска и получения персональных данных в сети организации; (4) скачивание умеренно большого объема информации на компьютер сети с внутренних ресурсов; (5) создание зашифрованного канала для передачи данных вне периметра организации; (6) отправка скаченных данных по созданному каналу на внешний IP адрес, который относится к разряду подозрительных (например, расположенный в другой стране или принадлежащий конкурирующей организации).

И если 1-й, 2-й и 3-й сценарии можно считать *простыми*, то 4-й и 5-й сценарии являются *сложными* с той точки зрения, что требуют учета динамических характеристик поведения инсайдеров, а именно аномалий в их поведении за некоторый период времени. Это реализуется в модели хранением данных о сетевой активности пользователя за некоторый промежуток времени, например, в виде массива дневных активностей пользователей за предыдущие месяцы.

6-й сценарий является комплексным, поскольку состоит из шести действий инсайдера, для обнаружения которых, очевидно, потребуется использование достаточно сложных алгоритмических решений. Суть сценария заключается в том, что инсайдер подключает к сети свой ноутбук, выбрав для этого время, когда другие пользователи отсутствуют на рабочих местах. Затем инсайдер осуществляет обдуманное скачивание интересующей его информации на ноутбук с внутренних

ресурсов. После чего производит предварительный анализ информации, отключает ноутбук от сети и уходит.

Последний (7-й) сценарий также относится к комплексным сценариям, которые состоят из нескольких этапов. Сценарий описывает поведение инсайдера, который в свое рабочее время старается скрытно собрать интересующую его информацию и отправить ее вне периметра организации, обходя при этом возможные DLP-системы путем шифрования трафика.

Список типовых сценариев атак может быть расширен другими без качественных изменений в схеме эксперимента.

Непосредственная настройка алгоритма путем его обучения была произведена в процессе эксперимента, описание которого будет дано в подразделе 3.3 (этапы 1 и 2 эксперимента).

2.4 Выводы по главе 2

1. Представлен подход к построению системы анализа поведения пользователей в компьютерной сети организации.
2. Создана модель представления больших данных об инсайдерских атаках в формате NoSQL, включающая модель инсайдера.
3. Разработаны экспертные правила с применением многокритериальной оценки и балльного начисления для обнаружения инсайдеров в КС.
4. Разработан алгоритм обнаружения инсайдеров в КС на основе разработанных экспертных правил.
5. Обоснован выбор классификаторов для работы алгоритма на основе методов машинного обучения, включая их комбинацию.
6. Разработана модель и комплекс алгоритмов обнаружения инсайдеров в компьютерной сети на основе экспертных правил и методов машинного обучения.

7. Предложена схема вычислений результата работы комплекса алгоритмов обнаружения инсайдеров.

8. Описаны типовые сценарии инсайдерских атак, состоящие из трех простых, двух сложных и двух комплексных, которые могут быть использованы для тестирования алгоритмов обнаружения инсайдеров в КС.

Глава 3. Методика, архитектура и программная реализация системы обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных и их экспериментальная оценка

3.1 Методика обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных

Предложенная методика обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных определяет основные этапы и шаги использования разработанных моделей и алгоритмов.

Методика обнаружения инсайдеров в КС основывается на процессе сбора данных об ассоциированной с пользователями сетевой активности, распараллеленной предварительной обработке данных с последующим применением алгоритмов на основе экспертных правил и методов машинного обучения, а также окончательном выводе результатов анализа в человеко-ориентированном виде. Методика основана на трех следующих этапах:

Этап 1. Сбор информации и настройка параметров.

Этап 2. Применение алгоритмов обнаружения инсайдеров.

Этап 3. Анализ выходных данных.

Каждый элемент методики ассоциирован с действиями одного из ее участников: оператора или системы обнаружения; также указаны соответствия действий системы с реализующими их компонентами. Схема методики представлена на рисунке 3.1.

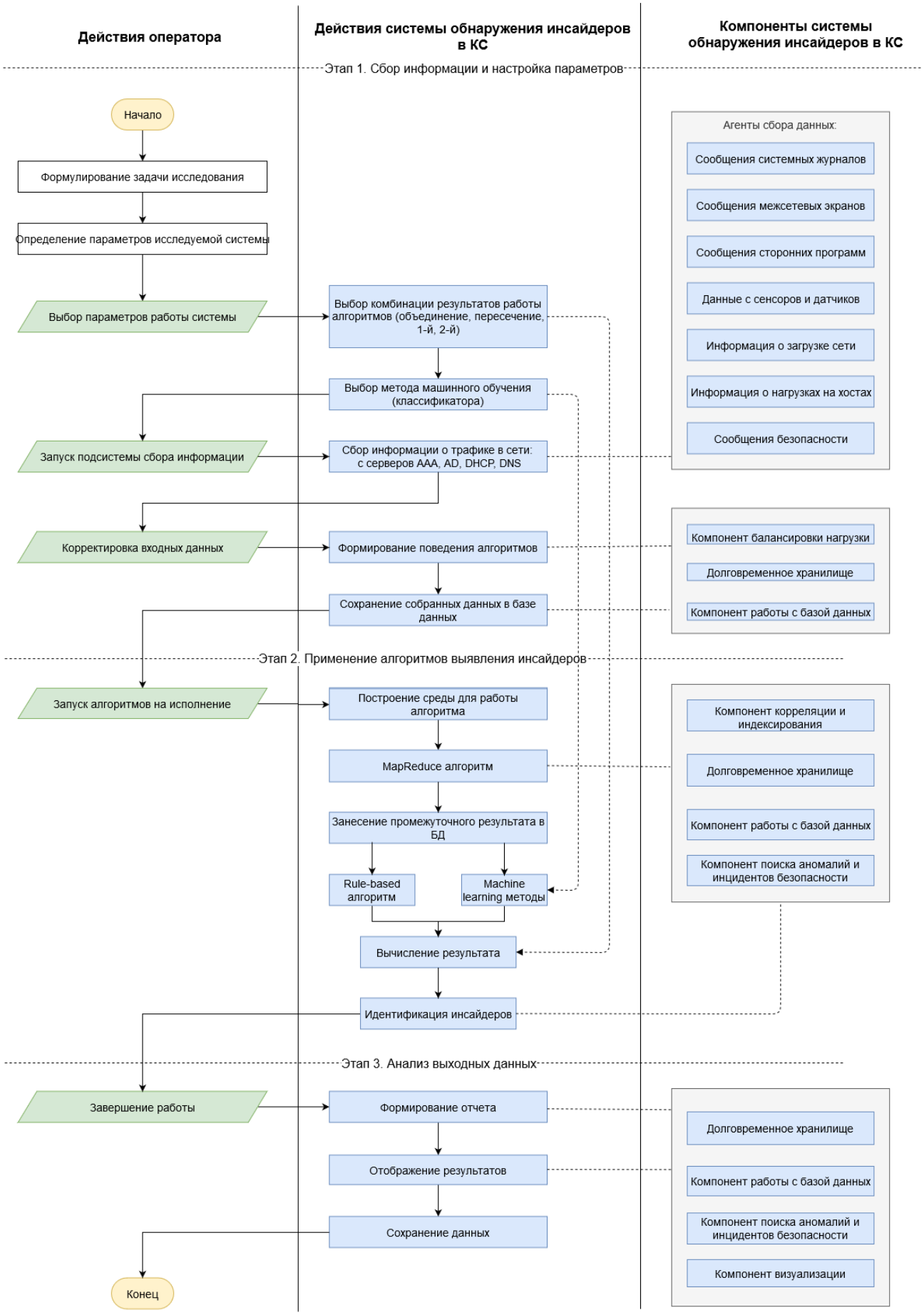

Рисунок 3.1 – Схема методики обнаружения инсайдеров в КС

Опишем этапы методики более подробно.

Этап 1. Сбор информации и настройка параметров

На данном этапе происходит сбор информации о сетевом трафике КС в подходящем для анализа виде с последующей настройкой параметров системы. Этап может быть поделен на следующие подэтапы:

– формулировка задачи исследования КС и определение параметров, по которым она будет исследоваться;

– выбор параметров работы системы, позволяющий оператору указать классификатор для работы метода машинного обучения, выбрав соответствующий метод из набора реализованных, а также комбинацию результатов работы алгоритмов на основе экспертных правил и методов машинного обучения;

– запуск процесса сбора информации системой обнаружения с серверов AAA, AD, DHCP и DNS; для сбора используются соответствующие агенты системы обнаружения, собирающие сообщения системных журналов, межсетевых экранов и сторонних программ, получающие данные с сенсоров и датчиков, накапливающие информацию о загрузке сети и нагрузке хостов, а также регистрирующие сообщения о нарушениях информационной безопасности;

– корректировка данных оператором, позволяющая внести некоторые корректировки в собранные данные для увеличения точности работы алгоритмов обнаружения;

– итоговое формирование поведения алгоритмов обнаружения инсайдеров (с помощью компонентов системы обнаружения, обеспечивающих балансировку нагрузки), а также сохранение информации в базе данных (с помощью соответствующих компонентов работы с БД).

Этап 2. Применение алгоритмов обнаружения инсайдеров

На данном этапе происходит обнаружение инсайдеров с помощью разработанного комплекса алгоритмов, реализованных в системе обнаружения. В

качестве исходных данных используются данные, собранные на Этапе 1. Этап 2 может быть поделен на следующие подэтапы:

– построение среды для работы алгоритмов после ручного запуска процесса оператором;

– выполнение MapReduce для распараллеленной обработки на нескольких компьютерах с целью предобработки данных (подсчет количества сессий, IP-пакетов, хостов и т.п.);

– занесение результатов обработки MapReduce в базу данных для оперативной работы с ними алгоритмов обнаружения инсайдеров;

– параллельный запуск алгоритмов из комплекса: на основе экспертных правил (Rule-based) и методов машинного обучения (Machine learning);

– сравнение результатов работы алгоритмов (объединение, пересечение или выбор одного) для определения вариации с наилучшими показателями качества, по которому и производится идентификация инсайдеров;

– на всем протяжении работы этапа 2 для работы с MapReduce используются компоненты корреляции и индексирования, для сохранения результатов – компоненты работы с базой данных.

Этап 3. Анализ выходных данных

На данном этапе происходит непосредственный вывод результатов оператору, который был получен на предыдущем этапе. Этап может быть поделен на следующие выполняемые подэтапы:

– формирование отчета, систематизирующего и классифицирующего найденных инсайдеров, включая пометки к сессиям, сделанные алгоритмом на основе экспертных правил;

– отображение результатов;

– сохранение данных, позволяющее ознакамливаться с ранее полученными результатами;

– на всем протяжении работы этапа для сохранения результатов используются компоненты работы с базой данных.

Для работы методики необходимо: использование агентов сбора информации о функционировании КС, поддержка базы данных, возможность распределенных вычислений, настроенный метод машинного обучения (на наборах типового и инсайдерского поведения пользователей), а также оболочка для визуализации отчетов.

Таким образом, входными данными методики является информация о функционировании КС, промежуточными данными – заполненная реальными данными модель представления больших данных в формате NoSQL, а выходными данными – отчет об обнаруженных инсайдерах.

Важной особенностью методики является ее настройка под точные требования оператора путем выполнения шага первого этапа «Выбор параметров работы системы», суть которой заключается в следующем – результаты выбора параметров системы на Этапе 1 непосредственно влияют на работу элементов системы на Этапе 2, что показано пунктирными стрелками на рисунке 3.1.

Метод машинного обучения, примененный в методике, может использовать один из 8-ми классификаторов, описанных ранее: DT, NB, k-NN, SVM, PV, WV, SV, Adaboost (первые 4 классификаторы являются базовыми, а последние 4 – их комбинацией). Результаты работы алгоритмов на основе экспертных правил и методов машинного обучения также могут комбинироваться следующим образом: объединение, пересечение, только 1-й, только 2-й. Выбор классификатора и итоговой комбинации результатов влияет на качество выявления инсайдеров, меры которого были определены как *r, p, a, e, f*. Следовательно, возможны следующие различные группы мер качества: 1 – при работе только алгоритма на основе экспертных правил, 8 – при работе только метода машинного обучения (для заданного классификатора), и по 8 – для объединения и пересечения их результатов (1 алгоритм для каждого из 8-ми классификаторов метода). Каждая из 25 групп (1 + 8 + 8 * 2) будет иметь собственные значения мер, которые могут иметь различную значимость для оператора. И хотя классической, используемой также в

диссертационной работе, является F-мера (*f*), тем не менее, для ряда других случаев остальные меры могут иметь более приоритетное значение.

Каждая из мер может быть выбрана в качестве определяющей для своей задачи: *полнота* – важным является то, сколько из проверенных пользователей мы определили, как инсайдеров; *точность* – важным является то, сколько из обнаруженных нами инсайдеров реально являются таковыми; *аккуратность* – важным является то, какова доля правильных предсказаний инсайдеров и законных пользователей среди всех, *ошибка* – когда важным является то, какова доля неправильных предсказаний инсайдеров и законных пользователей среди всех рассмотренных. Естественно, поскольку данные *относительные* меры (имеющие смысл долей или вероятностей) вычисляются на основании *абсолютных* – TP, TN, FP и FN, – последние также могут быть использованы в качестве критерия важности; однако, работа с абсолютными значениями не всегда оказывается целесообразной. В эксперименте по оценке обоснованности разработанной системы обнаружения инсайдеров в КС для сравнения с аналогами в качестве меры выбрана именно F-мера, что является вполне оправданным решением.

Рассмотрим применение разработанных моделей и алгоритмов, описанных ранее более подробно в методике обнаружения инсайдеров в КС.

На этапе сбора информации:

– Собранные данные о поведении пользователей помещаются в модель представления больших данных об инсайдерских атаках для последующей обработки;

– Оператор производит корректировку входных данных, которые берутся из модели представления больших данных об инсайдерских атаках;

– Скорректированные оператором данные сохраняются в базе данных;

На этапе применения алгоритмов выявления инсайдеров:

– Исходные данные для обработки с помощью алгоритма распределенных вычислений MapReduce берутся из модели базы данных;

– Результаты обработки алгоритмом распределенных вычислений MapReduce сохраняются в базе данных;

– Все исходные данные, предоставляемые алгоритмам обнаружения инсайдеров в КС, берутся из базы данных;

– Обработка собранных данных о поведении пользователя осуществляется алгоритмом на основе экспертных правил;

– Обработка собранных данных о поведении пользователя осуществляется комплексом алгоритмов на основе экспертных правил и методов машинного обучения;

– Результаты совместной обработки собранных данных, а также идентифицированные инсайдеры сохраняются в базе данных;

На этапе анализа выходных данных:

– Формирование отчета для оператора производится на основании данных об инсайдерской деятельности, занесенных в базу данных алгоритмами обнаружения;

– Отображение результатов делается на основании данных, хранимых в базе данных;

– Финальные данные по отчетам и другая статистическая информация (время проведения методики, корректировки оператора и т.п.) сохраняются в базе данных.

Одним из важнейших элементов методики является применение модели MapReduce, предназначенной для создания фреймворка для параллельных вычислений над большими данными. Также, методика включает в себя предложенные алгоритмы и модели, ориентированные на работу с большими данными. Кроме того, на последнем этапе методики при помощи графов происходит визуализация большого количества аналитических данных – записей из модели представления больших данных об инсайдерских атаках, – что входит в реализацию используемой для этого базы данных OrientDB и может быть отнесено к методам и техникам анализа больших данных.

3.2 Архитектура и программная реализация системы обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных

Архитектура системы. Архитектура системы обнаружения инсайдеров, изображенная на рисунке 3.2, включает три уровня обработки информации: (1) уровень сети и данных (данные от источников, специальные агенты сбора и отправки данных); (2) уровень предварительной обработки информации и событий безопасности (компонент балансировки и разделения нагрузки, компоненты корреляции и индексирования (MapReduce), хранилище (NoSQL база данных), компонент работы с базой данных, компонент предварительного анализа информации и событий безопасности); (3) уровень аналитической обработки информации и событий безопасности (компонент поиска аномалий и инцидентов безопасности, компонент визуализации).

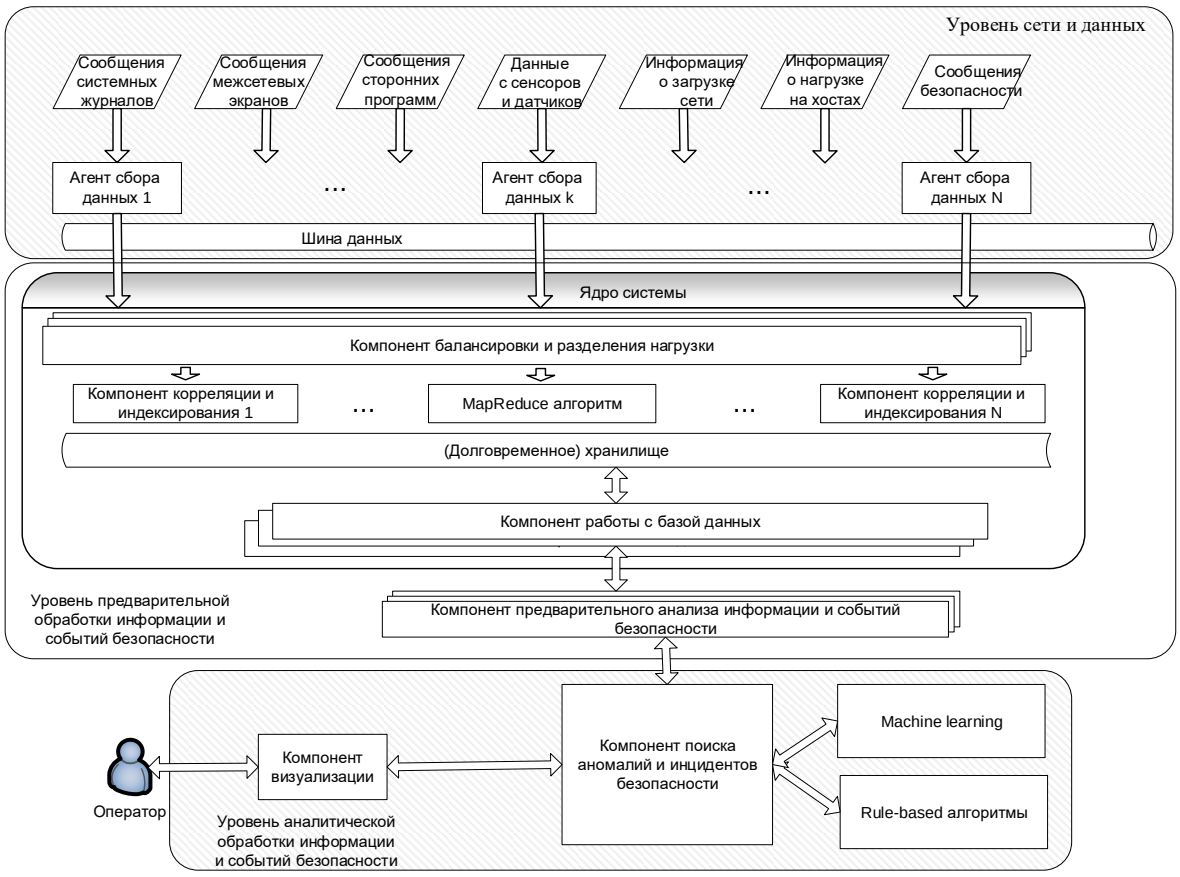

Рисунок 3.2 – Архитектура системы обнаружения инсайдеров на основе обработки сетевого трафика

Приведем описание компонентов каждого из уровней и их взаимодействия. Уровень сети и данных содержит компоненты для сбора сообщений и другой информации от множества источников. К первым относятся следующие: (1) сообщения из системных журналов, включающие данные о времени создания сообщений и их важности; (2) сообщения из системных журналов межсетевых экранов, включающие данные о TCP- и UDP-сессиях (номер сессии, флаги TCP-сообщений, время открытия сессии, время завершения сессии); (3) данные от сторонних программ, например, сведения о потенциально опасных приложениях от антивирусных программ. Ко вторым относятся: (4) данные с сенсоров и датчиков – количество подключенных устройств, количество используемых портов, а также (5) информация о загрузке сети – данные о количестве переданного и полученного трафика, статистика по используемым приложениям; (6) данные о нагрузке на

хостах, включая сведения о загрузке процессора, количестве запущенных приложений, загрузке жесткого диска, загрузке оперативной памяти.

С каждым сборщиком информации связан отдельный компонент уровня – агент сбора данных, преобразующий информацию в единый вид. Так, например, агент для сообщений системных журналов осуществляет преобразование сообщений из формата Syslog в формат csv, агент для данных с сенсоров и датчиков – обеспечивает раздельный сбор полученной информации по следующим категориям: количество подключенных устройств, количество используемых портов и т.д.

Полученная информация отправляется на отдельный компонент уровня – шину данных, предназначенную для распространения данных о событиях безопасности и их гарантированной доставки. Из-за большого количества входных данных возможны перегрузки системы, для предотвращения чего, в частности, предназначен следующий уровень.

Первым на уровне предварительной обработки информации и событий безопасности расположен компонент балансировки и разделения нагрузки, который обеспечивает равномерную обработку информации, собранной датчиками на предыдущем уровне. Данный компонент обеспечивает распределение информации между остальными компонентами: несколько компонентов корреляции и индексирования, включая преобразование данных с помощью парадигмы MapReduce. MapReduce обеспечивает работу распределенных вычислений над собираемыми большими данными и состоит из трех шагов: классификации данных, предварительной обработки собранной информации и их свертки. Все это обеспечивает долговременное хранение данных (путем использования соответствующего компонента).

Полученные таким образом данные заносятся в базу данных, обеспеченную соответствующим компонентом уровня. Работа базы данных основана на предложенной модели представления больших данных об инсайдерских атаках. Также, с базой данных работает компонент предварительного анализа данных и событий безопасности, который позволяет собрать необходимую выборку событий

и произвести первичный анализ полученной информации для дальнейшего предоставления компонентам уровня аналитической обработки информации и событий безопасности.

Следующим осуществляет работу уровень аналитической обработки информации и событий безопасности, который можно считать основным с точки зрения обнаружения инсайдеров, поскольку он содержит всю логику работы алгоритмов, построенных на базе экспертных правил и методов машинного обучения. Уровень состоит из четырех основных компонентов: (1) поиска аномалий и инцидентов безопасности, (2) алгоритма на основе экспертных правил, (3) алгоритма на основе методов машинного обучения и (4) компонента визуализации.

Первый компонент обеспечивает согласованную работу алгоритмов в составе комплекса – подготовку, настройку (например, выбор классификатора для метода машинного обучения) и подачу в них данных, а также параллельное вычисление и согласование результатов (например, их объединение).

Второй и третий компоненты обеспечивают работу алгоритмов, входящих в комплекс. С этой точки зрения архитектура может быть расширяемой за счет добавления новых алгоритмов, построенных на базе экспертных правил и машинного обучения, кроме тех, что реализованы на данный момент.

Четвертый компонент реализует интерфейс взаимодействия с пользователем. Так, последний может настроить работу комплекса алгоритмов (выбрать способ согласования результатов и классификатор для машинного обучения согласно полученным мерам качества, например, выбрав их пересечение, а не объединение, а также классификатор Adaboost). В результате работы системы обнаружения для предложенной архитектуры оператор получит отчет, содержащий идентификаторы выявленных инсайдеров, пометки о типе и важности инсайдерской атаки и другую информацию.

Согласно разработанной архитектуре был реализован прототип системы обнаружения инсайдеров в КС, применяемый в соответствующей методике. Для

проверки работоспособности прототипа и методики, а также для оценки их характеристик будет проведен соответствующий эксперимент.

Реализация архитектуры системы обеспечивает работу всех ее компонентов – моделей, алгоритмов, методики, – а, следовательно, построена на утилитах, библиотеках и фреймворках, предназначенных для работы с большими данными. Кроме того, для работы программного прототипа системы используется инструментарий Hadoop и файловая система HDFS, изначально спроектированные для поддержки больших данных.

Программный прототип системы. Для запуска программного прототипа был использован специальный стенд, развернутый на базе СПбГУТ (рисунок 3.3). При межсетевом взаимодействии устройства, находящиеся во внутренней сети университета, взаимодействуют с коммутаторами, которые копируют проходящий через них трафик, генерируемый пользователями в КС СПбГУТ (200Гб) на порты виртуальных машин в среде VMware vSphere (ESXi 6.5). Эти виртуальные машины составляют скоординированный кластер под управлением операционных систем CentOS 7 и с установленными на них продуктами компании Apache Hadoop с реализацией Cloudera – Apache Ambari. Кластер имеет следующие характеристики: 18vCPU 1.2 GHz, 48 GB RAM, 1,2 TB HDD. Пять компьютеров кластера выступают в роли DataNode и отвечают за хранение и обработку данных, один компьютер – NameNode – является менеджером для распределения задач и нагрузок кластера, а также хранит таблицы имен файлов. Кроме того, для выявления инсайдера с помощью алгоритма на основе экспертных правил дополнительно используются данные, поступающие с сервера аутентификации авторизации и учета, который развернут на базе решения Cisco ISE версии 2.0.

В имеющемся трафике содержатся данные сетевой активности пользователей сети университета, среди которых присутствуют особые участники – инсайдеры, то есть пользователи, имеющие своей целью нанести ущерб организации через свою деятельность в сети (согласно предложенным сценариям атак).

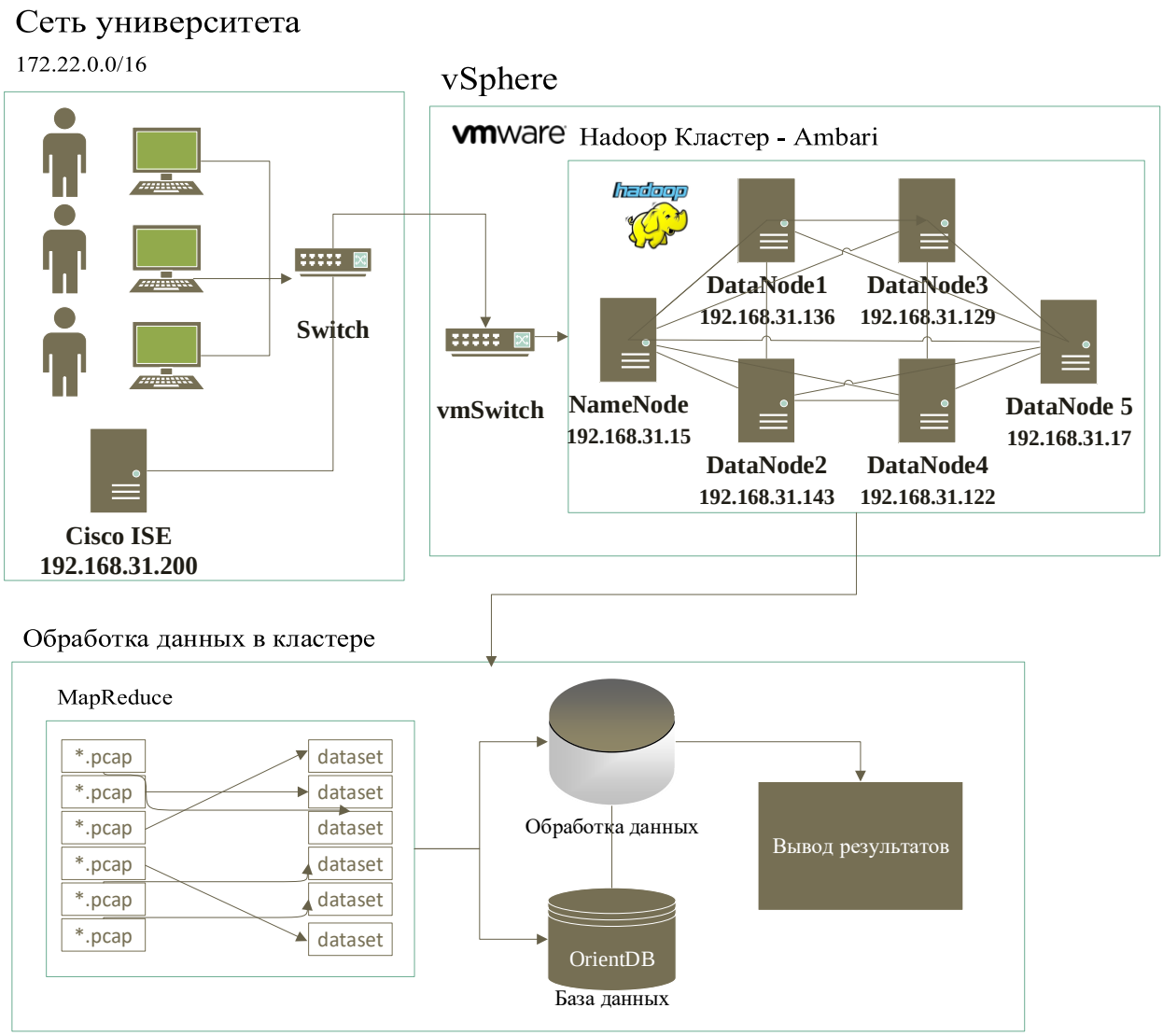

Рисунок 3.3 – Прототип программного-аппаратного комплекса обработки трафика

Пример работы прототипа системы. Попав на кластер, данные подвергаются обработке алгоритмом MapReduce, позволяющим быстро рассчитывать количество вхождений необходимых меток и проводящим сортировку информации в удобный вид. Из базы данных NoSQL OriendDB обработанные данные проходят через алгоритм на основе методов машинного обучения (нормализация, выделение признаков, кластеризация, классификация, оценка эффективности, итоговый вывод) и алгоритма, основанного на экспертных правилах.

В качестве источника данных выступают файлы, генерируемые Wireshark, имеющие расширение «.pcapng». Процесс конвертации состоит из двух этапов.

На первом этапе происходит считывание дампа, хранящегося в файлах .pcapng. Поскольку файлы .pcapng являются бинарными, необходимо для их считывания применять анализатор пакетов tshark, на выходе которого получаются данные, представленные на рисунке 3.4.

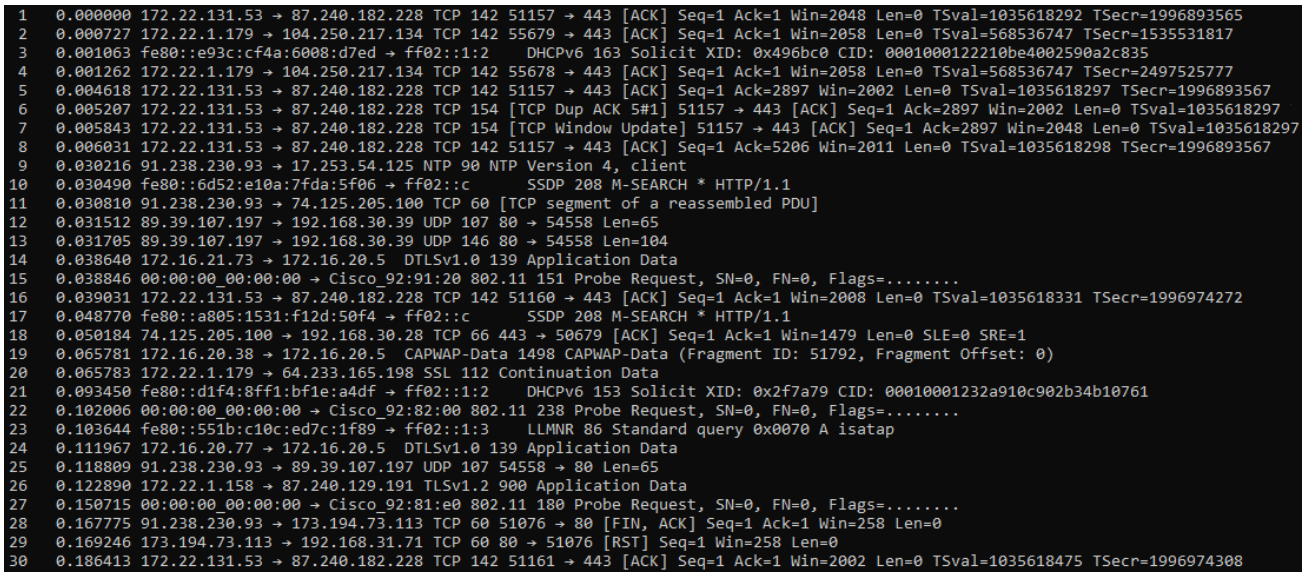

Рисунок 3.4 – Данные на выходе анализатора пакетов tshark

Далее эти данные обрабатываются с помощью программы, написанной на языке программирования python, на выходе которой они принимают вид, представленный на рисунке 3.5. Полученные таким образом данные записываются в csv-файл.

```
1,2019-09-19 11:28:37.927138,TCP,Apple_bb:98:38,172.22.131.53,51157,Cisco_bf:a8:00,87.240.182.228,443
2,2019-09-19 11:28:37.927865,TCP,Apple_02:da:8e,172.22.1.179,55679,Cisco_bf:a8:00,104.250.217.134,443
4,2019-09-19 11:28:37.928400,TCP,Apple_02:da:8e,172.22.1.179,55678,Cisco_bf:a8:00,104.250.217.134,443
5,2019-09-19 11:28:37.931756,TCP,Apple_bb:98:38,172.22.131.53,51157,Cisco_bf:a8:00,87.240.182.228,443
6,2019-09-19 11:28:37.932345,TCP,Apple_bb:98:38,172.22.131.53,51157,Cisco_bf:a8:00,87.240.182.228,443
7,2019-09-19 11:28:37.932981,TCP,Apple_bb:98:38,172.22.131.53,51157,Cisco_bf:a8:00,87.240.182.228,443
8,2019-09-19 11:28:37.933169,TCP,Apple_bb:98:38,172.22.131.53,51157,Cisco_bf:a8:00,87.240.182.228,443
9,2019-09-19 11:28:37.957354,NTP,Cisco_72:4a:98,91.238.230.93,65319,Cisco_cb:d5:3d,17.253.54.125,123
11,2019-09-19 11:28:37.957948,TCP,Cisco_72:4a:98,91.238.230.93,50679,Cisco_cb:d5:3d,74.125.205.100,443
12,2019-09-19 11:28:37.958650,UDP,Cisco_72:4a:98,89.39.107.197,80,IntelCor_b9:c4:45,192.168.30.39,54558
13,2019-09-19 11:28:37.958843,UDP,Cisco_72:4a:98,89.39.107.197,80,IntelCor_b9:c4:45,192.168.30.39,54558
15,2019-09-19 11:28:37.965984,802.11,00:00:00_00:00:00,00:00:00_00:00:00,28652,Cisco_92:91:20,Cisco_92:91:20,5247
16,2019-09-19 11:28:37.966169,TCP,Apple_bb:98:38,172.22.131.53,51160,Cisco_bf:a8:00,87.240.182.228,443
18,2019-09-19 11:28:37.977322,TCP,Cisco_72:4a:98,74.125.205.100,443,HonHaiPr_cf:e0:a7,192.168.30.28,50679
20,2019-09-19 11:28:37.992921,SSL,Apple_02:da:8e,172.22.1.179,55531,Cisco_bf:a8:00,64.233.165.198,443
22,2019-09-19 11:28:38.029144,802.11,00:00:00_00:00:00,00:00:00_00:00:00,28624,Cisco_92:82:00,Cisco_92:82:00,5247
25,2019-09-19 11:28:38.045947,UDP,Cisco_72:4a:98,91.238.230.93,54558,Cisco_cb:d5:3d,89.39.107.197,80
26,2019-09-19 11:28:38.050028,TLSv1.2,TctMobil_eb:bf:d7,172.22.1.158,38675,Cisco_bf:a8:00,87.240.129.191,443
27,2019-09-19 11:28:38.077853,802.11,00:00:00_00:00:00,00:00:00_00:00:00,28625,Cisco_92:81:e0,Cisco_92:81:e0,5247
28,2019-09-19 11:28:38.094913,TCP,Cisco_72:4a:98,91.238.230.93,51076,Cisco_cb:d5:3d,173.194.73.113,80
29,2019-09-19 11:28:38.096384,TCP,Cisco_72:4a:98,173.194.73.113,80,FujitsuT_dc:21:29,192.168.31.71,51076
30,2019-09-19 11:28:38.113551,TCP,Apple_bb:98:38,172.22.131.53,51161,Cisco_bf:a8:00,87.240.182.228,443
```

Рисунок 3.5 – Данные на выходе программы, написанной на языке python

В данном примере приведены значения девяти полей для тридцати пакетов:

– порядковый номер пакета (нумерация начинается для каждого файла pcap, в совокупности с именем файла pcap это позволяет однозначно идентифицировать каждый пакет при занесении его в базу данных);

– время отправки;

– протокол;

– MAC-адрес отправителя;

– IP-адрес отправителя;

– порт отправителя;

– MAC-адрес получателя;

– IP-адрес получателя;

– порт получателя.

На рисунке 3.6 представлено визуальное отображение взаимосвязей между выборкой IP-адресов, включающей в себя 100 IP-адресов отправителей и получателей трафика. Данный рисунок был получен путем визуализации выборки с помощью сервиса Apache Superset, установленного на Apache Ambari. Из рисунка видно, что наибольшее количество IP-соединений – между IP-адресами 192.168.31.44 и 87.240.185.226. Это объясняется тем, что на адресе 192.168.31.44 находится кафедральная инфраструктура. Кроме того, общее количество связей с

IP-адресом 172.16.228.22 превышает среднее количество взаимосвязей между IP-адресами в выборке в 40 раз. Этот факт может свидетельствовать о возможной DDoS атаке.

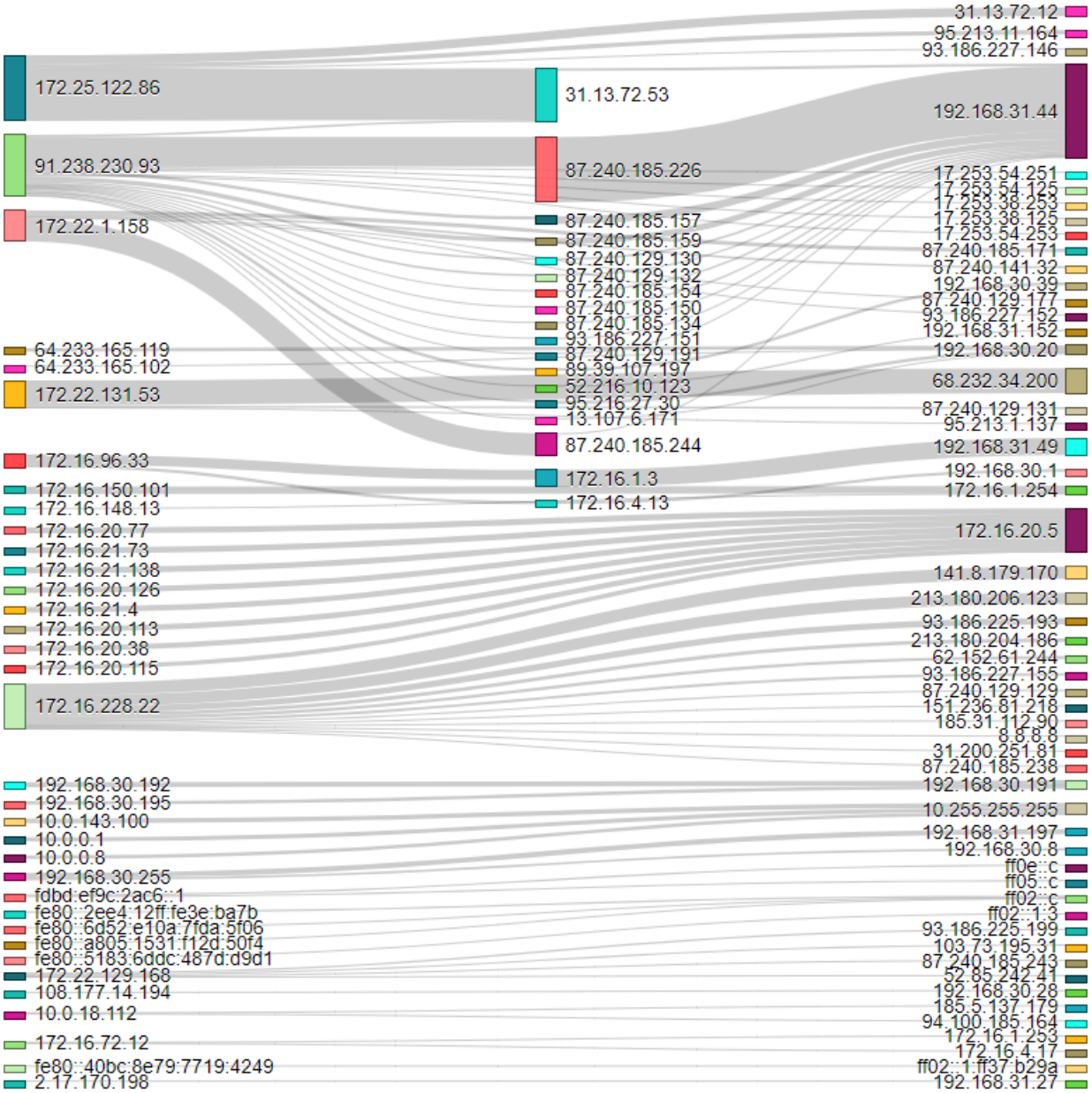

Рисунок 3.6 – Взаимосвязи IP-адресов

На рисунке 3.7 представлено соотношение типов трафика. Из рисунка видно, что TCP трафик занимает самую большую долю – порядка 66%, TLSv1.2 – 12%, 802.11 – 7%, UDP – 6%.

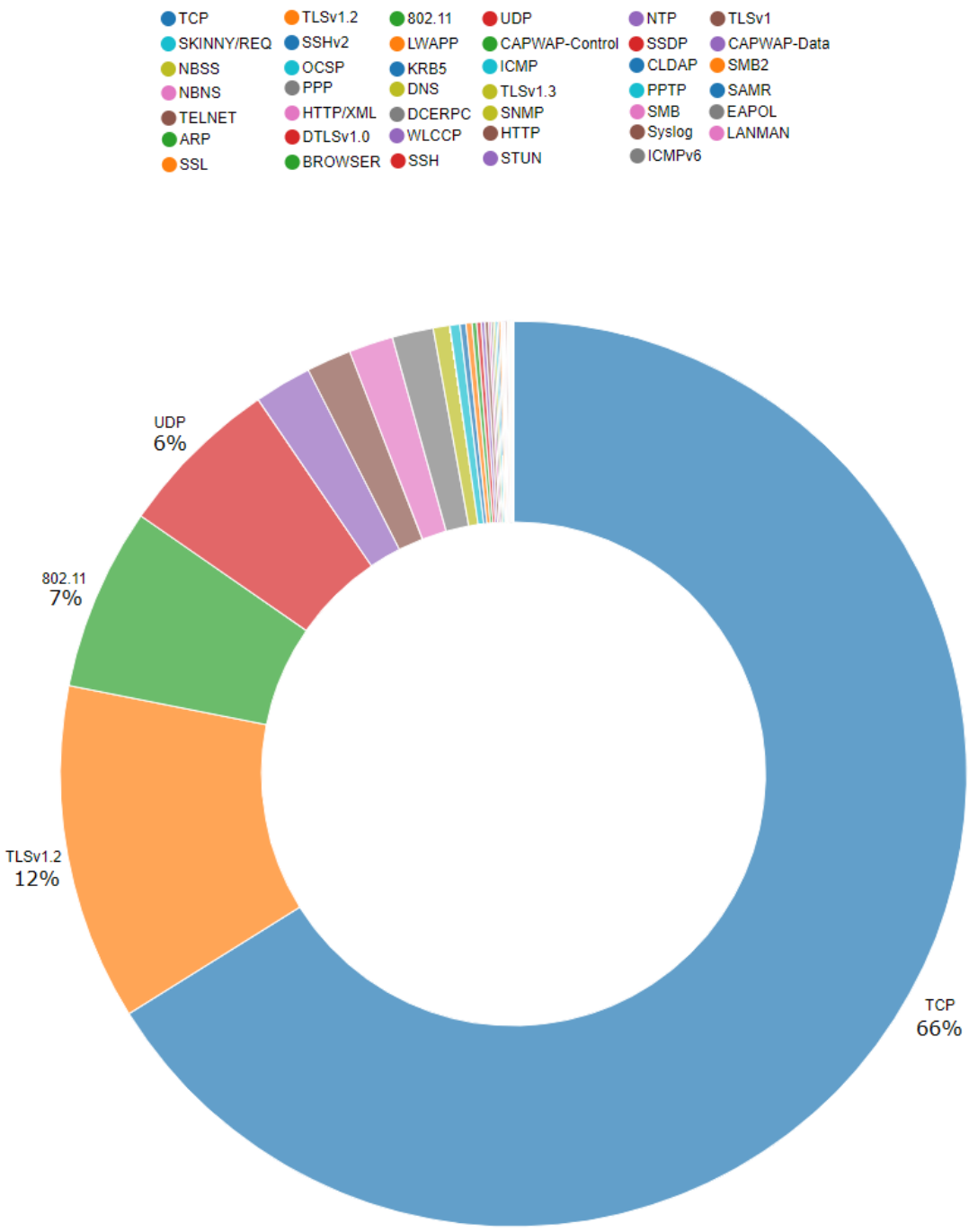

Рисунок 3.7 – Соотношение протоколов в выборке

Алгоритмы на основе экспертных правил и методов машинного обучения реализованы на языке Python и выполняются распределенно на DataNode-узлах кластера.

Визуализация результатов работы прототипа. Приведем примеры работы компонента визуализации для всех рассматриваемых сценариев атак – трех

простых, двух сложных и двух комплексных. Компонент выдает результаты в консоль в виде списка идентификаторов инсайдеров и дополнительной информации к ним. В качестве формата вывода используется формат JSON, обеспечивающий использование графических настроек (рисунок 3.8).

Рисунок 3.8 – Пример результатов работы прототипа обнаружения инсайдеров в КС

На рисунке в консольном выводе каждый ID инсайдера связан со своим сценарием атаки – от 1-го до 7-го. Комментарии добавлены вручную для пояснения результатов обнаружения инсайдеров.

Отдельные элементы прототипа, использующие элементы предложенного подхода, были зарегистрированы в Реестре программ для ЭВМ.

3.3 Экспериментальная оценка разработанной методики и программной реализации системы обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных

Произведем выбор систем, близких к разрабатываемой и сравним их эффективность.

Выбор существующих аналогов систем обнаружения инсайдеров. Понятие эффективности системы или процесса является достаточно многогранным, имеющим различные трактовки и способы вычисления. Достаточно известным, показавшим хорошую применимость на практике определением эффективности, считается совокупность трех ее показателей – своевременность, обоснованность и ресурсопотребление (введенные в п. 1.4 диссертации); для лаконичности применения, удобнее вместо последнего иногда использовать противоположный показатель – ресурсоемкость, определяя таким образом общее повышение эффективности как повышение всех или ряда ее показателей. Особенность такого определения заключается в том, что повышение одного из параметров (без дополнительных мер) неизменно ведет к понижению другого. Так, обнаружение б*о*льшего числа инсайдеров (повышение обоснованности) может быть получено за счет снижения скорости обнаружения (снижение своевременности) и/или затрачивания большего количества ресурсов (повышение ресурсопотребления). Следовательно, повышение любого из параметров при условии сохранения остальных будет однозначно означать общее повышение эффективности системы. Этот принцип и будет использован в качестве основного критерия достижения цели диссертационного исследования. Также, поскольку обоснованность системы не может рассматриваться отдельно от качества достижения поставленной задачи, будем под ней рассматривать совокупность введенных мер: *r*, *p*, *a*, *e* и *f*.

Для сравнения показателя обоснованности разработанной системы были выбраны следующие альтернативные системы.

Cisco StealthWatch. Компанию Cisco Systems Inc. по праву можно считать одним из лидеров по производству телекоммуникационного оборудования. Как следствие, она также предлагает и соответствующие решения в области информационной безопасности, одним из которых является Cisco StealthWatch. Данный продукт предназначен для анализа сетевого трафика на наличие угроз информационной безопасности, включая анализ различных устройств и сервисов в КС (маршрутизаторы, серверы, виртуальные машины, пользовательские

устройства и др.). Результатом обработки трафика может быть выявление различных аномалий в нем, включая инсайдерские атаки. Основными возможностями продукта являются следующие:

1. Обнаружение активности инсайдерской деятельности (например, сканирование портов).
2. Обнаружение взаимодействия зараженных хостов с хостами злоумышленника.
3. Детектирование некорректных сетевых пакетов.
4. Выявлений аномалий в сетевом трафике.
5. Проверка выполнения политик безопасности.

Обобщенная схема компонентов продукта представлена на рисунке 3.9.

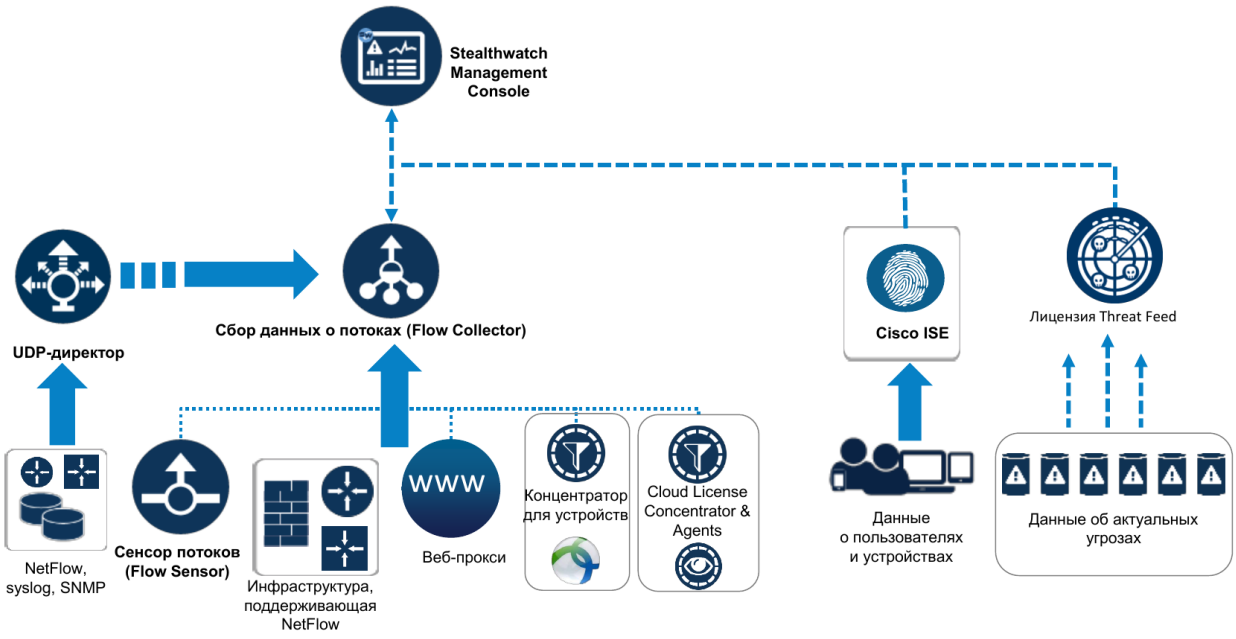

Рисунок 3.9 – Схема компонентов продукта Cisco StealthWatch

PacketFence. Альтернативой коммерческим решениям являются решения с открытым исходным кодом (Open Source продукты). В частности, одним из наиболее востребованных в области информационной безопасности КС является решение PacketFence, предназначенное для обеспечения контроля доступа в сеть. Возможности продукта схожи с Cisco StealthWatch и могут быть дополнены следующими функциями:

– Учет и регистрация новых устройств в сети.

– Обнаружение аномального трафика, связанного с потенциально враждебной активностью.

– Автоматическая или ручная изоляция проблемных хостов.

– Встроенные механизмы поиска уязвимостей.

Аналогично Cisco StealthWatch, данный продукт может быть использован как в небольших, так и в огромных гетерогенных сетях. Сетевая архитектура продукта представлена на рисунке 3.10.

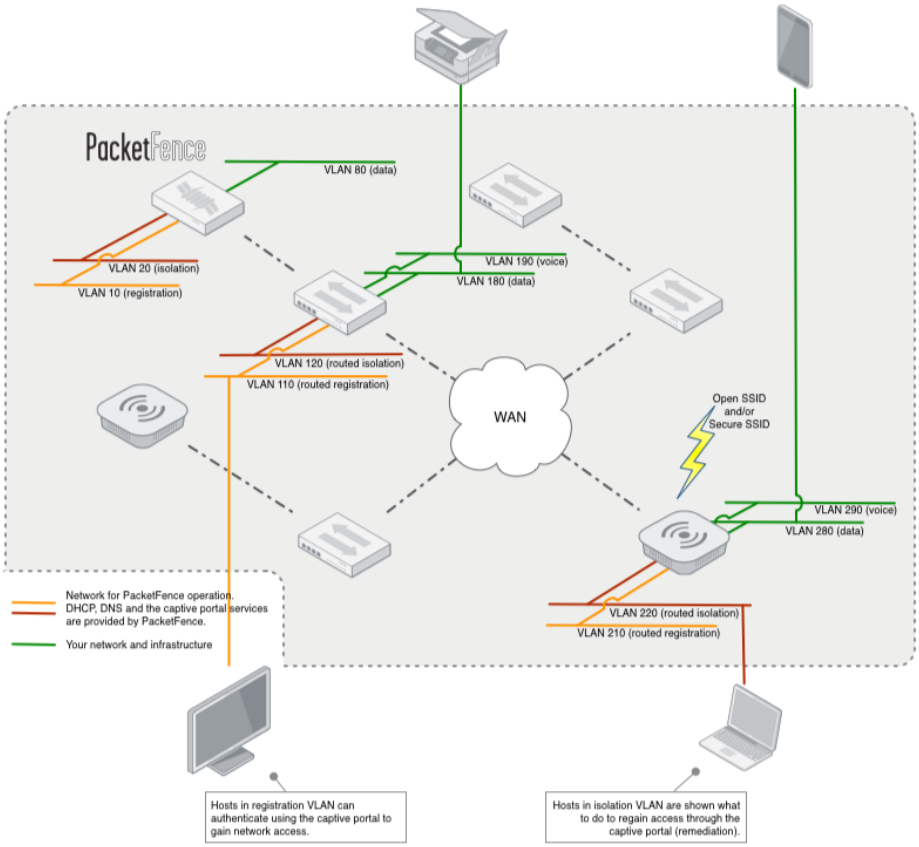

Рисунок 3.10 – Сетевая архитектура PacketFence

Согласно описанию продуктов, каждый из них может быть применен для решения поставленной задачи по обнаружению инсайдеров в КС и, следовательно, является ближайшим аналогом разрабатываемой системы.

Сравнение вариаций применения алгоритмов. Прежде чем сравнивать эффективность разработанной системы обнаружения инсайдеров с ближайшими аналогами, необходимо выбрать наилучшую вариацию алгоритма на основе экспертных правил и методов машинного обучения, на которую влияет формула вычисления конечного результата решения задачи по полученным результирующим множествам – их объединение, пересечение или выбор наилучшего с учетом различных классификаторов машинного обучения.

Опишем функциональную схему проведения экспериментов по оценке мер качества разработанной системы обнаружения (рисунок 3.11).

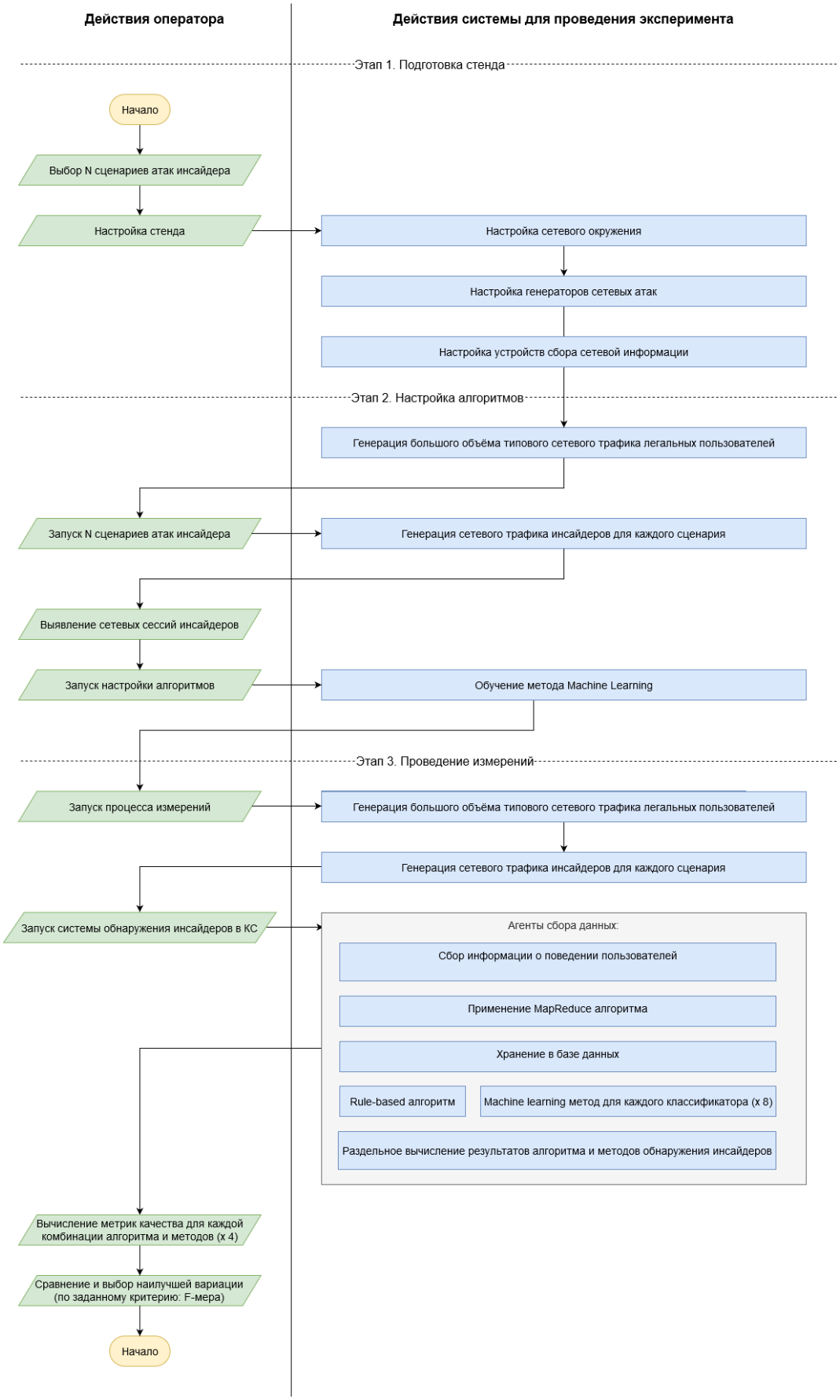

Рисунок 3.11 – Функциональная схема эксперимента по оценке мер качества системы обнаружения инсайдеров в КС

Функциональная схема состоит из следующих этапов и шагов:

Этап 1. Подготовка стенда

Шаг 1.1. Выбрать $N_{Scenary}$ типовых сценариев атак, на которых будет произведена экспериментальная оценка.

Шаг 1.2. Начать настройку стенда для проведения эксперимента.

Шаг 1.3. Настроить сетевое окружение.

Шаг 1.4. Настроить генератор сетевых атак.

Шаг 1.5. Настроить устройства сбора сетевой информации.

Этап 2. Настройка алгоритмов

Шаг 2.1. Сгенерировать большой объем типового трафика законных пользователей.

Шаг 2.2. Начать выполнение каждого из $N_{Scenary}$ типовых сценариев атак.

Шаг 2.3. Сгенерировать сетевой трафик инсайдеров по каждому из сценариев.

Шаг 2.4. Вручную проанализировать сетевой трафик и выявить сетевые сессии, связанные с инсайдерской деятельностью по каждому из сценариев.

Шаг 2.5. Начать настройку метода машинного обучения, используя сгенерированный сетевой трафик и выявленные в нем сессии, связанные с атаками инсайдеров.

Шаг 2.6. Произвести настройку метода машинного обучения.

Этап 3. Проведение измерений

Шаг 3.1. Начать процесс измерений.

Шаг 3.2. Сгенерировать большой объем типового сетевого трафика законных пользователей.

Шаг 3.3. Сгенерировать сетевой трафик инсайдеров по каждому из сценариев.

Шаг 3.4. Запустить программный комплекс системы обнаружения инсайдеров в КС.

Шаг 3.5. Произвести отработку всех элементов программного комплекса:

– сбор сетевой информации;

– применение алгоритма MapReduce;

– хранение информации в базе данных;

– выполнение алгоритма на основе экспертных правил;

– выполнение метода машинного обучения для каждого из его восьми классификаторов;

– комбинирование результатов работы алгоритма на основе экспертных правил и методов машинного обучения (25 вариаций): только алгоритм – 1, только метод – 8, пересечение для алгоритма и методов – 8; объединение для алгоритма и методов – 8;

Шаг 3.6. Вычисление мер качества (полнота, точность, аккуратность, ошибка, F-мера) для каждой из вариаций.

Шаг 3.7. Сравнение и выбор наилучшей вариации по совокупности мер качества.

Таким образом, выбранная вариация применения алгоритмов и классификаторов будет считаться итоговой для использования в методике.

Отметим, что Этапы 1 и 2 необходимы для обучения второго алгоритма из комплекса и должны быть проведены не только в рамках эксперимента, но и для настройки разрабатываемой системы с целью ее дальнейшего использования.

Информационная схема эксперимента, – то есть взаимосвязь исходных, промежуточных и результирующих данных, соответствующая описанной функциональной схеме, – представлена на рисунке 3.12.

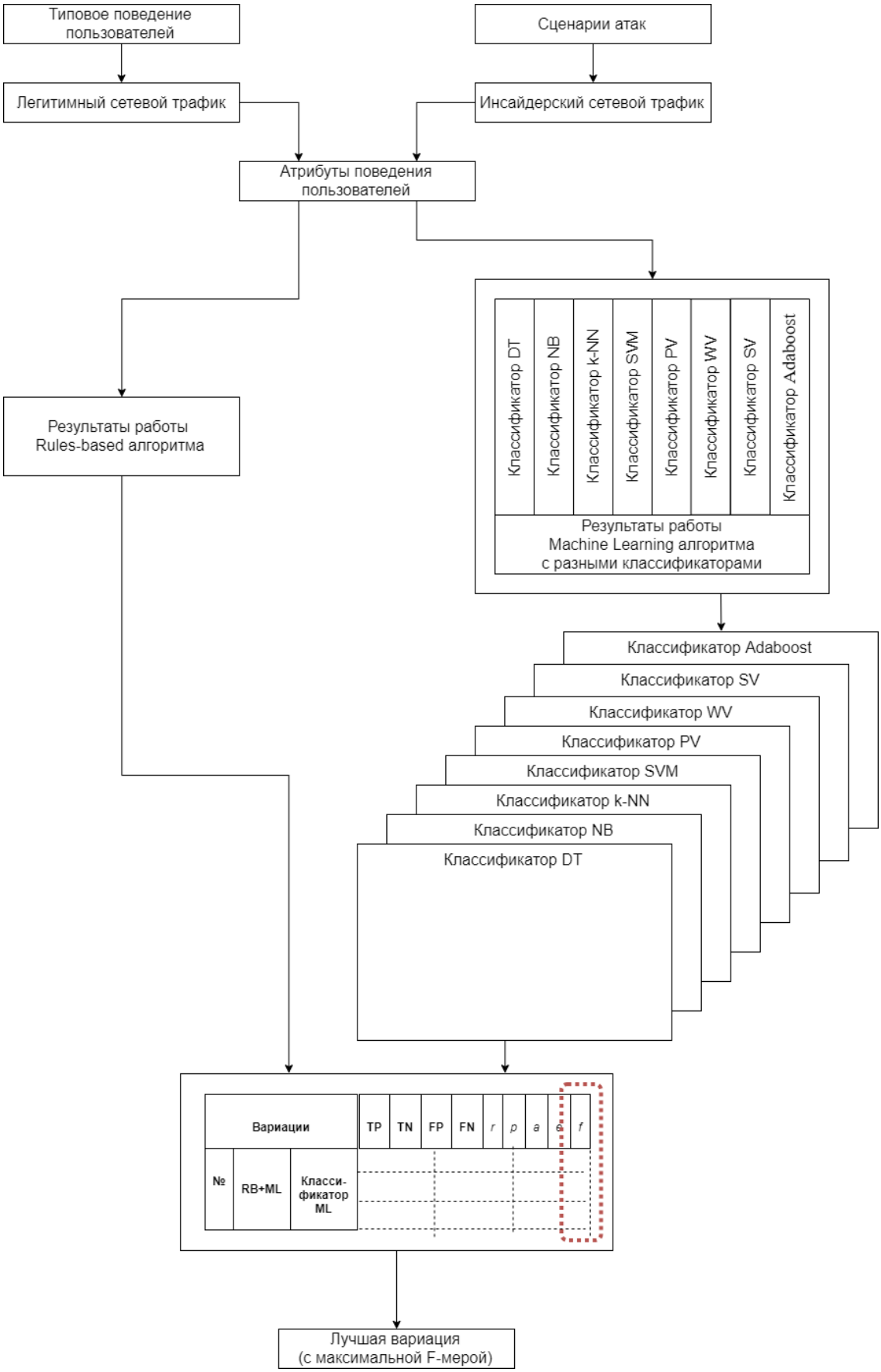

Рисунок 3.12 – Информационная схема эксперимента по оценке мер качества системы обнаружения инсайдеров в КС

Схема была реализована на базе стенда с архитектурой, состоящей из следующих компонентов:

1. Рабочее место оператора, выполняющего запуск, управление и анализ эксперимента.
2. Сканер сетевых пакетов.
3. Генераторы сценариев сетевых атак.
4. Разработанная система обнаружения инсайдеров в КС.

Для настройки работы алгоритма на основе экспертных правил пороговое значение, указывающее степень доверия к пользователю, было выбрано равным трем ($T = 3$). Так, в случае удовлетворения всем условиям алгоритма на базе экспертных правил: посещение подозрительных сайтов, начало или завершение сессии в нерабочее время, наличие приложения с подозрительной активностью, наличие нового периферийного устройства, наличие нестандартных подключений, наличие нестандартного устройства – счетчик (*Counter*) будет равен семи, в результате чего сессия получит пометку “Важность – Критическая”; в случае же удовлетворения более половине условий (Counter = 4, 5 или 6) сессия получит пометку – “Важность – Высокая”.

Настройка методов машинного обучения производилась следующим образом. В качестве обучающей выборки были взяты 100 000 записей, из числа которых 70 000 (т.е. 70%) соответствовали законному трафику пользователей, а 30 000 (т.е. 30%) были связаны с инсайдерскими атаками. Эксперимент по обнаружению инсайдеров проводился на выборке данных, состоящей из 20 000 записей (т.е. 20% от обучающей выборки), число записей, относящихся к инсайдерской деятельности составляло 10 000 (т.е. 50%).

Также, сценарии атак были поделены на две группы тестирования: в первую вошли 3 простых и 2 сложных, а во вторую – 2 комплексных. Такое деление было связано с предположением того, что наилучшее качество работы различных вариаций алгоритмов будет достигнуто именно в случае комплексной атаки, трудно описываемой одним подходом к обнаружению инсайдеров.

Результаты проведения эксперимента для 1-й группы сценариев позволили получить следующие значения для мер качества {*TP*, *TN*, *FP*, *FN*} а также {*r*, *p*, *a*, *e*, *f*} для каждой из вариаций алгоритмов, приведенных в таблице 3.1 (Клс. – тип классификатора машинного обучения).

Таблица 3.1 – Сравнение мер качества различных вариаций результатов обнаружения инсайдеров в КС для простых и сложных сценариев атак

Вариации			**Меры**								
№	**RB+ML**	**Клс. ML**	**TP**	**TN**	**FP**	**FN**	***r***	***p***	***a***	***e***	***f***
1	I_{RB}	(не исполь-зуется)	9933	9884	98	85	0.9915	0.9902	0.9909	0.0092	0.9909
2	I_{ML}	DT	9951	9964	56	29	0.9971	0.9944	0.9958	0.0043	0.9957
3	I_{ML}	NB	9942	9939	81	38	0.9962	0.9919	0.9941	0.0060	0.9941
4	I_{ML}	k-NN	9931	9977	43	49	0.9951	0.9957	0.9954	0.0046	0.9954
5	I_{ML}	SVM	9955	9949	51	45	0.9955	0.9949	0.9952	0.0048	0.9952
6	I_{ML}	PV	9947	9970	50	33	0.9967	0.9950	0.9959	0.0042	0.9958
7	I_{ML}	WV	9938	9933	87	42	0.9958	0.9913	0.9936	0.0065	0.9936
8	I_{ML}	SV	9947	9954	46	53	0.9947	0.9954	0.9951	0.0050	0.9950
9	I_{ML}	Adaboost	9948	9935	75	42	0.9958	0.9925	0.9942	0.0059	0.9942
10	$I_{RB} \vee I_{ML}$	DT	9985	9836	88	91	0.9910	0.9913	0.9911	0.0090	0.9911
11	$I_{RB} \vee I_{ML}$	NB	9940	9875	94	91	0.9909	0.9906	0.9908	0.0093	0.9908
12	$I_{RB} \vee I_{ML}$	k-NN	9921	9896	90	93	0.9907	0.9910	0.9909	0.0092	0.9909
13	$I_{RB} \vee I_{ML}$	SVM	9933	9886	92	89	0.9911	0.9908	0.9910	0.0091	0.9910
14	$I_{RB} \vee I_{ML}$	PV	9924	9899	89	88	0.9912	0.9911	0.9912	0.0089	0.9912
15	$I_{RB} \vee I_{ML}$	WV	9891	9930	88	91	0.9909	0.9912	0.9911	0.0090	0.9910
16	$I_{RB} \vee I_{ML}$	SV	9934	9883	93	90	0.9910	0.9907	0.9909	0.0092	0.9909
17	$I_{RB} \vee I_{ML}$	Adaboost	9916	9907	89	88	0.9912	0.9911	0.9912	0.0089	0.9912
18	$I_{RB} \wedge I_{ML}$	DT	9908	9934	66	92	0.9908	0.9934	0.9921	0.0079	0.9921
19	$I_{RB} \wedge I_{ML}$	NB	9956	9970	30	44	0.9956	0.9970	0.9963	0.0037	0.9963
20	$I_{RB} \wedge I_{ML}$	k-NN	9959	9933	67	41	0.9959	0.9933	0.9946	0.0054	0.9946
21	$I_{RB} \wedge I_{ML}$	SVM	9926	9960	40	74	0.9926	0.9960	0.9943	0.0057	0.9943

22	$I_{RB} \wedge I_{ML}$	PV	9924	9954	46	76	0.9924	0.9954	0.9939	0.0061	0.9939
23	$I_{RB} \wedge I_{ML}$	WV	9929	9953	47	71	0.9929	0.9953	0.9941	0.0059	0.9941
24	$I_{RB} \wedge I_{ML}$	SV	9928	9934	66	72	0.9928	0.9934	0.9931	0.0069	0.9931
25	$I_{RB} \wedge I_{ML}$	Adaboost	9941	9944	56	59	0.9941	0.9944	0.9943	0.0058	0.9942
	Заданные требования						≥ 0.90	≥ 0.92	≥ 0.92	≤ 0.06	≥ 0.92

Анализ полученных результатов (таблица 3.1) позволил сделать следующие выводы. Во-первых, для любых вариаций результатов работы алгоритма на основе экспертных правил и методов машинного обучения все меры – т.е. полнота (r), точность (p), аккуратность (a), ошибка (e), F-мера (f) – соответствуют заданным требованиям. Во-вторых, мера ошибки в случае объединения результатов работы алгоритмов имеет большее значение по сравнению с их пересечением. В-третьих, вариацией с наибольшим значением F-меры, равной **0.9963,** является пересечение алгоритма на базе экспертного правила и метода машинного обучения с классификатором NB (строка под номером 19, помеченная в таблице красным цветом). И, в-четвертых, показатели F-меры для любых комбинаций алгоритмов достаточно близки (разница между минимальным и максимальным значениями составляет ~0.5%). Таким образом, наилучшей с точки зрения F-меры вариацией для обнаружения инсайдеров с простыми и сложными сценариями атак является *пересечение результатов работы алгоритма на основе экспертных правил и метода машинного обучения при использовании классификатора NB*, однако, преимущество нельзя считать гарантированным. Это связано с тем, что при простых и сложных сценариях атак используемые алгоритмы показывают примерно одинаковый результат по обнаружению инсайдеров и, следовательно, сделать обоснованный выбор наилучшей комбинации алгоритмов и классификатора не представляется возможным.

Результаты проведения эксперимента для 2-й группы сценариев позволили получить следующие значения для мер качества {*TP*, *TN*, *FP*, *FN*} а также {*r*, *p*, *a*,

e, *f*} для каждой из вариаций алгоритмов, приведенных в таблице 3.2 (Клс. – тип классификатора машинного обучения):

Таблица 3.2 – Сравнение мер качеств различных вариаций результатов обнаружения инсайдеров в КС для комплексных сценариев атак

Вариации			**Меры**								
№	**RB+ML**	**Клс. ML**	**TP**	**TN**	**FP**	**FN**	***r***	***p***	***a***	***e***	***f***
1	I_{RB}	(не исполь-зуется)	8969	7781	2219	1031	0.90	0.80	0.84	0.16	0.85
2	I_{ML}	DT	9201	8111	1889	799	0.92	0.83	0.87	0.13	0.87
3	I_{ML}	NB	9341	8156	1844	659	0.93	0.84	0.87	0.13	0.88
4	I_{ML}	k-NN	9200	8109	1891	800	0.92	0.83	0.87	0.13	0.87
5	I_{ML}	SVM	9343	8136	1864	657	0.93	0.83	0.87	0.13	0.88
6	I_{ML}	PV	9394	8190	1810	606	0.94	0.84	0.88	0.12	0.89
7	I_{ML}	WV	9378	8163	1837	622	0.94	0.84	0.88	0.12	0.88
8	I_{ML}	SV	9399	8170	1830	601	0.94	0.84	0.88	0.12	0.89
9	I_{ML}	Adaboost	9362	8215	1785	638	0.94	0.84	0.88	0.12	0.89
10	$I_{RB} \vee I_{ML}$	DT	9822	7538	2462	178	0.98	0.80	0.87	0.13	0.88
11	$I_{RB} \vee I_{ML}$	NB	9913	7603	2397	87	0.99	0.81	0.88	0.12	0.89
12	$I_{RB} \vee I_{ML}$	k-NN	9728	7524	2476	272	0.97	0.80	0.86	0.14	0.88
13	$I_{RB} \vee I_{ML}$	SVM	9904	7501	2499	96	0.99	0.80	0.87	0.13	0.88
14	$I_{RB} \vee I_{ML}$	PV	9914	7660	2340	86	0.99	0.81	0.88	0.12	0.89
15	$I_{RB} \vee I_{ML}$	WV	9969	7669	2331	31	1.00	0.81	0.88	0.12	0.89
16	$I_{RB} \vee I_{ML}$	SV	9974	7780	2220	26	1.00	0.82	0.89	0.11	0.90
17	$I_{RB} \vee I_{ML}$	Adaboost	9901	7746	2254	99	0.99	0.81	0.88	0.12	0.89
18	$I_{RB} \wedge I_{ML}$	DT	8724	9607	393	1276	0.87	0.96	0.92	0.08	0.91
19	$I_{RB} \wedge I_{ML}$	NB	8734	9717	283	1266	0.87	0.97	0.92	0.08	0.92
20	$I_{RB} \wedge I_{ML}$	k-NN	8831	9620	380	1169	0.88	0.96	0.92	0.08	0.92
21	$I_{RB} \wedge I_{ML}$	SVM	8696	9801	199	1304	0.87	0.98	0.92	0.08	0.92
22	$I_{RB} \wedge I_{ML}$	PV	8890	9859	141	1110	0.89	0.98	0.94	0.06	0.93
23	$I_{RB} \wedge I_{ML}$	WV	8967	9754	246	1033	0.90	0.97	0.94	0.06	0.93
24	$I_{RB} \wedge I_{ML}$	SV	8968	9989	11	1032	0.90	1.00	0.95	0.05	0.95

25	$I_{RB} \wedge I_{ML}$	Adaboost	8885	9809	191	1115	0.89	0.98	0.93	0.07	0.93
	Заданные требования						≥ 0.90	≥ 0.92	≥ 0.92	≤ 0.06	≥ 0.92

Анализ полученных результатов (таблица 3.2) позволил сделать следующие выводы. Во-первых, только пересечения результатов работы алгоритмов (строчки 19, 20, 21, 22, 23, 24 и 25 в таблице) удовлетворяют требованиям F-меры (помечено синим и красным цветами). Во-вторых, в среднем, мера ошибки в случае объединения результатов работы алгоритмов имеет большее значение по сравнению с их пересечением. В-третьих, вариацией с наибольшим значением F-меры, равной **0.95,** является пересечение алгоритма на базе экспертного правила и метода машинного обучения с классификатором SV (строка под номером 24, помеченная в таблице красным цветом). И, в-четвертых, показатели F-меры для любых способов комбинаций алгоритмов достаточно различимы друг от друга (разница между минимальным и максимальным значениями составляет ~12%, а между наибольшим показателем F-меры и ближайшим ~2%). Таким образом, наилучшей с точки зрения F-меры вариацией для обнаружения инсайдеров с комплексными сценариями атак является *пересечение результатов работы алгоритма на основе экспертных правил и метода машинного обучения при использовании его классификатора SV*.

Необходимо отметить, что предложенная методика обнаружения инсайдеров в КС позволяет использовать любые из предложенных классификаторов и комбинаций результатов алгоритмов, выбираемых оператором на первом этапе. Так, например, с позиции максимального значения полноты (r), наилучшим можно считать объединение алгоритмов (строки с номерами 15 и 16).

Произведем анализ полученных экспериментальных результатов. Во-первых, алгоритм на базе экспертных правил хотя и обнаруживает большое количество инсайдеров (TP) и верно определяет часть законных пользователей (TN), вместе с тем ошибочно определяет законных пользователей в качестве инсайдеров (FP), а часть инсайдеров, соответственно, пропускается (FN); во-вторых, алгоритм на базе методов машинного обучения имеет похожее количество

обнаруженных и пропущенных инсайдеров (TP, FN), однако он реже ошибается на законных пользователях (TN выше, а FP ниже); объединение результатов работы алгоритмов практически полностью выявляет всех реальных инсайдеров (TP высокий, FN низкий), однако вместе с тем в данную категорию попадает и большее количество законных пользователей, ошибочно принимаемых за инсайдеров (TN низкий, FP высокий); пересечение результатов работы алгоритмов обнаруживает незначительно меньшее количество инсайдеров, чем объединение этих алгоритмов (TP ниже), но законные пользователи определяются существенно точнее (TN выше), а ошибок первого рода становится меньше (FP низкий), количество ошибок II-го рода несущественно выше аналогичного числа ошибок для отдельных алгоритмов (FN выше). Таким образом, пересечение результатов работы алгоритмов наиболее близко к *идеальной картине* обнаружения реальных инсайдеров в КС (TP и TN максимальные, FP и FN минимальные).

То есть, каждый из алгоритмов имеет достаточное количество как пропусков инсайдеров, так и ошибочных отношений числа законных пользователей к числу инсайдеров. По сравнению с работой каждого из алгоритмов, объединение их результатов улучшает обнаружение инсайдеров, но в тоже время увеличивает число ложных срабатываний. Пересечение же наоборот, ухудшает обнаружение инсайдеров, но в тоже время уменьшает число ложных срабатываний. Однако, усредненный эффект от работы пересечения (то есть совместная оценка с позиции полноты и точности) оказывается выше; при этом метод машинного обучения с классификатором SV также оказывается с этой позиции лучшим среди остальных методов машинного обучения.

Оценка обоснованности системы. Проверим соответствие данного показателя разработанной системы одному из нефункциональных требований, которое было задано в подразделе 1.4. Сравнение вычисленного показателя обоснованности системы (в виде мер качества) с требуемым значением подтверждает это.

Для сравнения обоснованности разработанной системы и ее ближайших аналогов необходимо вычислить и сравнить введенные меры качества для каждой

из систем. Для этого за основу была взята модификация эксперимента, аналогичного описанному в подразделе 3.3; суть модификации заключалась в использовании только выбранной вариации алгоритмов.

Результаты сравнения для 1-й группы (простые и сложные сценарии атак) и 2-й группы (комплексные сценарии атак) приведены в таблице 3.3.

Таблица 3.3 – Меры качества альтернативных систем обнаружения инсайдеров в КС

Название системы	**TP**	**TN**	**FP**	**FN**	***r***	***p***	***a***	***e***	***f***
1-я группа (простые и сложные сценарии атак)									
Cisco StealthWatch	9923	9896	104	77	0.99	0.99	0.99	0.0091	0.9910
PacketFence	9898	9801	199	102	0.99	0.98	0.98	0.0151	0.9850
2-я группа (комплексные сценарии атак)									
Cisco StealthWatch	9234	8934	1066	766	0.92	0.90	0.91	0.09	0.91
PacketFence	8843	8954	1046	1157	0.88	0.89	0.89	0.11	0.89

Сравнительный анализ мер качества для разработанной системы (таблицы 3.1 и 3.2) и ее ближайших аналогов (таблица 3.3) позволяет утверждать, что первая обладает лучшими показателями эффективности (согласно сравнению их F-мер) для обоих групп сценариев. Так, для 1-й группы сценариев атак F-мера разработанной системы превосходит ближайшего конкурента (Cisco StealthWatch) на $\frac{(0.9963 - 0.9910)}{0.9910} = 0.005\ (0.5\%)$, а для второй группы – на $\frac{(0.95 - 0.91)}{0.91} = 0.044\ (4.4\%)$. Полученные результаты можно обосновать тем, что разработанная система комплексирует в себе два алгоритма (на базе RB и ML), вбирая в себя наилучшие возможности каждого из них, и нивелируя при этом отрицательные.

Оценка своевременности системы. Проверим соответствие данного показателя разработанной системы одному из нефункциональных требований,

которое было задано в подразделе 1.4. Сравнение вычисленного показателя обоснованности системы с требуемым значением подтверждает это.

Для сравнения своевременности разработанной системы и ее ближайших аналогов необходимо вычислить и сравнить время обнаружения инсайдеров для каждой из систем. Для этого за основу была взята модификация эксперимента, аналогичного описанному в подразделе 3.3; суть модификации заключалась в использовании только выбранной вариации алгоритмов и вычислении времени обнаружения всех инсайдеров, связанных с генерируемыми сценариями атак.

Результаты сравнения среднего времени выполнения приведены в таблице 3.4.

Таблица 3.4 – Показатели своевременности разработанной и альтернативных систем обнаружения инсайдеров в КС

Название системы	**Время (сек) 1-я группа атак**	**Время (сек) 2-я группа атак**
Разработанная система	39	45
Cisco StealthWatch	42	48
PacketFence	46	52

Сравнительный анализ значений своевременности разработанной системы и ее ближайших аналогов позволяет утверждать, что первая обладает лучшим показателем эффективности.

Такой вывод также можно обосновать использованием в разработанной системе технологии распределенных вычислений (MapReduce).

Проверим соответствие вероятности (P^T) того, что время выполнения процесса обнаружения инсайдеров (T) разработанной системы не будет выше допустимого (T_{def}), заданному значению (P_0^T). Вероятность может быть вычислена по формуле:

$$P^T\left(T \le T_{def}\right) = F(Z),$$

где $F(Z)$ – значение функции Лапласа при

$$Z = \frac{T_{def} - T_{exp}}{\sqrt{\sigma^2(T_{exp})}},$$

для ожидаемого времени работы системы (T_{exp}) и его дисперсии ($\sigma^2(T_{exp})$), вычисляемых с помощью двухоценочной методики по следующим формулам:

$$\begin{cases} T_{exp} = \frac{3T_{min}+2T_{max}}{5} \\ \sigma^2(T_{exp}) = 0.4(T_{max} - T_{min})^2 \end{cases},$$

где T_{min} и T_{max}, соответственно, минимальное и максимальное время выполнения.

В процессе проведения эксперимента по обнаружению инсайдеров были произведены измерения времени работы разработанной системы для каждой атаки по каждому из сценариев (по 10 замеров на каждый сценарий). Результаты измерений приведены на следующей диаграмме (рисунок 3.13).

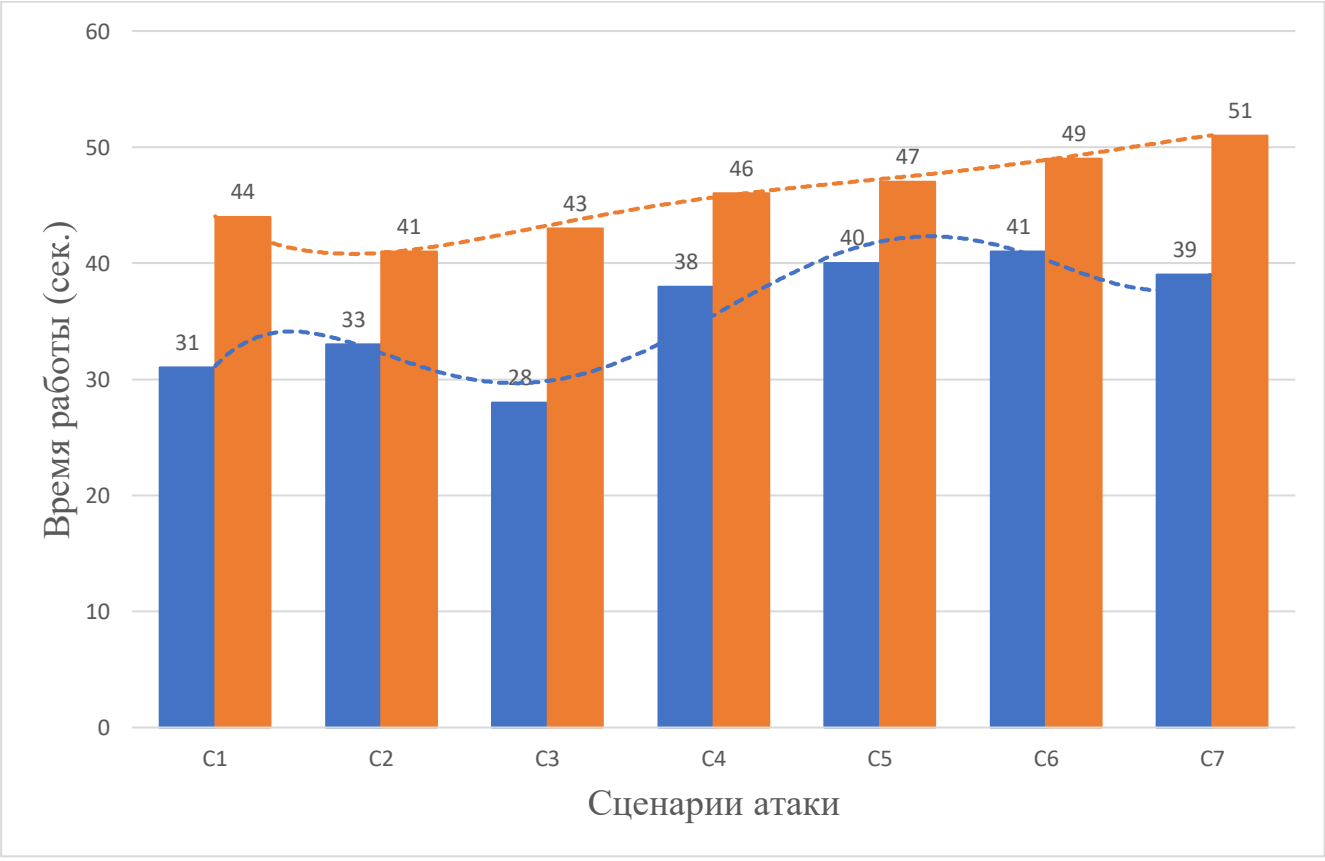

Рисунок 3.13 – Время работы разработанной системы для сценариев атак

Голубым цветом на рисунке помечено T_{min}, оранжевым – T_{max}.

Как видно из рисунка 3.13 по мере усложнения атак растет и время работы системы по обнаружению инсайдеров, что является закономерным.

Анализ результатов замеров позволяет получить следующие значения максимального и минимального времени работы системы:

$$\begin{cases} T_{min} = 35.71 \\ T_{max} = 45.86 \end{cases}.$$

Таким образом,

$$\begin{cases} T_{exp} = \frac{3*35.71+2*45.86}{5} = 39.77 \\ \sigma^2(T_{exp}) = 0.4(45.86 - 35.71)^2 = 41.15 \end{cases},$$

и, следовательно, для $T_{def} = 60$ сек.:

$$Z = \frac{T_{def} - T_{exp}}{\sqrt{\sigma^2(T_{exp})}} = \frac{60-39.77}{\sqrt{41.15}} = 3.15.$$

Значение функции Лапласа для $Z = 3.15$ равно 0.99837, что означает соответствие своевременности системы заданным требованиям:

$$P^T(T \leq 60) = 0.99837 > 0.98.$$

Оценка ресурсопотребления системы. Проверим соответствие данного показателя разработанной системы одному из нефункциональных требований, которое было задано в подразделе 1.4. Сравнение вычисленного показателя ресурсопотребления системы с требуемым значением подтверждает это.

Для сравнения ресурсопотребления разработанной системы и ее ближайших аналогов необходимо вычислить и сравнить все параметры в совокупности. Для этого за основу была взята модификация эксперимента, аналогичного описанному в подразделе 3.3. Суть модификации заключалась в использовании только выбранной вариации алгоритмов и вычислении параметров ресурсопотребления, затрачиваемого в процессе обнаружения всех инсайдеров, связанных с генерируемыми сценариями атак. Результаты сравнения приведены в таблице 3.5.

Таблица 3.5 – Показатели ресурсопотребления разработанной и альтернативных систем обнаружения инсайдеров в КС

Название системы	**Количество хостов (R_h)**	**Средний сетевой трафик (R_n),** Мб/сек	**Объем занимаемого пространства SSD/HDD (R_v), Гб**	**Средняя нагрузка на CPU (R_c)**	**Средняя загрузка памяти (R_S)**
Разработанная система	6	100	900	40%	45%
Cisco StealthWatch	6	100	1000	40%	45%
PacketFence	6	100	750	45%	50%

Сравнительный анализ значений ресурсопотребления разработанной системы и ее ближайших аналогов позволяет утверждать, что показатели всех трех систем приблизительно равны и могут не учитываться при сравнении.

Такой вывод также можно обосновать тем, что каждая система применяет собственные подходы и решения, имеющие достаточно высокий уровень оптимизации по ресурсопотреблению.

Исходя из того, что показатели своевременности и ресурсопотребления разработанной системы сравнимы с аналогичными показателями для ближайших аналогов (Cisco StealthWatch и PacketFence), а показатель обоснованности имеет превосходство (~5%), можно утверждать, что данная система обладает лучшей эффективностью, а, следовательно, повышает защищенность КС от инсайдерских атак.

Проведем сравнительный анализ разработанной методики с существующими исследованиями в предметной области на качественном уровне (таблица 3.6). В таблице 3.6 используются следующие обозначения: «+» указывает на наличие параметра в указанной работе, «+/-» говорит о частичном соответствии параметру, «-» говорит о его отсутствии. Сравнение производилось по следующим критериям:

- использование обработки больших данных;
- наличие распределенной обработки трафика;

– использование интеллектуальных систем (методов машинного обучения) [21];

– использование опыта эксперта;

– учет гетерогенности признаков.

Произведем сравнительный анализ и балльную оценку методик обнаружения инсайдеров следующим образом: «-» – 0 баллов, «+/-» – 0.5 балла, «+» – 1 балл.

Таблица 3.6 – Сравнительный анализ разработанной методики с существующими аналогами

Методика обнаружения инсайдеров	Учитываемые параметры					Оценка
	Использование обработки больших данных	Распределенная обработка	Использование интеллектуальных систем	Использование опыта эксперта	Учет гетерогенности признаков	
A Graph Based Framework for Malicious Insider Threat Detection [49]	+/-	+	+	-	+	3.5
Identifying and Visualizing the Malicious Insider Threat Using Bipartite Graphs [135]	+/–	–	+/–	+	+	4
Specializing network analysis to	+	+/–	+	–	+	3.5

detect anomalous insider actions [77]						
A probabilistic analysis framework for malicious insider threats [75]	+/–	–	+	–	+/–	2
A Behavioral Theory of Insider-Threat Risks: A System Dynamics Approach [106]	+	+/–	–	+	+	3.5
Предложен-ная методика	+	+	+	+	+	**5**

Анализ результатов сравнения существующих и предложенной методик обнаружения инсайдеров позволяет сделать следующие выводы. Во-первых, ни одна из методик, кроме предложенной, одновременно не удовлетворяет всем введенным критериям. Во-вторых, все методики в той или иной степени учитывают гетерогенность данных. В-третьих, полноценная распределенная обработка применяется только в предложенной методике и работе [49]. В-четвертых, отставание ближайших аналогов от предложенной методики составляет от 1 балла до 3-х. Таким образом, предложенная методика выигрывает у ближайших аналогов согласно введенным критериям; причины *проигрыша* других заключаются в следующем: в работах [49, 75, 77] не используется опыт экспертов, работа [135] использует обработку больших данных и интеллектуальных систем не в полном

объеме, работы [49, 77, 106] соответствует почти всем критериям лишь частично, в принцип работы [106] заложены лишь экспертные правила без использования возможностей интеллектуальных систем.

3.4 Предложения по применению системы обнаружения инсайдеров в КС с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных

Разработанные в диссертационной работе модели, алгоритмы, методика и система могут быть использованы для следующих решений.

Во-первых, предложенная модель представления больших данных может быть применена не только в интересах обнаружения инсайдеров, а также для анализа поведения законных пользователей в интересах повышения эффективности работы организации, оптимизации ее бизнес-процессов и др. Так, например, индикация и выявление закономерностей, связанных с ошибочными действиями пользователей по работе с внутренними документами может сигнализировать о том, что должностные обязанности сотрудников не являются очевидными им и требуют разъяснения. Частые пересылки небольших объемов данных между отделами могут говорить о востребованности их в использовании общей базы данных, что может быть решено созданием внутреннего информационно-справочного сервера.

Потенциально перспективным применением модели может стать анализ КС на предмет действий сотрудников, направленный на оценку текущего и прогнозирования будущего состояния организации в целом – определения ее тренда развития. Например, частое посещение информационных сайтов сотрудниками как в рабочее время, так и вне его, может говорить об их увлеченности поставленной задачей и высокой мотивацией работы в фирме. Связь определенных решений руководства компании (выпуск приказа о трудовой

дисциплине, роспуск отдела и т.п.) с посещением сотрудниками сайтов о найме на работу, очевидно, означает разочарованность сотрудников в организации и стремление сменить место работы, – что должно послужить сигналом для руководства о высоких рисках проектов.

Во-вторых, предложенные модель и алгоритмы комбинированного применения экспертных правил и методов машинного обучения, а также разработанная методика (после необходимой адаптации) применимы также и для противодействия внешним сетевым атакам, поскольку их основное ядро является достаточно гибким и не сильно зависимым от специфики КС. Например, алгоритм на основе экспертных правил может рассматривать инсайдера, как внешнего злоумышленника, и применять подобные правила для его обнаружения; а алгоритм на базе методов машинного обучения может быть настроен на обнаружение сценариев атак на организацию со стороны открытых сетей. При этом алгоритмы также могут работать в комплексе в рамках разработанной методики, которой потребуется несущественная модификация в части агентов сбора данных. Естественно, комплекс алгоритмов может быть дополнен и другими, наиболее подходящими для задачи противодействия внешним сетевым угрозам.

В-третьих, предложенная система обнаружения инсайдеров в КС может быть использована при разработке следующих, более крупных программных комплексов защиты от угроз информационной безопасности – как основа для реализации полноценной SIEM-системы нового поколения, осуществляющей не только обнаружение инсайдеров, но и их нейтрализацию еще до наступления существенного ущерба. Для этого потребуется согласование моделей и алгоритмов разработанной системы с работающими в SIEM-системах: необработанные данные сети должны изначально подвергаться анализу на предмет инсайдерской деятельности; разработанная модель должна быть источником данных для моделей защиты, синтезирующих контрмеры; должны быть реализованы способы оценки защищенности КС от инсайдерских действий; алгоритмы должны быть управляемы оператором, а также учитывать в своей работе защищенность КС; и др.

3.5 Выводы по главе 3

1. Предложена методика обнаружения инсайдеров в КС, использующая методы машинного обучения и обработки больших данных, состоящая из последовательности этапов и их шагов, описывающих действия оператора и системы обнаружения инсайдеров.

2. Разработана архитектура системы обнаружения инсайдеров в КС, обеспечивающая работу соответствующей методики; также реализован программный прототип системы обнаружения инсайдеров в КС.

3. Выбраны существующие системы обнаружения инсайдеров, близкие по функционалу к разработанной, сравнение с которыми позволило определить ее место среди аналогов.

4. Предложена схема эксперимента по вычислению характеристик разработанной методики и системы обнаружения инсайдеров в КС; описаны этапы и их шаги, а также взаимосвязь исходных, промежуточных и результирующих данных.

5. Произведено обучение алгоритма на базе машинного обучения с помощью типовых сценариев поведения инсайдеров в КС.

6. Произведен эксперимент и вычислены характеристики рассмотренных систем обнаружения инсайдеров в КС, его результаты показывают преимущества разработанной системы среди аналогов.

7. Предложены варианты применения разработанной системы обнаружения инсайдеров в КС.

Заключение

В диссертационной работе в целях повышения защищенности КС решена задача разработки модельно-методического аппарата для обнаружения инсайдеров в сети на основе комбинированного использования экспертных правил, методов машинного обучения и обработки больших данных, имеющая важное значение для развития технологий в области информационной безопасности. В том числе получены следующие научные результаты, составляющие **итоги** исследования:

1) проведен анализ существующих подходов к обнаружению инсайдеров в КС, моделей, методик и алгоритмов обнаружения инсайдеров в КС на основе методов машинного обучения и обработки больших данных;

2) разработана модель представления больших данных об инсайдерских атаках в формате NoSQL (включая модель инсайдера);

3) разработан алгоритм обнаружения инсайдеров в КС, основанный на экспертных правилах;

4) разработаны модель и алгоритмы комбинированного применения экспертных правил и методов машинного обучения в интересах обнаружения инсайдерских атак;

5) разработана методика обнаружения инсайдеров в компьютерных сетях с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных;

6) разработана архитектура системы обнаружения инсайдеров в компьютерных сетях с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных и выполнена ее программная реализация. Произведена настройка алгоритма на основе методов машинного обучения с помощью набора данных, характеризующих действия инсайдеров по заданному множеству сценариев атак и экспериментальная оценка разработанной методики системы обнаружения инсайдеров в КС.

Все результаты, выносимые на защиту, являются новыми. Разработана модель представления больших данных об инсайдерских атаках в формате NoSQL, отличающаяся от существующих возможностью обеспечения хранения и анализа признаков пользователей, полученных на базе UBA/UEBA-аналитики и характеризующих потенциальную инсайдерскую деятельность в компьютерных сетях, а также возможностью учета динамики изменения этих признаков. Также предложены новые модель и алгоритмы комбинированного применения экспертных правил и методов машинного обучения в интересах обнаружения инсайдерских атак, отличающиеся от существующих комплексным подходом к решению задачи обнаружения инсайдеров с учетом признаков и свойств пользователей, устройств, приложений, сервисов, включая параметр времени. Сформирована методика обнаружения инсайдеров, которая отличается от существующих использованием модели представления больших данных об инсайдерских атаках, а также модели и алгоритмов комбинированного применения экспертных правил, методов машинного обучения и обработки больших данных. Разработаны архитектура и программная реализация обнаружения инсайдеров, отличающиеся от известных реализацией комбинированного применения технологий обработки больших данных, экспертных правил и методов машинного обучения.

Сформулированы **рекомендации** по применению результатов работы в индустрии и в научных исследованиях. Так, например, модель представления больших данных об инсайдерских атаках путем ее модификации для наблюдения за законными пользователями может использоваться в интересах повышения эффективности работы организации. Развитие методики и алгоритмов обнаружения инсайдеров после необходимой доработки позволит их использовать для анализа внешних сетевых атак и их последующей нейтрализации. Полноценная реализация архитектуры системы обнаружения инсайдеров в КС позволит ей стать существенной частью крупных SIEM-систем. Кроме того, полученные в диссертации результаты создают перспективы для получения новых источников

сбора сведений о сетевой активности пользователей КС, а также их атрибутов, включая те из них, которые позволяют идентифицировать инсайдеров.

В качестве **перспектив дальнейшей разработки тематики** можно выделить следующие. Во-первых, расширение сценариев инсайдеров в КС, обнаруживаемых предложенными алгоритмами, методикой и системой. Во-вторых, интеграция механизмов автоматической нейтрализации инсайдерской деятельности, имеющей на основании работы алгоритмов критический уровень важности. В-третьих, расширение комплекса алгоритмов другими (такими, как отклонение от «сезонного поведения», применение дискретного вейвлет-анализа и др.), способствующими итоговому повышению качества и скорости обнаружения инсайдеров.

Все положения, выносимые на защиту, **соотнесены с пунктом 3 паспорта специальности** 05.13.19 – «Методы и системы защиты информации, информационная безопасность»: «Методы, модели и средства выявления, идентификации и классификации угроз нарушения информационной безопасности объектов различного вида и класса».

Список литературы

1) Банк данных угроз безопасности информации ФСТЭК России [Электронный ресурс]. – Режим доступа: https://bdu.fstec.ru/.

2) Беляев, И.В. Применение генетического алгоритма для повышения качества работы поисковых систем / И.В. Беляев, А.Р. Федоров, Л.Г. Гагарина // Известия высших учебных заведений. Электроника. – 2017. – Т. 22. – № 5. – С. 471-477.

3) Варламов, А.Д. Основные метрики, оценивающие качество работы систем поиска изображений / А.Д. Варламов // Алгоритмы, методы и системы обработки данных. – 2013. – № 2(24). – С. 3-11.

4) Василишин, Н.С. Исследование алгоритмов анализа сетевого трафика с использованием технологий больших данных для обнаружения компьютерных атак / Н.С. Василишин, И.А. Ушаков, И.В. Котенко // Аллея науки. – 2018. – Т. 3. – № 6(22). – С. 1012–1021.

5) Василишин, Н.С. Методы сбора и анализа сетевого трафика на основе технологий больших данных / Н.С. Василишин, Н.Д. Дубровин, И.А. Ушаков, А.А. Чечулин // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2017): сборник научных статей VI Международной научно-технической и научно-методической конференции. – СПб.: СПбГУТ, 2017. – С. 127-131.

6) Виткова, Л.А. Вопросы формирования безопасной информационной системы на основе технологии децентрализованных сетей / Л.А. Виткова, Е.И. Денисов, Д.В. Сахаров, И.А. Ушаков // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2018): сборник научных статей VII Международной научно-технической и научно-методической конференции. – СПб.: СПбГУТ, 2018. – С. 174-179.

7) Виткова, Л.А. Методы управления информационной безопасностью при возникновении чрезвычайных ситуации / Л.А. Виткова, В.С. Гераськина, И.А. Ушаков // Актуальные проблемы инфотелекоммуникаций в науке и образовании

(АПИНО-2018): сборник научных статей VII Международной научно-технической и научно-методической конференции. – СПб.: СПбГУТ, 2018. – С. 164-168.

8) Волкогонов, В.Н. Уязвимости программно-определяемых сетей / В.Н. Волкогонов, А.И. Преображенский, И.А. Ушаков // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2019): сборник научных статей VIII Международной научно-технической и научно-методической конференции. – СПб.: СПбГУТ, 2019. – С. 279-284

9) Десницкий, В.А. Защита информации в центрах обработки данных / В.А. Десницкий, Д.В. Сахаров, А.А. Чечулин, И.А. Ушаков, Т.Е. Захарова. – СПб.: СПбГУТ, 2019. – 92 с.

10) Дешевых, Е.А. Интеграция SIEM-систем с системами корреляции событий безопасности, основанных на технологии больших данных / Е.А. Дешевых, И.А. Ушаков, А.А. Чечулин // Информационные технологии в управлении (ИТУ-2016): материалы 9-й конференции по проблемам управления. – 2016. – С. 684-687.

11) Дешевых, Е.А. Исследование методов защиты от инсайдерских атак / Е.А. Дешевых, В.М. Конюхов, К.Ю. Крылов, И.А. Ушаков // Актуальные проблемы инфотелекоммуникаций в науке и образовании: сборник статей IV Международной научно-технической и научно-методической конференции. – СПб.: СПбГУТ, 2015. – С. 310-313.

12) Дешевых, Е.А. Обзор средств и платформ обработки больших данных для задач мониторинга информационной безопасности / Е.А. Дешевых, И.А. Ушаков, И.В. Котенко // Информационная безопасность регионов России (ИБРР-2015): материалы конференции. – 2015. – С. 67.

13) Дубровин, Н.Д. Применение технологии больших данных в системах управления информацией и событиями безопасности / Н.Д. Дубровин, И.А. Ушаков, А.А. Чечулин // Актуальные проблемы инфотелекоммуникаций в науке и образовании: сборник научных статей V международной научно-технической и научно-методической конференции. – СПб.: СПбГУТ, 2016. – С. 348–353.

14) Дубровин, Н.Д. Реализация прототипа на базе Hadoop для анализа больших данных / Н.Д. Дубровин, И.А. Ушаков, И.В. Котенко // Информационная безопасность регионов России (ИБРР-2015): материалы конференции. – 2015. – С. 69-70.

15) Захарова, Т.Е. Технология Trustsec, как инструмент обеспечения информационной безопасности / Т.Е. Захарова, И.А. Ушаков, В.Ю. Холоденко // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2018): сборник научных статей VII Международной научно-технической и научно-методической конференции. – СПб.: СПбГУТ, 2018. – С. 382-387.

16) Козьмян, А.В. Сравнительный анализ не реляционных баз данных / А.В. Козьмян, Ю.В. Твердохлебова, И.А. Ушаков // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2019): сборник научных статей VIII Международной научно-технической и научно-методической конференции. – СПб.: СПбГУТ, 2019. – С. 542-546.

17) Комарова, А.О. Экспериментальное исследование эффективности моделей информационного поиска / А.О. Комарова, Р.П. Калашников, С.П. Воробьев // Аллея Науки. – 2018. – Т. 8. – № 5(21). – С. 1159-1162.

18) Котенко, Д.И. Методы и средства моделирования атак в больших компьютерных сетях: состояние проблемы / Д.И. Котенко, И.В. Котенко, И.Б. Саенко // Труды СПИИРАН. – 2012. – № 3(22). – С. 5–30.

19) Котенко, И.В. Анализ защищенности компьютерных сетей на основе моделирования действий злоумышленников и построения графа атак / И.В. Котенко, М.В. Степашкин // Труды Института системного анализа Российской академии наук. – 2007. – Т. 31. – С. 126–207.

20) Котенко, И.В. Архитектура и программный прототип системы обнаружения инсайдера компьютерной сети на основе технологий больших данных / И.В. Котенко, А.Ю. Овраменко, И.А. Ушаков // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2019): сборник научных статей VIII Международной научно-технической и научно-методической конференции. – СПб.: СПбГУТ, 2019. – С. 568-572.

21) Котенко, И.В. Архитектура системы интеллектуальных сервисов защиты информации в критически важных инфраструктурах / И.В. Котенко, И.Б. Саенко // Труды СПИИРАН. – 2013. – № 1(24). – С. 21–40.

22) Котенко, И.В. Архитектура системы параллельной обработки больших данных для мониторинга безопасности сетей Интернета вещей / И.В. Котенко, И.Б. Саенко, А.Г. Кушнеревич // Труды СПИИРАН. – 2018. – № 4(59). – С. 5-30.

23) Котенко, И.В. Выявление инсайдеров в корпоративной сети: подход на безе UBA и UEBA / И.В. Котенко, И.А. Ушаков, Д.В. Пелевин, А.И. Преображенский, А.Ю. Овраменко // Защита информации. Инсайд. – 2019. – № 5(89). – С. 26-35.

24) Котенко, И.В. Гибридная модель базы данных NoSQL для анализа сетевого трафика / И.В. Котенко, И.А. Ушаков, Д.В. Пелевин, А.Ю. Овраменко // Защита информации. Инсайд. – 2019. – № 1(85). – С. 46-54.

25) Котенко, И.В. Использование технологий больших данных для мониторинга инцидентов информационной безопасности / И.В. Котенко, И.А. Ушаков // Региональная информатика "РИ-2016": материалы конференции. – 2016. – С. 168-169.

26) Котенко, И.В. Комплексный подход к обеспечению безопасности киберфизических систем на основе микроконтроллеров / И.В. Котенко, Д.С. Левшун, А.А. Чечулин, И.А. Ушаков, А.В. Красов // Вопросы кибербезопасности. – 2018. – № 3(27). – С. 29-38.

27) Котенко, И.В. Методики поиска инсайдеров в компьютерных сетях на основе технологий больших данных / И.В. Котенко, И.А. Ушаков //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2018): сборник научных статей VII Международной научно-технической и научно-методической конференции. – СПб.: СПбГУТ, 2018. – С. 501-506.

28) Котенко, И.В. Модели NoSQL баз данных для мониторинга кибербезопасности / И.В. Котенко, И.А. Ушаков // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2018): сборник научных

статей VII Международной научно-технической и научно методической конференции. – СПб.: СПбГУТ, 2018. – С. 498–501.

29) Котенко, И.В. Общая методика обнаружения инсайдера в компьютерной сети на основе технологий больших данных / И.В. Котенко, Д.В. Пелевин, И.А. Ушаков // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2019): сборник научных статей VIII Международной научно-технической и научно-методической конференции. – СПб.: СПбГУТ, 2019. – С. 572-576.

30) Котенко, И.В. Система сбора, хранения и обработки информации и событий безопасности на основе средств Elastic Stack / И.В. Котенко, А.А. Кулешов, И.А. Ушаков // Труды СПИИРАН. – 2017. – № 5(54). – С. 5–34.

31) Котенко, И.В. Технологии больших данных для мониторинга компьютерной безопасности / И.В. Котенко, И.А. Ушаков // Защита информации. Инсайд. – 2017. – № 3(75). – С. 23-33.

32) Красов, А.В. Масштабируемое Honeypot-решение для обеспечения безопасности в корпоративных сетях / А.В. Красов, Р.Б. Петрив, Д.В. Сахаров, Н.Л. Сторожук, И.А. Ушаков // Труды учебных заведений связи. – 2019. – Т. 5. – № 3. – С. 86-97.

33) Красов, А.В. Обеспечение безопасности передачи Multicast-трафика в IP-сетях / А.В. Красов, Д.В. Сахаров, И.А. Ушаков, Е.П. Лосин // Защита информации. Инсайд. – 2017. – № 3(75). – С. 34-42.

34) Красов, А.В. Построение доверенной вычислительной среды / А.В. Красов, А.М. Гельфанд, В.И. Коржик, И.В. Котенко, Р.Б. Петрив, Д.В. Сахаров, И.А. Ушаков, П.И. Шариков, Д.В. Юркин // СПб.: ИП Петрив Роман Богданович, 2019. – 108 с.

35) Обзор рынка систем поведенческого анализа – User and Entity Behavioral Analytics (UBA/UEBA) [Электронный ресурс]. – Режим доступа: https://www.anti-malware.ru/analytics/Market_Analysis/user-and-entity-behavioral-analytics-ubaueba.

36) Олифер, В. Компьютерные сети. Принципы, технологии, протоколы / В. Олифер, Н. Олифер. – СПб: Питер, 2017. – 992 с.

37) Паращук, И.Б. К вопросу обоснования систем показателей качества процессов принятия решения и поддержки принятия решения в интересах управления информационными сетями / И.Б. Паращук, А.С. Башкирцев // Информация и космос. – 2016. № 2. – С. 65-71.

38) Полтавцева, М.А. Модель угроз безопасности систем управления большими данными / М.А. Полтавцева, Д.П. Зегжда, М.О. Калинин // Проблемы информационной безопасности. Компьютерные системы. – 2019. – № 2. – С. 16-28.

39) Попкова, А.А. Оценка эффективности и сравнение моделей бизнес-аналитических проектов с применением технологий DATA MINING. Экономика, статистика и информатика / А.А. Попкова // Вестник УМО. – 2013. – № 4. – С. 184-190.

40) Робинсон, Я.Графовые базы данных. Новые возможности для работы со связанными данными / Я. Робинсон, Д. Вебер, Э. Эифрем. – М.: ДМК Пресс, 2016. – 256 с.

41) Савинов, Н.В. Исследование модели сети ЦОД на основе политик Cisco ACI / Н.В. Савинов, К.А. Токарева, И.А. Ушаков, А.В. Красов, Д.В. Сахаров // Защита информации. Инсайд. – 2019. – № 4(88). – С. 32-43.

42) Ушаков, И.А. Анализ методик применения концепции больших данных для мониторинга безопасности компьютерных сетей / И.А. Ушаков, И.В. Котенко, К.Ю. Крылов // Информационная безопасность регионов России (ИБРР-2015): материалы конференции. – 2015. – С. 75–76.

43) Ушаков, И.А. Модель обнаружения внутренних нарушителей на основе использования технологий больших данных / И.А. Ушаков, И.В. Котенко // Региональная информатика и информационная безопасность. – 2017. – С. 253-254.

44) Abourezq, M. Database-as-a-Service for Big Data: An Overview / M. Abourezq, A. Idrissi // International Journal of Advanced Computer Science and Applications (IJACSA). – 2016. – Vol. 7. – Iss. 1.

45) Al-Taharwa, I.A. Cloud-based anti-malware solution / I.A. Al-Taharwa, A.B. Jeng, H.-M. Lee, S.-M. Chen // The International Symposium on Grids and Clouds (ISGC). – 2011. – PP. 1-10.

46) Althebyan, Q. A knowledge-base model for insider threat prediction / Q. Althebyan, B. Panda // 2007 IEEE SMC Information Assurance and Security Workshop. – IEEE, 2007. – C. 239-246.

47) Alvaro, A. Big Data Analytics for Security Intelligence. Cloud Security Alliance / A. Alvaro. – P. 1-12

48) Alvesa, J. et al. Threat Intelligence. Analysis of using equivalent instructions at the hidden embedding of information into the executable files

49) Anagi, G.A Graph Based Framework for Malicious Insider Threat Detection / G. Anagi, S. Li, B. Serdar // Hawaii International Conference on System Sciences (HICSS). – 2017.

50) Anderson, J.P. Computer security threat monitoring and surveillance / J.P. Anderson // Technical Report, James P. Anderson Company. – 1980.

51) Andreolini, M. A software architecture for the analysis of large sets of data streams in cloud infrastructures / M. Andreolini, M. Colajanni, S. Tosi // Proceedings of the IEEE 11th International Conference on Computer and Information Technology (CIT). – 2011. – PP. 389–394.

52) Announcing Apache Knox [Электронный ресурс]. – Режим доступа: URL: https://knox.apache.org/.

53) Apache Hadoop [Электронный ресурс]. – Режим доступа: http://hadoop.apache.org/.

54) Apache HBase [Электронный ресурс]. – Режим доступа: https://hbase.apache.org/.

55) Apache Metron [Электронный ресурс]. – Режим доступа: https://github.com/apache/incubator-metron.

56) Apache Oozie [Электронный ресурс]. – Режим доступа: https://oozie.apache.org/.

57) Apache Pig [Электронный ресурс]. – Режим доступа: https://pig.apache.org/.

58) Apache Ranger [Электронный ресурс]. – Режим доступа: https://ranger.apache.org/.

59) Apache ZooKeeper [Электронный ресурс]. – Режим доступа: URL: https://zookeeper.apache.org/.

60) Avramov, L. / L. Avramov, M. Portolani // The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology. – Wiliams: Cisco Press, 2014. – 384 p.

61) Banks, J. Handbook of simulation: principles, methodology, advances, applications, and practice / J. Banks // John Wiley & Sons, 1998.

62) Barros, A. A Comparison of UEBA Technologies and Solutions [Электронный ресурс] / A. Barros, A. Chuvakin // Gartner. – Режим доступа: https://www.gartner.com/en/documents/3645381.

63) Bellovin, S. The insider attack problem nature and scope / S. Bellovin // Insider Attack and Cyber Security. – Springer, Boston: MA, 2008. – PP. 1-4.

64) Berdal, S. / S. Berdal. – A Holistic Approach to Insider Threat Detection : Diss. – 2018.

65) Bishop, M. A risk management approach to the “insider threat” / M. Bishop et al. // Insider threats in cyber security. – Springer, Boston: MA, 2010. – PP. 115-137.

66) Boyd, D. Critical Questions for Big Data / D. Boyd, K. Crawford // Information, Communication & Society. – 2012. – Vol. 15. – Iss. 5. – PP. 662-675.

67) Brackney, R. Understanding the Insider Threat / R. Brackney, R. Anderson // Proceedings of a March 2004 Workshop. – 2004.

68) Brdiczka, O. Proactive insider threat detection through graph learning and psychological context / O. Brdiczka et al. // 2012 IEEE Symposium on Security and Privacy Workshops. – IEEE, 2012. – PP. 142-149.

69) Cappelli, D. The CERT guide to insider threats: how to prevent, detect, and respond to information technology crimes (Theft, Sabotage, Fraud). / D. Cappelli, A. Moore, R. Trzeciak // Addison-Wesley. – 2012.

70) Caruana, G. A MapReduce based parallel SVM for large scale spam filtering / G. Caruana, M. Li, M. Qi // Proceedings of the 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). – 2011. – Vol. 4. – PP. 2659–2662.

71) Center, C. Insider Threat Control: Using a SIEM signature to detect potential precursors to IT Sabotage [Электронный ресурс]. / C. Center. – 2011. – Режим доступа: https://insights.sei.cmu.edu/insider-threat/2012/01/insider-threat-control-using-a-siem-signature-to-detect-potential-precursors-to-it-sabotage.html.

72) Cheh, C. Data-driven model-based detection of malicious insiders via physical access logs / C. Cheh et al. // International Conference on Quantitative Evaluation of Systems. – Springer, Cham, 2017. – PP. 275-291.

73) Chen, C. Data-intensive applications, challenges, techniques and technologies: A survey on big data / C. Chen, C.-Y. Zhang // Information Sciences. – 2014. – PP. 314–347.

74) Chen, M. Big data: Related Technologies, Challenges and Future Prospects / M. Chen, S. Mao, Y. Zhang, V. Leung // Springer, 2014.

75) Chen, T. A probabilistic analysis framework for malicious insider threats / T. Chen et al. // International Conference on Human Aspects of Information Security, Privacy, and Trust. – Springer, Cham, 2015. – PP. 178-189.

76) Chen, Y. NeuroNet: An Adaptive Infrastructure for Network Security / Y. Chen, H. Chen // International Journal of Information, Intelligence and Knowledge. – 2009. – Vol. 1. – Iss. 2.

77) Chen, Y. Specializing network analysis to detect anomalous insider actions / Y. Chen, S. Nyemba, W. Zhang, B. Malin // Security Informatics. – 2012. – Vol. 1. – Art. 5. – PP. 1-24.

78) Cisco Annual Cybersecurity Report 2018 [Электронный ресурс]. – Режим доступа: https://www.cisco.com/c/dam/m/hu_hu/campaigns/security-hub/pdf/acr-2018.pdf.

79) Cisco White Paper, Cisco Visual Networking Index: Forecast and Trends [Электронный ресурс]. – Режим доступа:

https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.

80) Classes. OrientDB Manual [Электронный ресурс]. – Режим доступа: https://orientdb.com/docs/2.0/orientdb.wiki/Tutorial-Classes.html.

81) Cole, E. / E. Cole, S. Ring // Insider threat: Protecting the enterprise from sabotage, spying, and theft. – Elsevier, 2005.

82) Collins, M. Common sense guide to prevention and detection of insider threats 5th edition / M. Collins, M. Theis, R. Trzeciak, J. Strozer, J. Clark, D. Costa, T. Cassidy, M. Albrethsen, A. Moore // Published by CERT, Software Engineering Institute, Carnegie Mellon University. – 2016.

83) Dantu, R. Risk Management using Behavior based Attack Graphs / R. Dantu, K. Loper, P. Kolan // Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04). – 2004.

84) Dean, J. MapReduce: Simplified Data Processing on Large clusters / J. Dean, S. Ghemawat // Communications of the ACM. – 2008. – Vol. 51. – Iss. 1. – PP. 107-113.

85) Dimkov, T. Portunes: representing attack scenarios spanning through the physical, digital and social domain / T. Dimkov, W. Pieters, P. Hartel // Joint Workshop on Automated Reasoning for Security Protocol Analysis and Issues in the Theory of Security. – Springer, Berlin, Heidelberg, 2010. – PP. 112-129.

86) DTCC and FS-ISAC [Электронный ресурс]. – Режим доступа: https://soltra.com/.

87) Dumitras, T. Toward a Standard Benchmark for Computer Security Research: the Worldwide Intelligence Network Environment (WINE) / T. Dumitras, D. Shou // Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS'11). – Salzburg, Austria, 2011. – PP. 89-96.

88) Dutta, P. Simulated User Bots: Real Time Testing of Insider Threat Detection Systems / P. Dutta et al. // 2018 IEEE Security and Privacy Workshops (SPW). – IEEE, 2018. – PP. 228-236.

89) Emani, C. Understandable big data: A survey. / C. Emani, N. Cullot, C. Nicolle // Computer science review. – Vol. 17. – PP. 70–81.

90) Fagade, T. Malicious Insider Threat Detection: A Conceptual Model / T. Fagade, T. Tryfonas // Security and Protection of Information. – 2017. – PP. 31-44.

91) Furht, B. Introduction to big data / B. Furht, F. Villanustre // In Big Data Technologies and Applications. Springer International Publishing, 2006. – PP. 3-11.

92) Furht, B. Random forest implementation and optimization for Big Data analytics on LexisNexis's high performance computing cluster platform / B. Furht // J Big Data. – 2019. – Vol. 6 – Iss. – PP. 68.

93) Gandomi, A. Beyond the hype: Big data concepts, methods, and analytics / A. Gandomi, M. Haider // International Journal of Information Management. – 2015. – Vol. 35. – PP. 137–144.

94) Giura, P. Using Large Scale Distributed Computing to Unveil Advanced Persistent Threats / P. Giura, W. Wang // Science Journal. – 2013. – Vol. 1. – Iss. 3. – PP.93-105.

95) Gooyert, V. Developing dynamic organizational theories; three system dynamics based research strategies / V. Gooyert // Quality & Quantity. – 2019. – Vol. 53. – Iss. 2. – PP. 653-666.

96) Gorodetski, V. The Multi-agent Systems for Computer Network Security Assurance: frameworks and case studies / V. Gorodetski, I. Kotenko // 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS-2002). – 2002. – PP. 297-302.

97) Graves, J. How machine learning is catching up with the insider threat [Электронный ресурс] / J. Graves // Cyber Security: A Peer-Reviewed Journal. – Режим доступа:
URL:https://www.henrystewartpublications.com/sites/default/files/CSJGraves.pdf.

98) Greitzer, F. Unintentional insider threat: contributing factors, observables, and mitigation strategies / F. Greitzer et al. // 47th Hawaii International Conference on System Sciences. – IEEE, 2014. – PP. 2025-2034.

99) Harilal, A. The Wolf Of SUTD (TWOS): A Dataset of Malicious Insider Threat Behavior Based on a Gamified Competition / A. Harilal, F. Toffalini, I. Homoliak, J. Castellanos, J. Guarnizo, S. Mondal, M. Ochoa // Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA). – 2018. – Vol. 9. – Iss. 1. – PP. 1-32.

100) Herrera, V. Random forest implementation and optimization for Big Data analytics on LexisNexis's high performance computing cluster platform / V. Herrera et al. // Journal of Big Data. – 2019. – Vol. 6. – Iss 1. – PP. 68.

101) Householder, A. Computer Attack Trends Challenge Internet Security / A. Householder, K. Houle, C. Dougherty // Security & Privacy, Supplement to IEEE Computer. – 2002.

102) Huang, Y. Dynamic variable precision rough set approach for probabilistic set-valued information systems / Y. Huang, T. Li, C. Luo, H. Fujita, S.-j. Horng // Knowledge-Based Systems. – 2017. – Vol. 122. – PP. 131–147.

103) Huang, Y. Matrix-based dynamic updating rough fuzzy approximations for data mining / Y. Huang, T. Li, C. Luo, H. Fujita, S.-j. Horng // Knowledge Based Systems. – 2017. – Vol. 119. – PP. 273–283.

104) Hunker, J. Insiders and Insider Threats-An Overview of Definitions and Mitigation Techniques / J. Hunker, C. Probst // JoWUA. – 2011. – Vol. 2. – Iss. 1. – PP. 4-27.

105) IBM Watson Analytics [Электронный ресурс]. – Режим доступа: https://www.ibm.com/ analytics/ru/ru/watson-analytics/.

106) Ignacio, J. A Behavioral Theory of Insider-Threat Risks: A System Dynamics Approach / J. Ignacio, M. Martinez, R. Eliot, C. Stephen, D. Andersen, T. Stewart // ACM Transactions on Modeling and Computer Simulation (TOMACS). – Vol. 18. – Iss. 2. – PP. 1-27.

107) Indyk, W. Web spam detection using MapReduce approach to collective classification / W. Indyk, T. Kajdanowicz, P. Kazienko, S. Plamowski // Proceedings of the International Joint Conference CISIS12-ICEUTE´12-SOCO´12 Special Sessions. – Springer, 2013. – PP. 197–206.

108) Jadhav, A. A survey on approaches to efficient classification of data streams using concept drift / A. Jadhav, L. Deshpande // International Journal. – Vol. 4.

109) Kammüller, F. Modeling and verification of insider threats using logical analysis / F. Kammüller, C. Probst // IEEE systems journal. – 2015. – Vol. 11. – Iss 2. – PP. 534-545.

110) Kantzavelou, I. A game-based intrusion detection mechanism to confront internal attackers / I. Kantzavelou, S. Katsikas // Computers & Security. – 2010. – Vol. 29. – Iss. 8. – PP. 859-874.

111) Khan, N. Big data: survey, technologies, opportunities, and challenges / N. Khan, I. Yaqoob, I. Hashem, Z. Inayat, A. Mahmoud, M. Alam, M. Shiraz, A. Gani // The Scientific World Journal. – 2014.

112) Kim, J. Insider Threat Detection Based on User Behavior Modeling and Anomaly Detection Algorithms / J. Kim et al. // Applied Sciences. – 2019. – Vol. 9. – Iss. 19. – PP. 4018.

113) Korshunov, G. Decision support systems for information protection in the management of the information network / G. Korshunov, V. Lipatnikov, A. Shevchenko // Fuzzy Technologies in the Industry - FTI 2018 Proceedings of the II International Scientific and Practical Conference. – 2018. – PP. 418-426

114) Kotenko, I. Aggregation of Elastic Stack instruments for collecting, storing and processing of security information and events / I. Kotenko, A. Kuleshov, I. Ushakov // 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). – 2017. – PP. 1550-1557.

115) Kotenko, I. An Approach to Aggregation of Security Events in Internet-of-Things Networks Based on Genetic Optimization / I. Kotenko, I. Saenko // The 16th IEEE International Conference on Scalable Computing and Communications (ScaleCom 2016). – 2016. – PP. 657-664.

116) Kotenko, I. Framework for Mobile Internet of Things Security Monitoring based on Big Data Processing and Machine Learning / I. Kotenko, I. Saenko, A. Branitskiy // IEEE Access. – 2018. – Vol.6. – PP. 10.

117) Kotenko, I. Parallelization of security event correlation based on accounting of event type links / I. Kotenko, A. Fedorchenko, I. Saenko, A. Kushnerevich // Proceedings of the 25th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP 2018). – Cambridge, UK, 2018. – PP. 462-469.

118) Kotenko, I., Saenko I. Creating New Generation Cybersecurity Monitoring and Management Systems / I. Kotenko, I. Saenko // Herald of the Russian Academy of Sciences. – 2014. – Vol. 84. – Iss. 6. – PP. 993-1001.

119) Kunda, D. A Comparative Study of NoSQL and Relational Database / D. Kunda, H. Phiri // Zambia (ICT) Journal. – 2017. Vol. 1. – Iss. 1.

120) Kune, R., Konugurthi P., Agarwal A., Chillarige, R., Buyya R. – 2016.

121) Le, D. Evaluating insider threat detection workflow using supervised and unsupervised learning / D. Le, A. Zincir-Heywood // 2018 IEEE Security and Privacy Workshops (SPW). – IEEE, 2018. – PP. 270-275.

122) Leveraging a Big Data Model in the Network Monitoring Domain. White Paper. VSS Monitoring [Электронный ресурс]. – 2014. – Режим доступа: http://www.vssmonitoring.com/ resources/whitepapers/Leveraging-a-BD-Model-Whitepaper.pdf.

123) Lindauer, B. Generating Test Data for Insider Threat Detectors / B. Lindauer, J. Glasser, M. Rosen, K. Wallnaul // Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications. – 2014. – Vol. 5. – Iss. 2. – PP. 80-94.

124) Liu, D. Game-theoretic modeling and analysis of insider threats / D. Liu, X. Wang, J. Camp // International Journal of Critical Infrastructure Protection. – 2008. – Vol. 1. – PP. 75-80.

125) Liu, D. Mitigating inadvertent insider threats with incentives / D. Liu, X. Wang, L. Camp // International Conference on Financial Cryptography and Data Security. – Springer, Berlin, Heidelberg, 2009. – PP. 1-16.

126) Logstash [Электронный ресурс]. – Режим доступа: https://www.elastic.co/products/logstash.

127) Luckey, D. Assessing Continuous Evaluation Approaches for Insider Threats: How Can the Security Posture of the US Departments and Agencies Be Improved / D. Luckey et al. // Santa Monica, CA: RAND Corporation, 2019.

128) Luo, C. Efficient updating of probabilistic approximations with incremental objects / C. Luo, T. Li, H. Chen, H. Fujita, Z. Yi // Knowledge-Based Systems. – Vol. 109. – PP. 71–83.

129) Magklaras, G. Insider threat prediction tool: Evaluating the probability of IT misuse / G. Magklaras, S. Furnell // Computers & Security. – 2002. – Vol. 21. – Iss. 1. – PP. 62–73.

130) McAfee, A. Big Data: The Management Revolution / A. McAfee, E. Brynjolfsson et al. // Harvard business review. – 2012. – Vol. 90. – PP. 60–68.

131) Modi, C. A survey of intrusion detection techniques in cloud / C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan // Journal of Network and Computer Applications. – Vol. 36. – Iss. 1. – 2013. – PP. 42–57.

132) Moriano, P. Stopping the Insider at the Gates: Protecting Organizational Assets through Graph Mining / P. Moriano et al. // JoWUA. – 2018. – Vol. 9. – Iss. 1. – PP. 4-29.

133) Myerson, R. Game theory / R. Myerson // Harvard university press, 2013.

134) Najafabadi, M. Deep learning applications and challenges in big data analytics / M. Najafabadi, F. Villanustre, T. Khoshgoftaar, N. Seliya, R. Wald, E. Muharemagic // Journal of Big Data. – 2015. – Vol. 2. – Iss. 1.

135) Nance, K. Identifying and Visualizing the Malicious Insider Threat Using Bipartite Graphs / K. Nance, R. Marty // 44-th Hawaii Int. Conf. on System Sciences. – 2011. – PP. 1-9.

136) Nebrase, E. Insider threats in information security categories and approaches / E. Nebrase, Y. Shuang-Hua, Y. Lili // The 21st International Conference on Automation and Computing (ICAC). – Glasgow, United Kingdom, 2015. – PP. 1-6.

137) Neumann, P. Combatting insider threats / P. Neumann // Insider Threats in Cyber Security. – Springer, Boston, MA, 2010. – PP. 17-44.

138) Noel, S. Multiple coordinated views for network attack graphs / S. Noel, M. Jacobs, P. Kalapa, S. Jajodia // IEEE Workshop on Visualization for Computer Security (VizSEC2005). – Minneapolis, MN, 2005.

139) Noever, D. Classifier Suites for Insider Threat Detection / D. Noever // arXiv preprint arXiv:1901.10948. – 2019.

140) NoSQL stores // ACM Computing Surveys. – 2018. – Vol. 51. – Iss. 2. – Art.40.

141) Oh, J. Advanced insider threat detection model to apply periodic work atmosphere / J. Oh, T. Kim, K. Lee // TIIS. – 2019. – Vol. 13. – Iss 3. – PP. 1722-1737.

142) Oltsik, J. Defining Big Data Security Analytics. Networking Nuggets and Security Snippets (Blog) [Электронный ресурс] / Oltsik J. – 2013. – Режим доступа: http://www.networkworld.com/ community/blog/defining-big-data-security-analytics.

143) Ophoff, J. A descriptive literature review and classification of insider threat research [Электронный ресурс] / J. Ophoff, A. Jensen, J. Sanderson-Smith, M. Porter, K. Johnston // Proceedings of Informing Science & IT Education Conference (InSITE). – 2014. – PP. 211-223. – Режим доступа: http://Proceedings.InformingScience.org/InSITE2014/InSITE14p211-223Ophoff0543.pdf.

144) OrientDB [Электронный ресурс]. – Режим доступа: https://orientdb.com.

145) Ou, X. MulVAL: A Logic-based Network Security Analyzer / X. Ou, S. Govindavajhala, A. Appel // 14th Usenix Security Symposium (August). – 2005.

146) Park, K. On the relationship between file sizes, transport protocols, and self-similar network traffic / K. Park, G. Kim, M. Crovella // In Proceedings of International Conference on Network Protocols. – 1996. – PP. 171-180.

147) Petrenko, A. Big data technologies in the information security field / A. Petrenko, S. Petrenko // Protection of the information. Inside. – 2016. – Iss. 4. – P.82-88.

148) Pfleeger, S. Insiders behaving badly: Addressing bad actors and their actions / S. Pfleeger et al. // IEEE Transactions on Information Forensics and Security. – 2009. – Vol. 5. – Iss. 1. – PP. 169-179.

149) Pham, N. Detection of recurring software vulnerabilities / N. Pham, T. Nguyen, H. Nguyen, T. Nguyen // Proceedings of the IEEE/ACM International Conference on Automated Software Engineering. – Antwerp, Belgium, 2010. – PP. 447–456.

150) Philip, R. Enabling Distributed Security in Cyberspace / R. Philip et al. // Departament of Homeland Security. – 2011.

151) Phyo, A. A detection-oriented classification of insider it misuse / A. Phyo, S. Furnell // Third Security Conference. – 2004.

152) Priyanka, A Review of NoSQL Databases, Types and Comparison with Relational Database / Priyanka, AmitPal // International Journal of Engineering Science and Computing (May). – 2016.

153) Rajaraman, V. Big data analytics / V. Rajaraman // Resonance. – Vol. 21. – PP. 695–716.

154) Razzak, M. Deep learning for medical image processing: Overview, challenges and future / M. Razzak, S. Naz, A. Zaib // arXiv preprint arXiv:1704.06825.

155) Reason, J. Human error / J. Reason // Cambridge university press. – 1990.

156) Reguieg, H. Using MapReduce to scale events correlation discovery for business processes mining / H. Reguieg, F. Toumani, H. Motahari-Nezhad, B. Benatallah // Business Process Management. – Springer, 2012. – PP. 279–284.

157) Saenko, I. Parallel Processing of Big Heterogeneous Data for Security Monitoring of IoT Networks / I. Saenko, I. Kotenko, A. Kushnerevich // Proceedings of the 25th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP 2017). – St. Petersburg, Russia, 2017. – PP. 329-336.

158) Salem, M. A survey of insider attack detection research / M. Salem, S. Hershkop, S. Stolfo // Insider Attack and Cyber Security. – Springer, Boston, MA, 2008. – PP. 69-90.

159) Salem, M., Stolfo S. Masquerade attack detection using a search-behavior modeling approach / M. Salem, S. Stolfo // Columbia University, Computer Science Department, Technical Report CUCS-027-09. – 2009.

160) Santos, O. Network Security with NetFlow and IPFIX: Big Data Analytics for Information Security / O. Santos // Cisco Press. – 2015. – 320 p.

161) Sasaki, T. A Framework for Detecting Insider Threats using Psychological Triggers / T. Sasaki // JoWUA. – 2012. – Vol. 3. – Iss 1/2. – PP. 99-119.

162) SASL (Simple Authentication and Security Layer) [Электронный ресурс]. – Режим доступа: URL: https://ru.bmstu.wiki/SASL_(Simple_Authentication_and_Security_Laye).

163) Schonlau, M. Computer intrusion: Detecting masquerades / M. Schonlau et al. // Statistical science. – 2001. – Vol. 16. – Iss. 1. – PP. 58-74.

164) Sharma, A. User And Entity Behavior Analytics (UEBA) [Электронный ресурс] / A. Sharma // Market 2018-2025 Global Industry Research Report. – Режим доступа: URL:https://www.researchgate.net/publication/330958395_User_And_Entity_Behavior_Analytics_UEBA_Market_2018-2025_Global_Industry_Research_Report.

165) Shashanka, M. User and entity behavior analytics for enterprise security / M. Shashanka, M. Shen, J. Wang // 2016 IEEE International Conference on Big Data (Big Data). – Washington, DC, 2016. – PP. 1867-1874.

166) Shim, K. MapReduce algorithms for big data analysis / K. Shim // Proceedings of the VLDB Endowment. – 2012. – Vol. 5, – Iss. 12. – PP. 2016–2017.

167) Shteiman, B. UEBA: Applying data science and machine learning to cybersecurity [Электронный ресурс] / B. Shteiman // Computer Business Review. – Режим доступа: http://www.cbronline.com/news/cybersecurity/protection/ueba-finding-cyber-security-norm-data-science-machine-learning/.

168) Shu, X. Massive Distributed and Parallel Log Analysis For Organizational Security / X. Shu, J. Smiy, D. Yao, H. Lin // IEEE Globecom Workshops (December). – 2013. – P.194-199.

169) Skowron, A. Interactive granular computing / A. Skowron, A. Jankowski, S. Dutta // Granular Computing. – 2016. – Vol. 1. – PP. 95–113.

170) Stepashkin, I. Network Security Evaluation based on Simulation of Malefactor's Behavior / I. Stepashkin // SECRYPT 2006 – International Conference on Security and Cryptography, Proceedings International Conference on Security and Cryptography (SECRYPT 2006). – 2006. – PP. 339–344.

171) Straub, D. Coping with systems risk: security planning models for management decision making / D. Straub, R. Welke // MIS quarterly. – 1998. – PP. 441-469.

172) Sun, J. Dynamic financial distress prediction with concept drift based on time weighting combined with adaboost support vector machine ensemble / J. Sun, H. Fujita, P. Chen, H. Li // Knowledge-Based Systems. – Vol. 120. – PP. 4–14.

173) Ted, E. Detecting insider threats in a real corporate database of computer usage activity / E. Ted et al. // Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. – ACM, 2013. – PP. 1393-1401.

174) The anatomy of big data computing // Software: Practice and Experience. – Vol. 46. – PP. 79–105.

175) Tsai, C. Big data analytics / C. Tsai, C. Lai, H. Chao, A. Vasilakos // In Big Data Technologies and Applications. – Springer, 2016. – PP. 13–52.

176) Udoeyop, A. Cyber profiling for insider threat detection / A. Udoeyop. – 2010. – PP. 263-262.

177) Usha, D. A Survey of Big Data Processing in Perspective of Hadoop and Mapreduce / D. Usha, A. Aps // International Journal of Current Engineering and Technology. – 2014.

178) Venkatraman, S. SQL Versus NoSQL Movement with Big Data Analytics / S. Venkatraman, K. Fahd, S. Kaspi, R. Venkatraman // International Journal of Information Technology and Computer Science. – 2016. – Iss. 12. – PP. 59–66.

179) Wall, D. Enemies within: Redefining the insider threat in organizational security policy / D. Wall // Security journal. – 2013. – Vol. 26. – Iss 2. – PP. 107-124.

180) Wang, H. An overview on the roles of fuzzy set techniques in big data processing: Trends, challenges and opportunities / H. Wang, Z. Xu, W. Xu // Knowledge-Based Systems. – 2017. – Vol. 118. – PP. 15–30.

181) Wang, L. Machine learning in big data / L. Wang // International Journal of Advances in Applied Sciences. – 2016. – Vol. 4. – PP. 117–123.

182) Wilkins, S. CCNP Security Secure 642-637 Official Cert Guide / S. Wilkins, T. Smith. – Indianapolis: Cisco Press, 2011. – 800 p.

183) Xi, X. Method and System for Detecting Anomalous User Behaviors: An Ensemble Approach / X. Xi et al. // SEKE. – 2018. – PP. 263-262.

184) Yen, T. Beehive: Large-Scale Log Analysis for Detecting Suspicious Activity in Enterprise Networks / T. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson // Proceedings of the 29th Annual Computer Security Applications Conference. – 2013. – PP. 199

185) Zaharia, M. Improving MapReduce performance in heterogeneous environments / M. Zaharia, A. Konwinski, A. Joseph, R. Katz, I. Stoica // Proceedings of the USENIX Symposium on Operating Systems Design and Implementation. – San Diego, CA, USENIX, 2008. – PP. 29–42.

186) Zegzhda, D. Approaches to modeling the security of cyberphysical systems / D. Zegzhda, Y. Vasil'ev, M. Poltavtseva // Automatic Control and Computer Sciences. – 2018. – Vol. 52. – Iss. 8. – PP. 1000-1009.

187) Zhang, J. Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems / J. Zhang, T. Li, D. Ruan, D. Liu // International Journal of Approximate Reasoning. – 2012. – Vol. 53. – PP. 620–635.

188) Zhou, L. Posterior probability based ensemble strategy using optimizing decision directed acyclic graph for multi-class classification / L. Zhou, H. Fujita // Information Sciences. – Vol. 400. – PP. 142–156.

189) Zuech, R. Intrusion detection and Big Heterogeneous Data: a Survey / R. Zuech, T. Khoshgoftaar, R. Wald // Journal of Big Data. – Springer, 2015. – PP. 1-42.

190) “CERT Insider Threat Data Set,” Software Engineering Institute, Carnegie Mellon University, CERT Division and Exact Data LLC [Электронный ресурс]. – Режим доступа: https://www.cert.org/insider-threat/tools/.

Приложение А

Копии актов о внедрении результатов диссертационной работы

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА» (СПбГУТ)

Санкт-Петербург

УТВЕРЖДАЮ
Первый проректор - проректор по учебной работе
д.т.н., проф. _____ Г. М. Машков
«_31_» _01_ 2020 г.

Акт

об использовании результатов диссертационной работы
Ушакова Игоря Александровича
«Обнаружение инсайдеров в компьютерных сетях на основе комбинирования экспертных правил, методов машинного обучения и обработки больших данных» в учебном процессе университета

Настоящий Акт составлен в том, что результаты диссертационной работы Ушакова Игоря Александровича, а именно:

- модель представления больших данных об инсайдерских атаках в формате NoSQL, обеспечивающая хранение и анализ признаков пользователей в компьютерных сетях в различные моменты времени;
- модель и алгоритмы комбинированного применения экспертных правил и методов машинного обучения в интересах обнаружения инсайдерских атак;
- методика обнаружения инсайдеров в компьютерных сетях с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных;
- архитектура и программная реализация системы обнаружения инсайдеров в компьютерных сетях с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных

используются кафедрой «Защищенные системы связи» федерального государственного бюджетного образовательного учреждения высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» в учебном процессе на старших курсах обучения бакалавров по направлению подготовки 10.03.01 «Информационная безопасность» по дисциплинам «Безопасность компьютерных сетей» (рабочая программа дисциплины, регистрационный № 18.05/522-Д) и «Безопасность беспроводных локальных сетей» (рабочая

программа дисциплины, регистрационный № 16.05/230-Д) при чтении курсов лекций, проведении практических занятий и лабораторных работ.

Председатель комиссии:

Заведующий кафедрой
«Защищенные системы связи»,
к.т.н., доцент Красов Андрей Владимирович

Члены комиссии:

учёный секретарь кафедры
«Защищенные системы связи»,
к.т.н., доцент Кушнир Дмитрий Викторович

МИНОБРНАУКИ РОССИИ
федеральное государственное
бюджетное образовательное учреждение
высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПРОМЫШЛЕННЫХ ТЕХНОЛОГИЙ
И ДИЗАЙНА»
(СПбГУПТД)

Б. Морская ул., д. 18, Санкт-Петербург, 191186
Тел. (812) 315-75-25 Факс (812) 571-95-84
E-mail: rector@sutd.ru http://www.sutd.ru
ОКПО 02068605, ОГРН 1027809192102,
ИНН/КПП 7808042283/784001001

31.01.2020 № 38-08-10/03-37

на № _____ от _____

УТВЕРЖДАЮ

Проректор по научной работе

СПбГУПТД

д.т.н., проф. Макаров А.Г.

«_____» 2020г.

АКТ

о внедрении научных результатов диссертационной работы Ушакова Игоря Александровича на тему: «Обнаружение инсайдеров в компьютерных сетях на основе комбинирования экспертных правил, методов машинного обучения и обработки больших данных»

Комиссия в составе: зам. зав. кафедрой интеллектуальных систем и защиты информации (ИСЗИ) к.т.н. доц. Вагнер В.И., к.т.н., ст. преп. Штеренберг С.И., составила настоящий акт о том, что научные результаты, Ушакова И.А., полученные им в ходе диссертационного исследования на тему «Обнаружение инсайдеров в компьютерных сетях на основе комбинирования экспертных правил, методов машинного обучения и обработки больших данных», используются на кафедре ИСЗИ СПбГУПТД при подготовке лекционно-практических занятий, а именно:

1. Методика обнаружения инсайдеров в компьютерных сетях с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных для направления подготовки бакалавров 10.03.01 – «Информационная безопасность» по дисциплине «Комплексная защита информации на предприятии».

2. Архитектура и программная реализация системы обнаружения инсайдеров в компьютерных сетях с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших

данных для направления подготовки бакалавров 10.03.01 – «Информационная безопасность» по дисциплине «Технологии и методы программирования».

Комиссия считает, что внедрение указанных научных результатов Ушакова И.А. в образовательный процесс СПбГУПТД позволило повысить качество подготовки бакалавров по направлению подготовки бакалавров 10.03.01 Информационная безопасность.

Комиссия также отмечает практическую значимость и новизну полученных в работе результатов.

Председатель комиссии:

зам. зав. кафедрой Интеллектуальных систем и защиты информации, к.т.н., доцент		Вагнер Виктория Игоревна

Члены комиссии:

ст. преп. Интеллектуальных систем и защиты информации, к.т.н.		Штеренберг Станислав Игоревич

РОСКОМНАДЗОР

УПРАВЛЕНИЕ ФЕДЕРАЛЬНОЙ СЛУЖБЫ ПО НАДЗОРУ В СФЕРЕ СВЯЗИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И МАССОВЫХ КОММУНИКАЦИЙ ПО СЕВЕРО-ЗАПАДНОМУ ФЕДЕРАЛЬНОМУ ОКРУГУ
(Управление Роскомнадзора по Северо-Западному федеральному округу)

ул. Галерная, д. 27, Санкт-Петербург, 190000
тел./факс: (812) 678-95-26
e-mail: rsockanc78@rkn.gov.ru

10.01.2020 № 10021-09178

на № _____ от _____

УТВЕРЖДАЮ

Руководитель Управления Роскомнадзора по Северо-Западному федеральному округу, к.т.н.

Д.В.Сахаров

«10» января 2020г.

АКТ

о внедрении научных результатов, полученных Ушаковым Игорем Александровичем

Комиссия в составе: председателя - Потехина Игоря Юрьевича, заместителя руководителя Управления, к.ф.-м.н, членов комиссии: – Мельникова Сергея Евгеньевича, заместителя руководителя Управления и Скакуна Артема Александровича Управления, к.ф.н., составила настоящий акт том, что при разработке мероприятий по технической защите информации в распределенной информационно-вычислительной сети Управления Роскомнадзора по СЗФО использованы следующие результаты диссертационной работы Ушакова Игоря Александровича «Обнаружение инсайдеров в компьютерных сетях на основе комбинирования экспертных правил, методов машинного обучения и обработки больших данных», а именно:

1.Модель представления больших данных об инсайдерских атаках в формате NoSQL, обеспечивающая хранение и анализ признаков пользователей в компьютерных сетях в различные моменты времени.

2. Модель и алгоритмы комбинированного применения экспертных правил и методов машинного обучения в интересах обнаружения инсайдерских атак.

3.Методика обнаружения инсайдеров в компьютерных сетях с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных.

4.Архитектура и программная реализация системы обнаружения инсайдеров в компьютерных сетях с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных.

Комиссия отмечает практическую значимость и новизну полученных в работе результатов.

Председатель комиссии: И.Ю. Потехин

Члены комиссии: С.Е. Мельников

А.А. Скакун

00448

УТВЕРЖДАЮ
Генеральный директор
ООО «Фаст Лейн»
_____ Княжицкий В.В.
«20» ____01____ 2020 г.

Акт

об использовании результатов диссертационной работы
Ушакова Игоря Александровича
«Обнаружение инсайдеров в компьютерных сетях на основе комбинирования экспертных правил, методов машинного обучения и обработки больших данных»

Настоящий Акт составлен в том, что результаты диссертационной работы Ушакова Игоря Александровича, а именно:

- модель представления больших данных об инсайдерских атаках в формате NoSQL, обеспечивающая хранение и анализ признаков пользователей в компьютерных сетях в различные моменты времени;
- модель и алгоритмы комбинированного применения экспертных правил и методов машинного обучения в интересах обнаружения инсайдерских атак;
- методика обнаружения инсайдеров в компьютерных сетях с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных;
- архитектура и программная реализация системы обнаружения инсайдеров в компьютерных сетях с использованием комбинирования экспертных правил, методов машинного обучения и обработки больших данных

используются ООО «Фаст Лейн» в рамках рабочего процесса при организации безопасности компьютерной сети организации.

Председатель комиссии: _____ Княжицкий В.В.

Члены комиссии: _____ Петрунин А.Ю.

Приложение Б

Свидетельства о регистрации программ для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2019666737

«Компонент предобработки трафика в корпоративной компьютерной сети с использованием алгоритма Map Reduce в Hadoop кластере»

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» (СПбГУТ) (RU)***

Авторы: ***Ушаков Игорь Александрович (RU), Котенко Игорь Витальевич (RU), Овраменко Александр Юрьевич (RU)***

Заявка № **2019665946**
Дата поступления **05 декабря 2019 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***13 декабря 2019 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2019666959

«Система обнаружения инсайдера в корпоративной компьютерной сети, используя алгоритмы, основанные на экспертных правилах»

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» (СПбГУТ) (RU)***

Авторы: ***Ушаков Игорь Александрович (RU), Котенко Игорь Витальевич (RU), Пелёвин Дмитрий Владимирович (RU)***

Заявка № **2019665940**
Дата поступления **05 декабря 2019 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***17 декабря 2019 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2019666738

«Система обнаружения инсайдеров в корпоративной компьютерной сети с использованием технологий машинного обучения»

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» (СПбГУТ) (RU)***

Авторы: ***Ушаков Игорь Александрович (RU), Котенко Игорь Витальевич (RU), Твердохлебова Юлия Владимировна (RU)***

Заявка № **2019665944**
Дата поступления **05 декабря 2019 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***13 декабря 2019 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев